Where social noise and structure converge

Learning with social semantics

Cover design: Rombout Casander

Cover illustration: A donut chart of an LDA model with 10 topics trained on the contents of this dissertation. The network in the background is the author's Twitter network visualized using Gephi.

ISBN 978-90-393-6033-0

NUR 980

Where social noise and structure converge

Learning with social semantics

Waar sociale ruis en structuur samenkomen

Leren met sociale semantiek
(met een samenvatting in het Nederlands)

Proefschrift

ter verkrijging van de graad van doctor
aan de Universiteit Utrecht
op gezag van de rector magnificus, prof.dr. G.J. van der Zwaan,
ingevolge het besluit van het college voor promoties
in het openbaar te verdedigen op
dinsdag 7 januari 2014 des middags te 2.30 uur

door

Frederik Thomas Markus

geboren op 3 december 1983 te Zeist

Promotor:	Prof.dr. J.E.J.M. Odijk
Co-promotor:	Dr. P. Monachesi

Contents

Appendices

Chapter 1

Introduction

1 Introduction

An uncle listens with kind interest to his little nephew who starts talking enthusiastically about his new hobby; 'dinosaurs'. At first he still understands that his nephew's favorite dinosaur is a type of 'Pterosaurs'; a winged variety. He loses track when the nephew rapidly tells him about the differences between the 'Quetzalcoatlus' and 'Hatzegopteryx' during the 'Triassic'. He tries to remember the name of his nephew's favorite dinosaur such that he can buy him a replica for his next birthday, but in the end gets confused due to all the terminology involved.

The uncle did not get confused because he was not interested or smart enough, he just does not know enough about dinosaurs. It is almost as if the young nephew was speaking a different language although it is still his own. The nephew and the uncle thus differ in their knowledge of certain subjects. More specifically, the nephew knows more about the way different species of dinosaur relate to each other, their names, appearances and when they lived. The uncle has a basic understanding of what dinosaurs are and when they lived, but lacks detailed knowledge and the associated terms.

This difference in domain knowledge can be understood from a linguistic perspective. From this perspective, the differences that arise in the conversation between the nephew and his uncle point to a "vocabulary problem" (Furnas et al., 1987). This link between the use of a certain vocabulary and the knowledge of certain domains also applies to other areas of interest such as computer supported learning, i.e. *e-learning*. Current methods of accessing online information heavily rely on the proper vocabulary in order to retrieve appropriate resources, e.g. traditional keyword-based search engines. Importantly, this can limit a learner's access to relevant resources and not using the appropriate vocabulary is a strong indication that the level of conceptual knowledge of an individual is insufficient.

This dissertation is concerned with the computational modeling of how language and knowledge interrelate in learning situations; access to knowledge is mediated by language and language is an expression of knowledge. Formal conceptualizations of domains, i.e. ontologies,

can be used to assist learners with educating themselves about new domains. However, these ontologies are frequently out of date. In order to address this, this dissertation presents an ontology enrichment methodology that includes new terms and concepts based on relevant information in Social Media. An ontology enriched with this information is better able to mediate between a learner and content available in Social Media. Additionally, the lexical differences between an individual and his or her domain of interest can be used to determine their level of understanding which, in turn, can be used to improve the recommendation of learning objects and automatic feedback.

The following sections will provide some context as to what theoretical issues this dissertation addresses (section 2), highlight what technical approach is taken to address the issues in the context of the Semantic Web and Social Web (section 3) and what impact the technology developed in this dissertation has on e-learning (section 4). Finally, in section 5, I will outline the specific computational approach proposed in this thesis and provide an outline of the individual chapters in section 6.

2 Context

The 'vocabulary problem' mentioned in the introduction is highly typical of learning situations. A learner is an individual that is new to a domain and does not know the proper *vocabulary*, but the expert community presupposes that the learner does. Additionally, the learner might not just lack the proper vocabulary, but the *conceptual structure* of the domain is also likely to be either absent or incomplete. Learners vary with respect to their 'lexical competence' , i.e the proficiency in which someone is able to recognize and use words in a language in the way that speakers of the language use them[1] (Marconi, 1995).

As such, the preferred terms that are used to denote concepts are tied to specific *communities* (Wenger and Snyder, 2000; Hung and Chen, 2001; Diederich and Iofciu, 2006). Collaboration among like-minded peers leads to better learning outcomes (Means et al., 2010), but requires learners to find complementary or more knowledgeable peers and the best learning objects for both their personal and collaborative needs. The ability for learners to accomplish these objectives greatly depends on their ability to communicate clearly with community members. The information revolution, and the rise of the Internet in particular, has revolutionized accessing, sharing and creating information. This growing amount of information is increasingly shared, created and annotated on Social Media sites. People are currently, on average, spending 20% of their PC time and 30% of their mobile time on Social Media[2]. Social Media provide platforms on which communities of experts share, discuss and create content. These high quality sources of information can also be exploited by novices for learning purposes. For example, MIT currently has 126,630 subscribers and 30,852,623 views on YouTube, a video sharing website on a wide range of subjects ranging from advanced chemistry to linear algebra[3].

[1] Loosely based on the definition from `http://www.sil.org/lingualinks/languagelearning/otherresources/gudlnsfralnggandcltrlrnngprgrm/WhatIsLexicalCompetence.htm` accessed 13-12-2012

[2] `http://www.nielsen.com/us/en/reports/2012/state-of-the-media-the-social-media-report-2012.html` accessed 14-06-2013

[3] The figures are taken from `http://www.youtube.com/user/MIT` on 18-11-2011

However, the use of the Web in support of learning is highly dependent on 'lexical competence' and the 'vocabulary problem'. This mandatory requirement for lexical competence is fragile in a learning context, because it is often lacking, i.e. an individual may not be able to express his or her information need (Taylor, 1962). It may not be assumed that novices have sufficient knowledge of proper terminology or that they can assess which specific terminology and associated conceptual knowledge they have acquired to a sufficient degree[4]. The sheer amount of resources that is available to learners can drive them towards surface learning and fact finding (Beattie IV et al., 1997), with little motivation towards deep learning, i.e. understanding, reflection and abstraction. This dissertation will address some of these concerns using a combination of knowledge engineering and language technology.

3 Social Semantic Web

Knowledge engineering (Studer et al., 1998) is concerned with the modeling and transfer of knowledge by means of formal domain models. More recently, knowledge engineering practices have been applied to the Web, effectively giving rise to the Semantic Web effort (Berners-Lee et al., 2001; Shadbolt et al., 2006; Antoniou and Van Harmelen, 2004). Ontologies, i.e. formal domain models, play a central role in the Semantic Web, through its uniform usage of knowledge representation standards such as RDF and OWL. The Semantic Web allows one to interconnect ontologies and datasets created by different parties that are located on arbitrary parts of the Internet. It is possible to represent knowledge using an ontology with no regard as to how the knowledge is expressed, i.e. what words are used to denote the concepts, properties and relations in natural language documents. Ontologies that do take these lexical aspects into account are referred to as lexicalized ontologies (Buitelaar and Cimiano, 2008). Ontologies make a clear distinction between the conceptual and lexical aspects of domain knowledge. They can be used to support knowledge retrieval, understanding and discovery, but in practice knowledge is shared using vague or ambiguous terms. Proper lexicalizations in the context of a lexicalized ontology are crucial when one wants to link ontologies to the knowledge that is used by people in practice. Relations between terms and their respective concepts can be automatically established using language technology in order to leverage the advantages of ontologies for resources that have not been annotated using ontological concepts.

One prominent example of the use of terms as opposed to concepts for the annotation of resources is Social Media. Social Media have transformed the way in which users interact with content on the Web. An important characteristic of many of these Social Media is that they enable users to use short terms, i.e. *tags*, to label resources of interest. They allow a community of users to structure information using a self-constructed vocabulary, i.e. a folksonomy Vander Wal (2007); Mika (2005). A folksonomy does not explicitly distinguish between concepts and terms, but only establishes a shared vocabulary in a social context.

[4] I would like to illustrate that these issues with misaligned vocabularies are common in practice, Recently, I had been thinking about information distribution in lossy semi-structured peer to peer networks using a type of pseudo-random peer selection scheme. I had discussed it with friends and related developers and we believed the approach to be unique and innovative. We even started to employ newly invented jargon to refer to the ideas. It was only when an online acquaintance used the term 'gossip network' that I suddenly had access to a significant amount of research literature. This simple term allowed me to progress on the subject at an accelerated pace.

The adoption of a common vocabulary in a community is attested by the shared use of terms. Information becomes more accessible and easier to navigate when a community of users aligns their respective vocabularies. As previously mentioned, Social Media are actively used to share arbitrary information, which includes learning objects, i.e. resources that can be used for learning. However, no explicit support for learning is present in Social Media. The domain structure is only indirectly expressed through term co-occurrences, whereas an ontology makes domain structure much more accessible in an explicit manner.

The mediation between knowledge sources and learners is historically the domain of teachers, tutors and librarians, but automated systems can provide immediate 24/7 feedback and support which better fits a continuous online environment. One way this can be accomplished is through the application of Language Technology and Knowledge Engineering in the form of Semantic Web technologies. More specifically, how technologies can be used to assist individuals in getting access to, and understand, information.

4 e-Learning

Books in a library are for many no longer the primary source of knowledge. Books are increasingly replaced by alternatives such as blog articles, online encyclopedias and tweets. Information is also increasingly accessed, shared and filtered through Social Media. Important aspects of learning have transitioned to Internet and computer-based technology. These computer-supported approaches to learning are referred to as *e-learning*. E-learning concerns the design and use of computational tools that support the learning process on both the collaborative and individual levels. The field of Computer Supported Collaborative Learning (CSCL) (Stahl et al., 2006) aims to design systems that support the social learning processes of individuals and groups. It is important, not to underestimate the impact of lightweight community vocabularies in Social Media. Apparently, annotations such as tags, allow scattered Communities of Practice (Wenger and Snyder, 2000) to emerge and align their respective vocabularies. The assimilation of the community vocabulary by individuals is actually an important component of learning and development, where individuals learn to "master a speech genre" (Bakhtin et al., 1986) and integrate within the associated "Community of Practice" (Wenger and Snyder, 2000) in a situated environment (Hung and Chen, 2001).

It is desirable to exploit the conceptual and lexical aspects of knowledge for learning support through the integration of the Social Web and the Semantic Web, because knowledge itself is fundamentally social. The vocabulary that learners employ plays a central role in my approach, because it determines their access to information and is an expression of their current level of understanding. Automatic recognition of this interplay between the vocabulary and the conceptual understanding of learners is important for both resource retrieval and automated feedback and learning support. This can only be accomplished if a community's understanding, as represented by a folksonomy, is integrated with that of formal domain ontologies on both a lexical and conceptual level.

The EU FP7 LTfLL project ("Language Technology for Life-Long Learning") (2008-2011)[5] (Berlanga et al., 2009) emphasized the important relations among learning, language and

[5] http://www.ltfll-project.org

knowledge. The goal was to exploit existing Natural Language Processing (NLP) and Knowledge Engineering tools and techniques to provide a set of services for assisting learners in a life-long learning context, i.e. learning without institutional support or in the work place. The work performed in the LTfLL-project can be roughly divided in three themes (Berlanga et al., 2009). First, determine the current position of a learner through the analysis of a learner's portfolios, e.g. blog posts (Wild et al., 2010) and visualize a learner's knowledge using pseudo-*concept maps*. Second, provide feedback on the quality of a learner's collaborative behavior, i.e. course-specific chat sessions and forums, using NLP and Social Network Analysis (Rebedea et al., 2010b,a) and on the quality of a learner's course summaries using Latent Semantic Analysis (Loiseau et al., 2011). Third, support social, formal and informal learning using ontologies, tagging, Social Network Analysis and concept annotation grammars (Monachesi and Markus, 2010b; Monachesi et al., 2011). The various aforementioned services developed in the context of the LTfLL project were integrated using widgets in order to construct a Personalized Learning Environment. Some of the contributions presented in this dissertation have roots in the LTfLL-project and vision.

5 Approach

In order to assist learners in a digital environment computational methods are required that can automatically mediate between their conceptual understanding and vocabulary, because the vocabulary and associated lexical competence is key to information access in a learning context. More specifically, access to suitable learning objects is determined by (1) *knowledge* of the concepts of interest, (2) use of the proper *vocabulary* to communicate about them, and (3) access to the *community* that maintains and develops a knowledge domain, and (4) the ability to act in the community by using the proper vocabulary in response to community artifacts and interaction. Figure 1.1 presents these core aspects and their interaction schematically. The fourth aspect is not depicted separately, but is a function of the other three combined.

The use of an ontology, i.e. a formalization of a conceptualization (Gruber, 1993), takes a crucial role in my approach to address the vocabulary problem. Ontologies have been designed to succinctly describe the important concepts and relations of a domain in a way that can be made useful for learners (Monachesi et al., 2008, 2010, 2011; Westerhout et al., 2010, 2011). For example, ontologies can help identify whether two terms, although different in form, actually refer to the same or related concept and explain how concepts are related. In

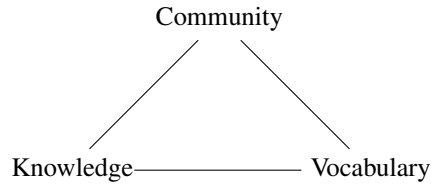


Figure 1.1: Primary concepts of this dissertation.

the context of e-learning, ontologies represent the domain as it is viewed by experts. As such, an ontology can fulfill the role of a teacher or tutor in the sense that it can mediate between a novice and an expert-approved conceptualization of a domain.

The digital environment that learners employ for learning is increasingly supported by Social Media, i.e. blogs, social bookmarking sites, microblogs, etcetera. Many learners are already familiar with them and their popularity and size supports that they are scalable methods for sharing content. Learners can search for information in these Social Media using specific terms (keywords) or automatically discover new content through recommendation. Frequently used tags are considered to be an expression of the *community vocabulary*.

For ontologies to succeed in fulfilling their mediating role in learning, it is vital to relate the vocabulary that is employed by novices, e.g. on Social Media, to the concepts and terms from an ontology. This can be achieved by embedding knowledge in a social environment and, vice versa, recommending appropriate knowledge in social contexts. In order to accomplish this, an integration of the community vocabulary required, i.e. the tags and the concepts that they represent need to be integrated with a pre-existing ontology. This integration allows ontologies to improve access to information by mediating between the vocabulary that is used by novices and the expert-approved conceptualization and vocabulary. As a result, the 'vocabulary problem' that exists between novices and Communities of Practice can be addressed. This integration of conceptual and lexical knowledge is non-trivial, as many terms and abbreviations are ambiguous, i.e. a single term can be interpreted as more than one concept. A robust link between the vocabulary of a community and the concepts from an ontology requires a method for automatically resolving this problem using a disambiguation algorithm. Once these challenges are met, *semantic search* can be employed to considerably improve the quality of search results in Social Media.

The lexical differences between novices and experts can also be used for assessment purposes. More specifically, these differences help determine which parts of the domain have been sufficiently acquired and which require additional work. This is accomplished through the exploitation of the lexical differences between that of a community, i.e. a network of individuals sharing resources on Social Media, and a learner. Language technology, and more specifically, topic modeling techniques, allow one to quantitatively analyze these lexical differences and use them to provide personalized feedback. Such feedback can be used by learners, for example, for reflective purposes and allow them to moderate their learning strategies in response. Language technology can be used to provide such personalized feedback on a massive scale, virtually instantly and in an objective and non-judgmental manner (Jordan, 2011).

There are thus four important elements to this dissertation:

1. Integration of a community's conceptualization, as expressed in Social Media using tags, with that of domain ontologies, i.e. the formal domain conceptualizations of experts. (chapter 3)
2. A robust method of linking terms to appropriate concepts in order to resolve ambiguous references and to move from the lexical to the conceptual level of analysis and filtering. (chapter 4)
3. Enhanced methods for retrieving resources, i.e. semantic search, that exploits the knowledge captured by domain ontologies in order to improve the relevance and quality of online search requests. (chapter 5)
4. Automatic assessment techniques based on the vocabulary employed by learners for reflection and evaluation purposes. (chapter 6)

The following section will describe each of these aspects in greater detail and give a short outline of each chapter.

6 Outline

This section summarizes the content of each chapter in order to give an overview of this dissertation, how the chapters interrelate and what are the main contributions of each chapter.

Chapter 2 Provide an introduction to three important research areas that are employed in various chapters. This includes: (1) an introduction to the Semantic Web and its relevant components, (2) an introduction to Social Media and social tagging and (3) a short introduction to educational theory and, more specifically, social constructivism for providing a theoretical framework with regard to learning in a social context.

Chapter 3 This chapter is primarily concerned with the integration of the community vocabulary and conceptualization with ontologies. This work started in the context of the LTfLL-project in 2009 and was largely completed in 2011. Ontologies can help people navigate a domain that they are not familiar with by starting from known points of reference. For example, the uncle in the introduction might use an ontology to start from the concept of 'Dinosaur' and navigate to the specific type his nephew mentioned. However, an ontology can only help the uncle when there exists lexical and conceptual overlap between how he understands the domain and how an expert, such as his nephew, understands it. Laymen might use 'incorrect' lexical items when communicating about dinosaurs, but these are necessary points of reference within the conceptual structure of an ontology. These familiar points of reference allows him to navigate the ontology and learn new things. The methodology presented in this dissertation aligns the conceptualization of an existing ontology with a community's understanding and vocabulary. This methodology is referred to as *Social Ontology Enrichment* in

chapter 3. More specifically, the approach presented in chapter 3 unifies the expert view of the domain, as represented by a domain ontology, with the community's understanding of the same domain, as represented by the tags that a community uses in Social Media. This is accomplished through the use of Wikipedia as an intermediary in order to integrate formal concepts from an ontology with the loose conceptualization of a community.

The primary contribution of this chapter involves a novel methodology for the integration of the domain conceptualizations of communities in Social Media and expert-approved conceptualizations available in domain ontologies. This concerns both the enrichment of an ontology with new concepts and relations, but also considerable lexical enrichment.

Chapter 4 The previous paragraph may have suggested that a relationship between a term and a concept is a trivially established. However, in practice this is far from the truth, especially in a noisy Social Media environment. One term might refer to many different concepts, i.e. it is ambiguous. Similarly, the terms 'Tyrannosaurus Rex' and 't-rex' might be two terms for a single concept, i.e. they are synonyms. Automated identification of the relationship between terms and concepts can help users access resources that refer to the same concept, but are expressed by different terms. This is also an important component of Social Ontology Enrichment as introduced in chapter 3. This chapter focuses on the *disambiguation* of terms from Social Media, i.e. to link a term to an appropriate concept. More specifically, the approach uses the biggest centralized source of concepts on the Web: Wikipedia, as a point of reference. The graph-structure of Wikipedia, available as datasets through DBpedia (Bizer et al., 2009b), is leveraged to perform this disambiguation automatically for any domain covered by Wikipedia. The research on the disambiguation algorithm started in the context of the LTfLL-project in 2009 and has been considerably extended as part of this dissertation. Chapter 4 will show the effectiveness of this approach to the disambiguation of tags in particular, i.e. coocurring terms in Social Media.

The primary contribution of this chapter is an algorithm for tag disambiguation based on associative relations in DBpedia (Bizer et al., 2009b). Additionally, it establishes a relatively large test set for tag disambiguation evaluation and it is on par with the state of the art and conceptually elegant and transparent. Finally, it also quantifies the impact of semantic coherence as part of the disambiguation and its contribution to disambiguation accuracy.

Chapter 5 The techniques developed in chapter 3 and chapter 4 are integrated in chapter 5 in order to improve online keyword-based search. It specifically deals with situations where a user employs a keyword that is ambiguous. For example, a search for 'ajax' will return resources about either Greek mythology, a Dutch soccer club or a type of Internet technology. However, in most cases only one of these is actually intended by the user. *Semantic Search* automatically identifies the meaning, i.e. concept, of an ambiguous search query in the context of a domain ontology. As a result, it can distinguish between relevant and irrelevant resources on a conceptual level, which does not suffer from term ambiguity. In effect, results that are not relevant to a user's search request can be automatically removed, thereby significantly increasing the quality and specificity of the search results.

The main contribution of this chapter involves a semantic search methodology for arbitrary tag-based Social Media that makes use of a domain ontology to improve the relevance of

search results for ambiguous terms. It does not rely on detailed conceptual annotation of resources, but employs the disambiguation algorithm, introduced in chapter 4, both to perform ontology mapping and to realize concept-based filtering of search results.

Chapter 6 The previous chapters have focused on improving the access to information, resources and domain conceptualizations. This chapter is primarily concerned with the automated analysis of the language used by individuals in the context of a learning corpus. Increased divergence between the lexico-semantic associations made by a learner and those present in a corpus of expert-approved material is evidence for poor understanding. Topic models are employed to provide a computational model of the lexical semantics of a corpus. Individuals that have a good grasp of the domain will be able to successfully interpret the topics, whereas individuals that lack the knowledge do not. This approach has been evaluated in the context of two courses at Utrecht University. Chapter 6 will show that there is a significant correlation between university student's ability to interpret a topic model and their actual learning outcomes, i.e. exam grades. As a result, this light-weight methodology can be employed to efficiently evaluate a learner's knowledge for a domain purely based on a learning corpus specific to that domain.

The primary contributions of this chapter are threefold. First, it shows that domain understanding correlates with a learner's ability to interpret a topic model. Second, it makes a case for personalized topic models and shows that they can outperform regular topic models with respect to learning outcomes. Third, the chapter establishes several metrics to evaluate manual topic labeling that are predictive of a learner's domain understanding.

Chapter 7 Overall conclusions of the work presented in this dissertation and discussion of future work.

Chapter 2

The Semantic Web, Social Networks and Learning

1 Introduction

The Internet is a dynamic network of systems that accommodates to changing information requirements and new forms of interaction. A multitude of trends can be distinguished on the Internet, but this chapter will focus on just two. The first is the Semantic Web, a collection of approaches to creating and exchanging structured information that can be exploited by computers. This information can be used, for example, to support more intelligent ways to answer questions and retrieve information. The second trend is the Social Web which refers to the increasing impact of social media and its influence on how people communicate and share content. Both the Semantic Web and the Social Web deal with managing information and communication, but each with its own emphasis, on structured information and social processes, respectively.

Learning and education is an area where the Social and Semantic Web can play a relevant role. The educational theory of social constructivism emphasizes that (conceptual) learning is about the acquisition of knowledge in a social context. Both the knowledge aspects of the Semantic Web and social aspects of the Social Web can play crucial roles during learning. Semantic Web artifacts like domain ontologies support formalizing and exchanging knowledge. The Social Web provides support for sharing and discussing that knowledge within a learning community.

The aim of this chapter is to provide the necessary background information for understanding subsequent chapters. Section 2 provides an overview of the Semantic Web and its most relevant components: the RDF language, ontologies and datasets. Section 3 provides an overview of the Social Web, its origins and relevant concepts such as tags and folksonomies. Section 4 introduces essential aspects of educational theory followed by an overview of computer-supported collaborative e-learning. Section 5 brings the sections on the Semantic Web, Social Web and learning together and illustrates why their combination is practically and theoreti-

cally appealing in the context of educational theory.

2 Semantic Web

The amount of information on the Web is increasing every day. In order to deal with this, computers need to be able to process data more intelligently. However, they require information that is structured in a way that is easier to process and automatically interpret than information represented in natural language. At present, the primary method of getting information is to compose a search request using natural language and then manually inspecting the search results for the desired information. Using the right type of structured information, a computer should be able to answer questions directly, instead of people having to read through all the documents to find the answer themselves.

The *Semantic Web* (Berners-Lee et al., 2001; Shadbolt et al., 2006; Antoniou and Van Harmelen, 2004) is an interdisciplinary and international effort with the aim of creating the infrastructure and tools for sharing meaningful data in an interoperable manner. All tools and techniques for the Semantic Web have been specifically designed to operate within a decentralized, large, online networked environment.

The following sections will go into detail as to what the Semantic Web consists of and discuss tools and techniques relevant in the context of this dissertation. More specifically, these sections will introduce ontologies (section 2.1), Linked Data & vocabularies (section 2.2) and reference repositories (section 2.3).

2.1 Ontologies

In order to achieve the goal of having smarter computers that can exploit large amounts of structured information, it is crucial to describe knowledge in a standardized way. Additionally, it is important to interconnect various bits of knowledge such that computers can make more informed inferences and are not restricted to reasoning about small isolated pieces of information. It might be beneficial to reason about the domains of history and geography together in order to identify important relationships between the two. For example, how the colonial age has affected the borders of states on the African continent. Knowledge of one domain is often intrinsically interconnected to that of another.

Each of these knowledge domains, e.g. history and geography, is unique, but that does not entail that each requires a different method of conceptualizing the information. For any domain, its *concepts* and *relations* can be captured using a universal type of formal structure. Such a formal structure is commonly referred to as either a knowledge model or an *ontology*. Ontologies are the foundation of the Semantic Web, because they are the preferred way of making human knowledge machine interpretable.

Ontologies achieve this by means of a clear model (a conceptualization) of the properties of the knowledge domain. An ontology is a "body of formally represented knowledge based on a conceptualization: the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them" (Genesereth and Nilsson, 1987).

The term 'properties' in the context of an ontology refers to the properties of an ontological concept such as its definition or date of modification. The term 'relations' refers to relationships between concepts. The fact that an ontology is a formalization of a conceptualization is important, because "every knowledge base, knowledge-based system, or knowledge-level agent is committed to some conceptualization, explicitly or implicitly" (Gruber, 1993).

This dissertation makes ample use of a special type of ontology called a *lexicalized ontology* (Buitelaar et al., 2006). A *lexicalized ontology* also includes the terms with which the concepts, objects and other entities are expressed in natural language. That is, a clear distinction is made between the identifier that is used to refer to the concept and the natural language terms that can be used to denote the concept. A term that denotes an ontological concept, and is clearly labeled as such, is called a *lexicalization* . A lexicalized ontology can distinguish between different types of lexicalizations. For example, a multi-lingual lexicalized ontology can distinguish between lexicalizations from different natural languages for the same concept. Another common distinction with respect to lexicalizations is that between *preferred*, *alternative* and *hidden terms*[1]. A preferred term is the first choice when representing the concept in natural language. An alternative term is an acceptable synonym for the same concept, but does not have the same status as the preferred term. A hidden term can be used to denote the concept, but its use is discouraged. An example of an area where hidden terms are useful is to accommodate slang.

Types of ontologies Generally, two different classes of ontologies are distinguished. First, some types of ontologies contain information about a specific domain, these are referred to as *domain ontologies*. The Semantic Web effort has led to quite a number of domain ontologies in a wide variety of domains[2] ranging from the biomedical sciences to music, computing (Monachesi et al., 2008) and movie artists. Domain ontologies are also the type of conceptualizations that are suitable for learning support (Monachesi et al., 2009) and play a crucial role in chapters 3 and 5.

Second, there are also abstract ontologies that describe relationships which are valid in many different domains. These are referred to as *upper level ontologies*. Both upper level and domain ontologies enforce a specific conceptualization. Most upper level ontologies deal with the conceptualization of "objects, processes, properties, relations, space, time, roles, functions, categories, individuals or similar. An upper-level ontology is an ontology that defines and axiomatizes these most general categories" (Hoehndorf, 2010). I will refer to the overall conceptualization of an ontology as 'conceptual structure'.

Most domain ontologies use an upper level ontology for their overall conceptual structure and then extend it with domain specific concepts and relations. Upper level ontologies provide a conceptual structure that is valid in multiple domains and can be used to interlink different domain ontologies. Upper level ontologies thus enable the integration of complementary domains in a single knowledge model. Ontology engineers frequently reuse existing upper level ontologies (Keet, 2011) such as UMBEL[3] which in turn relies on OpenCyc (Matuszek

[1] http://www.w3.org/TR/2009/NOTE-skos-primer-20090818/#seclabel

[2] For an impression of available ontologies see: http://www.obofoundry.org/, http://ckan.net/package?q=owl, http://datahub.io

[3] http://umbel.org/

et al., 2006), SUMO (Niles and Pease, 2001) or DOLCE (Masolo et al., 2002). When two complementary domain ontologies use the same upper level ontology, integration via the upper level ontology is trivial. Upper level ontologies play an important role in chapter 3. Note that both upper level and domain ontologies can be lexicalized ontologies.

All ontologies in the Semantic Web use the same underlying formal language (RDF) for describing the knowledge model that they constitute. Ontologies that require a more constrained domain conceptualization can use RDF-based standards like RDFS or OWL. These languages will be discussed in more detail in sections 2.1.1 and 2.1.2. The use of the same formal language (RDF) allows one to interlink ontologies from different domains.

The methodologies and tools to create ontologies have matured and are slowly being adopted by the industry. The flexibility and extensibility of Semantic Web ontologies are the key reasons for adoption by governments to share their public data (Sheridan and Tennison, 2010). This includes the US[4], UK[5] and Dutch[6] governments amongst others [7].

The following sections will go into some detail as to what the important properties of the RDF model and language are, what types of ontologies exist in the Semantic Web and how ontologies are created and maintained.

2.1.1 RDF

Ontologies are constructed using a low-level language describing the various parts of an ontology such as concepts, relations and terms. The language used to create ontologies using Semantic Web technologies is called *Resource Description Framework*, commonly abbreviated to RDF. The RDF standard was released in 1999 as a W3C recommendation[8]. The RDF language is unrestricted in the sense that it does not impose any limitation on the ontologies that one can create with it, i.e. ontologies that are created using RDF are not restricted to any particular domain. RDF can also be used to represent less complicated formalizations of knowledge such as vocabularies (section 2.2) and reference repositories (section 2.3).

The RDF language has been specifically designed to operate on the Internet. Its designers did not only want to formally describe knowledge, but also share, interlink and distribute it. Automated sharing and inference of semantic content is only possible if the information is stored in a machine interpretable form, such as RDF. Although there are other approaches for the exchange of data, XML being the prime example, the additional capabilities of RDF and its extensions offers compelling advantages (Berners-Lee, 1998; Decker et al., 2000). This chapter only includes aspects of RDF that are relevant in the context of this dissertation.

However, it is important to point out that RDF is not designed to be a universal replacement for all data formats. RDF provides a standard data model for complex data with many entities and complex relations between those entities. XML is generally more suited to the exchange of data of a limited, clearly predefined form. However, the choice for a specific data format

[4] http://www.data.gov/
[5] http://data.gov.uk/
[6] http://data.overheid.nl/
[7] http://en.wikipedia.org/wiki/Open_data retrieved 02-10-2012
[8] http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

Term	URI
“Country”	http://dbpedia.org/resource/Country
“Thomas Markus”	http://www.thomasmarkus.nl/foaf/me
“philosophy”	http://dbpedia.org/resource/Philosophy

Table 2.1: A table with some examples of terms and their associated concepts represented by URIs

Prefix	URL fragment
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
dbpedia	http://dbpedia.org/resource/
dbpedia-ontology	http://dbpedia.org/property/
foaf	http://xmlns.com/foaf/0.1/
dbpedia-owl	http://dbpedia.org/ontology/
yago	http://dbpedia.org/class/yago/

Table 2.2: A table of commonly used prefixes in this dissertation and the Semantic Web in general.

does not exclude the use of another. RDF and XML can be combined to leverage their respective advantages (Hunter and Lagoze, 2001). For example, XML documents can be annotated with RDF concept identifiers to make the semantic interpretations of certain tags or attributes transparent (Kemps-Snijders et al., 2009).

The Universal Resource Identifier (URI) and Universal Resource Locator (URL) also form an important part of the RDF language in order to support the interlinking of data. URIs and URLs act as identifiers that consistently and precisely denote entities such as people, concepts, classes and or categories. A specific type of URI, the URL, is used extensively in the Semantic Web. Every URL is a type of URI, but only URLs constitute actual addresses for the things that they refer to. URIs only act as identifiers. URLs are already in wide use on the Internet as part of websites. The Semantic Web extends the use of the URI/URL towards semantic entities (things/instances) and concepts as expressed by the term: ‘Country’ (a concept), ‘Thomas Markus’ (an instance)) and ‘philosophy’ (a concept). The entity expressed by the term ‘Thomas Markus‘ would be an example of an instance of the concept ‘Person‘. For example, `http://www.thomasmarkus.nl/foaf/me` is a unique URL identifier for the author of this dissertation. Another example is *URN:ISBN:9780140623642* which is a unique URI identifier for a book called: “Gullivers Travels”. The unique identifiers for the concepts and instances just mentioned are listed in table 2.1. I will use the term *resource* to refer to a URI that may be a concept or an instance.

URLs can become quite long as can be observed in table 2.1. It is therefore possible to define a so called *prefix* (shorthand) to refer to the commonly occurring parts of the full URL. Some common examples of such prefixes are listed in table 2.2. We can simplify a few of the previous example URLs in table 2.1 using the prefixes we have just defined in

table 2.2. Two URLs can be reduced in size using the prefixes: `dbpedia:Netherlands` and `dbpedia:Philosophy`. Prefixes will be used in this dissertation to increase readability and prevent unnecessary repetition of common URLs. The URLs that have been abbreviated, using a prefix, are however equivalent to their long form.

Everything on the Semantic Web that can be the topic of discussion has to have a unique identifier. The reuse and sharing of identifiers is strongly recommended in order to guarantee that different parts of the Semantic Web can contribute information about the same thing (identifier). The use of RDF does not necessarily entail that data from different sources can be integrated automatically, because different identifiers may be used for the same thing. This problem is similar to that of "a language in principle unintelligible to anyone but its originating user" (Candlish and Wrisley, 2012). In RDF this problem of an 'unintelligible private language' appears when two ontologies about the same domain use completely different identifiers for concepts, instances and relations.

RDF extensively uses URIs for unambiguously denoting the concepts and instances of ontologies. The use of shared identifiers then allows different ontologies to be interlinked on the Semantic Web. As already mentioned, an ontology also has relations between concepts in addition to concepts and instances. The relations describe how concepts relate to each other. The same strategy for concepts is applied to relations; all relations are represented by a unique URL. Although concepts and relations are expressed using URLs, there is also a need for types of information which cannot be expressed using a URL. This includes dates, strings, numbers, booleans, etc. These types of information do not fit within a URL identifier and are referred to as *literals*. For example, the string literal 'thomas markus' can be the name of the author of this dissertation who is identified by the unique URL *http://www.thomasmarkus.nl/foaf/me*. Literals can be used to represent concept lexicalizations.

In order to distinguish between the literals of different languages additional metadata needs to be added to each literal to reflect this fact. The additional metadata about the natural language to which a literal belongs is called the *language tag*. Using language tags, lexicalizations from different languages denoting the same concept can be formally modeled as shown in lines 4, 5 and 6 in listing 2.1.

```
1 <http://www.JohnDoe.me/foaf> dbpedia:loves <http://MaryJane.me/foaf> .
2 dbpedia:Tweetie              rdf:type      dbpedia:Bird .
3 <http://www.thomasmarkus.nl> foaf:name     "Thomas Markus" .
4 dbpedia:House                rdfs:label    "home"@en .
5 dbpedia:House                rdfs:label    "maison"@fr .
6 dbpedia:House                rdfs:label    "huis"@nl .
```

Listing 2.1: "Example of an RDF-fragment expressed using the N3 syntax"

An RDF language fragment, such as the one shown in listing 2.1, consists of a finite number of simple facts called *statements*. The form of these statements is very simple, but is expressive enough for describing arbitrary knowledge. A statement in RDF always consists of three parts and it is for this reason that statements are also commonly referred to as *triples*. Each RDF statement is analogous to sentences of the form "Subject Predicate Object". For example,

"John loves Mary" or "Tweetie saw a pussy cat". An RDF statement consists of a URI in the first position and refers to an instance or a concept. A URI in the second position refers to a relation. Finally, the third position contains a URI or literal. The RDF example in listing 2.1 contains 6 statements in total. Any type of information that can be expressed using binary predicates can be expressed directly using the RDF-language. Information requiring ternary or unary predicates will have to be transformed to a set of binary predicates. The use of URIs prevents confusion as to which concept or person is referenced in an RDF statement. It is also clear what the exact relation is between the subject and object of each statement.

If one has to manage a lot of triples, the use of an RDF database may be required. RDF databases are commonly referred to as 'triple stores'. They provide scalable storage and querying capabilities for arbitrary RDF. An important property is that triple stores treat the RDF data that they store as a **set** of triples, i.e. a triple is stored only once even if it is added repeatedly.

Specialized query languages for RDF have been created that make use of the triple-based structure. At first there were competing dialects such as RDQL[9] and SeRQL (Broekstra and Kampman, 2003). In 2008, the W3C promoted the SPARQL query language to recommendation status. Its aim was to create a single standard query language for all RDF repositories. SPARQL is for triple stores the equivalent of SQL for relational databases. This effort is in line with the design goals of RDF with respect to standardizing the way information is stored, queried and accessed in the Semantic Web and the Web at large.

2.1.2 RDF-based ontology languages

RDF is thus a flexible datamodel for the representation of interoperable semantic information. As previously stated, the RDF data model is expressive enough to describe any conceptualization. However, RDF by itself imposes no semantic constraints, which allows one to create meaningless, but syntactically valid RDF. Listing 2.2 shows an example of meaningless RDF. In order to impose semantic constraints on RDF a so called *schema language* is required.

```
lt4el:ComputerMouse     lt4el:ComputerMouse     lt4el:ComputerMouse .
lt4el:Java              lt4el:Java              lt4el:Java .
lt4el:ComputerLanguage  lt4el:ComputerLanguage  lt4el:ComputerLanguage .
```

Listing 2.2: A syntactically valid RDF-fragment that is semantically meaningless.

An example of an extension of the RDF standard from 1999 that provides some support for schemas is the RDF-Schema (RDFS)[10] standard. RDFS was promoted to recommendation status in 2004. Just like any other Semantic Web standard it consists of a number of RDF relations and classes, but these have been given special meaning in computer software that supports RDFS. An RDF *schema*, such as RDFS, allows one to specify the "organization of vocabularies in typed hierarchies: subclass and subproperty relationships, domain and range

[9] http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/
[10] http://www.w3.org/TR/rdf-schema/

restrictions, and instances of classes" (Antoniou and Harmelen, 2009, p.93). RDFS contains the subsumption relation between concepts which is very common for domains that have some sort of hierarchical structure, e.g. the animal kingdom or geographical data. It is also possible, using the RDFS standard, to restrict the domain and range of relations. For example, the *drinks* relation can only hold between animals (domain) and liquids (range).

When a greater amount of specificity is required it is possible to transition to the OWL standard[11], completed in 2004, for ontologies. Each transition from RDF to RDFS to OWL introduces more domain invariant classes and relations that allow ontology designers to construct more detailed knowledge models with more restrictions. The transition from RDF to RDFS to OWL also reflects the chronological order in which each standard appeared. The most recent version of OWL is OWL 2[12]. It was completed in 2009 and is designed to meet any knowledge representation need.

For most of the data on the Semantic Web the semantic restrictions supported by RDFS are sufficient for representing the information. That is to say that most RDF data is constrained by some collection of RDFS-based vocabularies (these will be discussed in more detail in section 2.2).

A much smaller part of the data produced by the Semantic Web initiative employs OWL for more elaborate knowledge representation needs, but the reuse of multiple complex OWL-based ontologies is non-trivial. The ontologies that are used in chapters 2 and 5 are represented using OWL, but their OWL-specific details are not relevant to discuss in the context of this dissertation. Section 2.2 contains an extended discussion on the roles that RDF, RDFS and OWL play in the current developments within the Semantic Web initiative.

This dissertation focuses on RDF as a universal data model for the exchange of semantically meaningful data. There are two important reasons for this. First, a significant number of data sources used in this dissertation is exposed as RDF and therefore it makes sense to reuse existing data sets as-is instead of converting them to some other arbitrary data model. Second, in the context of this dissertation it is crucial to integrate multiple data sources into a single model. For example, chapter 3 will model both tags and ontologies using RDF and integrate them. This is relatively straightforward to accomplish using RDF when compared to other approaches such as XML or a relational database.

In summary, RDF provides a universal data model for the Semantic Web. The use of RDF allows for the possibility to share, interlink and reuse ontologies, vocabularies (discussed in section 2.2) and reference repositories (discussed in section 2.3). All other Semantic Web languages, e.g. RDFS and OWL, are based on the same RDF-foundation.

2.1.3 Ontology learning & enrichment

Previous sections have assumed the existence of various types of ontologies, but it is important to understand how they are constructed and, more importantly, updated in light of new developments in their respective domain. It has become relatively easy to create an ontology

[11] http://www.w3.org/TR/owl-features/
[12] http://www.w3.org/TR/owl2-overview/

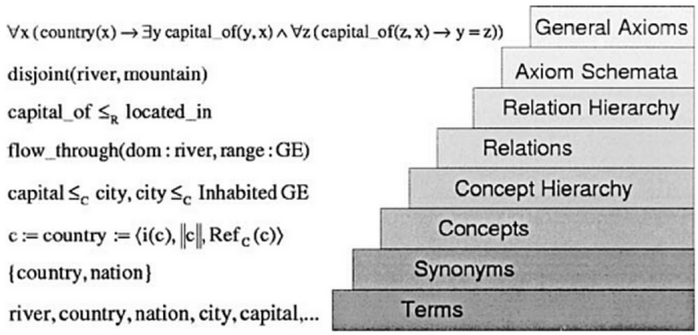


Figure 2.1: Ontology learning cake. The figure has been taken from (Cimiano, 2006).

using software tools[13]. These tools allow users to create an ontology using a user friendly interface and then export this ontology to a standardized RDF serialization format.

Initially the challenge was to actually manually create domain ontologies, but as ontologies grow their complexity increases and so does the maintenance effort. In the long term this leads to a proliferation of high quality but limited ontologies which slowly become obsolete. There are various reasons for ontologies to become obsolete, e.g. the use of outdated formats or lack of adoption. A more pressing reason is the issue of so called "conceptual dynamics" (Hepp, 2007). Conceptual dynamics concerns the shifts and changes in a knowledge domain over time. Conceptual dynamics is about the interaction between the community that uses the knowledge and the artifacts of that community such as ontologies or dictionaries. Changes in the way that knowledge is used or expressed within each community should be reflected in their artifacts.

Figure 2.2 gives a very abstract view of the ontology development process. It starts out with the *initial engineering lag* which results in a first conceptualization of the domain. This version is then optionally followed by a maintenance lag where errors are corrected and new important additions to the domain conceptualization are made. This process may reiterate for an undetermined number of cycles, but it is without question that there will always be some gap between the knowledge embodied by a community and the knowledge captured by a domain ontology. The gap between the community and the ontology is not solely determined by the number of concepts. An ontology that is complete with respect to its coverage of concepts might lack relations or lexicalizations. Figure 2.1 shows the different types of information which can be modeled in ontologies. The different layers have been placed in order of complexity.

Automatic ontology enrichment, the addition of new relations, concepts and lexicalizations to an existing ontology , is therefore a useful addition to manual ontology maintenance. It automatically reduces the gap between the domain ontology's conceptualization and that of the relevant community. Ontology enrichment requires data in addition to the domain ontology itself that reflects the community's current understanding of the domain. These data can

[13] Well-known tools for creating ontologies and vocabularies include Protégé (Gennari et al., 2003), Moki (Ghidini et al., 2009), Semantic MediaWiki (Krötzsch et al., 2006), Neologism (Basca et al., 2008), NeOn Toolkit (Haase et al., 2008) and TopBraid Composer

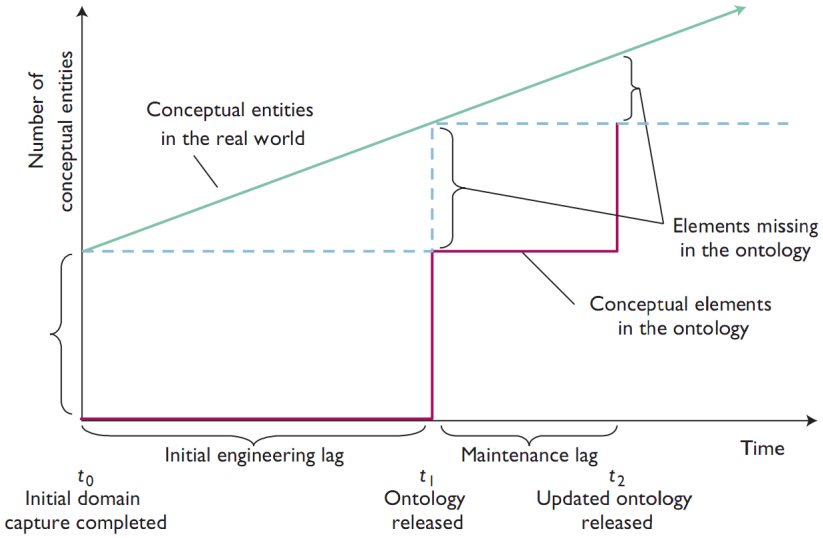


Figure 2.2: "Conceptual dynamics and ontology content and coverage. During the ontology-building process, new conceptual elements become relevant in the domain of discourse, which aren't included in the initial domain capture."(Hepp, 2007) The figure has been taken from (Hepp, 2007).

be used to enrich a domain ontology in combination with reference repositories. In chapter 3, an ontology enrichment methodology is investigated that exploits reference repositories and social media in order to bridge the gap between a domain ontology and a community.

2.1.4 Ontology mapping

The previous section has introduced the topic of ontology learning and enrichment. An important aspect of ontology enrichment that is exploited in chapter 3, is the reuse of pre-existing data on the Semantic Web. However, in order to identify whether data on the Semantic Web is relevant for the enrichment of a particular ontology it is vital to assess whether the concepts are actually the same, even if the URIs that are used are different. Ontology mapping allows one to reliably identify whether concepts from another source are identical in meaning and thus a useful starting point for enrichment.

Ontology mapping is the process of determining whether two concepts from two different sources are identical in meaning. Ontology mapping is an important component of chapters 3 and 5. Integration of information about equivalent concepts ideally happens at the level of the URI. If two concepts from two different ontologies are equivalent then the URI used to denote that concept should, preferably, be equivalent as well. However, in practice, it is common for two ontologies to use two different URIs to refer to the same concept.

For example, one ontology might denote the concept Python, the programming language using the URI `dbpedia:Python_(programming_language)`), whereas another uses the URI `lt4el:Python`. The objective of ontology mapping is to determine whether the two con-

cepts denoted by their URIs are equivalent. The basic idea of concept mapping is known by many names and includes 'alignment', 'merging', 'articulation', 'fusion', 'integration' and 'morphism' depending on the research community (Kalfoglou and Schorlemmer, 2003). Ontology mapping is not restricted to strict equivalences, but subsumption relations between classes are sometimes also included.

One might ask why ontology mapping is worth pursuing at all. I will illustrate the relevance of ontology mapping with an example. Assume that two organizations both formalize the notion of a *computer mouse*. Both organizations use a different identifier for this concept as showing in listing 2.3 on lines 4 and 7.

```
1  PREFIX org1: <http://organisation1.com/ontology#>
2  PREFIX org2: <http://organisation2.com/anotherOntology#>
3
4  org1:ComputerMouse   skos:prefLabel         "mouse"@en
5  dbpedia:Logitech     dbpedia:manufactures   org1:ComputerMouse
6  dbpedia:Logitech     rdf:type               dbpedia:Company
7  org2:mouse           rdfs:subClassOf        org2:PointingDevice
```

Listing 2.3: An RDF-fragment describing information about computer mice.

In listing 2.3 we have two relevant ontology namespaces: `org1` and `org2`. The statement on line 7 allows us to derive that `org2:mouse` is a kind of `org2:PointingDevice`.

However, given this RDF-fragment, we cannot infer that `org2:mouse` is manufactured by `dbpedia:Logitech`. This is due to the fact that there is no statement in listing 2.3 that states that `org1:ComputerMouse` and `org2:mouse` denote the same concept. Intuitively the information associated with `org1:ComputerMouse` can be combined with the information associated with `org2:mouse` by formally stating that the `org1:ComputerMouse` and `org2:mouse` URIs refer to the same concept, i.e. they are interchangeable. The way in which this information is represented using RDF is presented in an extended example in listing 2.4.

```
1  PREFIX org1: <http://organisation1.com/ontology#>
2  PREFIX org2: <http://organisation2.com/anotherOntology#>
3
4  org1:ComputerMouse   skos:prefLabel         "mouse"@en
5  dbpedia:Logitech     dbpedia:manufactures   org1:ComputerMouse
6  dbpedia:Logitech     rdf:type               dbpedia:Company
7  org2:mouse           rdfs:subClassOf        org2:PointingDevice
8  org1:ComputerMouse   owl:sameAs             org2:mouse
```

Listing 2.4: An RDF-fragment extended with an additional statement reflecting the output of ontology mapping.

Listing 2.4 shows the same RDF-fragment as in listing 2.3, but extended with one additional statement on line 8. This statement explicitly states that the two concepts are equivalent using the `owl:sameAs` relation. This statement enables automatic reasoning software to combine the information available for each of the two concepts.

Because of line 8 in Listing 2.4, the concept `org2:mouse` can reuse the preferred lexicalization of `org1:ComputerMouse` on line 4. Additionally, it also allows one to derive that `dbpedia:Logitech` is a manufacturer of `org2:mouse`, because of line 5. Finally, the concept `org1:ComputerMouse` is also affected, because it can now be identified as a type of `org2:PointingDevice`.

Ontology mapping is a process that automatically generates equivalences between concepts using the structure of the ontology, lexical overlap, a domain corpus or a combination of these. Through these equivalences new information from another ontology about a concept can be unambiguously integrated.

2.2 Linked Data & vocabularies

Although the Semantic Web has a compelling vision, widespread adoption of its principles and techniques has been slow. In 2007 the W3C therefore launched a new initiative, now referred to as *Linked Open Data*, that aims to promote the use of Semantic Web concepts and techniques. The *Linked Data*, *Linked Open Data, LOD* (LOD) or *Web of Data* initiative is often conflated with the Semantic Web. It is important to understand where they diverge and overlap. The LOD initiative aims to create large amounts of interlinked and well-standardized data for use within the Semantic Web. "The term Linked Data refers to a set of best practices for publishing and connecting structured data on the Web. These best practices have been adopted by an increasing number of data providers over the last three years, leading to the creation of a global data space containing billions of assertions - the Web of Data" (Bizer et al., 2009a, p.1).

The LOD community emphasizes a bottom-up strategy for publishing existing datasets as RDF supported by widely adopted vocabularies (explained in more detail later in this section). Whereas the traditional focus of the Semantic Web community is on the development of interoperable knowledge models (ontologies) that computers can use to perform advanced reasoning and consistency checks. LOD, instead focuses on creating large **interlinked** datasets published as RDF that present information in a uniform and semantically interoperable manner, i.e. Linked Data generally consists of large amounts of simple information, whereas Semantic Web ontologies generally consist of small amounts of complex information (Falconer et al., 2007). In this dissertation I will assume that both LOD and the top-down Semantic Web approach are required in order to realize the vision of a Semantic Web.

A 5-star rating was proposed in 2010 by Tim Berners-Lee to differentiate between different levels of adoption of Linked Open Data[14] ideas. This 5-star rating emphasizes an incremental adoption of the principles of LOD for situations where adoption of the complete Semantic Web stack is not feasible. A summary of the requirements for each star is listed in table 2.3. A larger number of stars corresponds to better adherence to the principles of Linked Open Data described in table 2.4.

Creating Linked Open Data is relatively straightforward and consists of four principles[15], listed in table 2.4. These four principles are a more precise and technical interpretation of

[14] `http://www.w3.org/DesignIssues/LinkedData.html`
[15] `http://www.w3.org/DesignIssues/LinkedData.html`

⋆	Available on the Web (whatever format) but with an open license, to be Open Data
⋆⋆	Available as machine-readable structured data (e.g. MS Excel instead of image scan of a table)
⋆⋆⋆	as (2) plus non-proprietary format (e.g. CSV instead of Excel)
⋆⋆⋆⋆	All the above plus, Use open standards from W3C (RDF and SPARQL) to identify things, so that people can point at your stuff
⋆⋆⋆⋆⋆	All the above, plus: Link your data to other people's data to provide context

Table 2.3: 5-star rating system for expressing the interoperability of Open Data

the 5-star rating scheme in table 2.3. It is important to note that these four Linked Open Data principles are really nothing more than a compact summary of RDF.

1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL)
4. Include links to other URIs. so that they can discover more things.

Table 2.4: Four main requirements for Linked Open Data as proposed by Tim Berners-Lee

The backbone of the success of LOD, which is not explicitly mentioned in tables 2.3 and 2.4, is the increased reuse of URLs and the consistent use of the same *vocabularies*. RDF vocabularies contain only a very limited, but detailed set of relationships between concepts when compared with a domain or upper level ontology. RDF vocabularies do not enforce a conceptual structure on an ontology or dataset that reuses them.

Vocabularies are sometimes derived from existing annotations schemes for metadata such as Dublin Core[16]. These vocabularies establish common identifiers, such as for expressing the author of a work. Without such a vocabulary authors might be referenced as 'author', 'writer' or 'auteur' in different datasets. Another example of a vocabulary is FOAF[17] which standardizes the vocabulary used to denote online social relationships between agents. Vocabularies are thus simple RDF-standards that promote consistency and the interlinking of data without enforcing a conceptual structure. Vocabularies play an important role in chapters 3, 4 and 5.

Vocabularies and shared URIs can then be employed to interlink various ontologies and data sets leading to a web of Linked Open Data. Figure 2.3 contains a visualization of part of the

[16] http://dublincore.org/documents/dc-rdf/
[17] http://www.foaf-project.org/

available Linked Open Data. Most, if not all, of these datasets are available in the form of RDF.

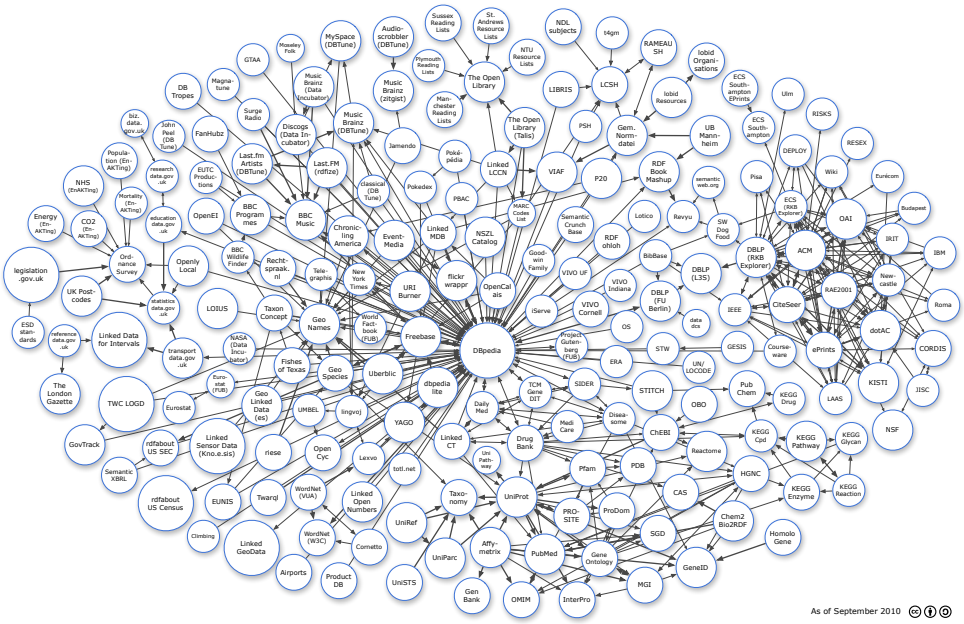


Figure 2.3: Linked open datasets published at CKAN at 22-09-2010. Linking Open Data cloud diagram, by Richard Cyganiak and Anja Jentzsch. `http://lod-cloud.net/`

In summary, the Linked Open Data initiative emphasizes simple principles and a bottom-up approach towards the creation of large amounts of machine interpretable data. Linked Open Data emphasizes the use of simple shared vocabularies for interlinking data in order to prevent the complex knowledge representation issues associated with ontologies.

The Linked Open Data initiative has already resulted in a Web of Data, in which some parts of the Web of Data function as hubs that interconnect many sources of data. An important example of such a hub, i.e. a frequently shared or linked set of data, are the *reference repositories*. The largest dataset in the center of fig. 2.3, DBpedia (discussed in section 2.3), is a widely used reference repository.

The following section will look in detail at reference repositories, and more specifically the DBpedia reference repository, because of the pivotal role that they play in LOD.

2.3 Reference repositories

The previous section has introduced the general idea of Linked Data and its focus on the creation of vast amounts of well standardized, but conceptually simple, data. An important aspect of Linked Open Data is the use of links to connect diverse data sets. Some of these data sets play a central role in the resulting Web of Data, i.e. the identifiers (HTTP URIs)

are frequently reused by other data sets. More specifically, this section will introduce the reference repository as a source of concepts that can be reused and linked to by other data sets.

Reference repositories consist of large collections of resources (concepts) with a minimal amount of conceptual structure, i.e. a large collection of semi-structured word senses with some amount of metadata. The concepts in reference repositories are referred to as *reference concepts* (Bergman, 2010): a stable identifier for information with the aim of interlinking data without enforcing a larger conceptual scheme. The primary goal of a reference repository is to provide *stable referents for a large collection of entities*.

Reference repositories can be populated by extracting RDF data from semi-structured information sources such as Wikipedia. They are also usually rich in metadata about the reference concepts that they contain. This includes, amongst others, biographical information about famous people, demographics of countries and cities, connections between mathematical systems and word senses for various acronyms.

Reference repositories do not enforce a conceptual scheme, such as ontologies, and therefore lack detailed structural information about the concepts that they contain. "As a repository of reference concepts, it is extremely rich. But the organizational structure is weak" (Bergman, 2010). However, the concepts are represented in a way that is compatible with upper-level and domain ontologies. Reference repositories, such as DBpedia, usually contain a huge amount of concepts compared to domain or upper level ontologies.

A concept from a domain ontology is frequently specific to that ontology, i.e. it cannot be used "beyond the boundaries of their originating context, though they are ubiquitously accessible" (Simperl, 2009, p.906). Most ontologies are "rarely built to be shared or reused" (Simperl, 2009, p.906). In contrast, the use of a reference concept in other ontologies is actively encouraged. The reasons for doing this are that it increases the use of shared identifiers, i.e. promotes interlinking data, and is not tied to a specific conceptualization. Some examples of reference repositories are DBpedia[18], Freebase[19], Yago[20], GeoNames[21] and WordNet (Miller, 1995). DBpedia is the reference repository of choice in this dissertation. Reference repositories play an important role in chapters 3, 4 and 5.

There are several recommendations for creating a large store of reference concepts. They have been implicitly adopted by current reference repositories such as DBpedia (Bizer et al., 2009b), but are not officially enforced. These guidelines provide insight into the type of information that is available in reference repositories and how it is structured. The following list, based on that from Bergman (2010), summarizes the recommendations:

1. *Stable URL* - Every reference concept or property should have a stable URL from which additional information about the reference concept can be retrieved.
2. *Preferred and alternate label* - A preferred label (preferably multiple languages) for the concept, because the URL alone is uninformative. The *Simple Knowledge Organization System* (SKOS) (Miles et al., 2005) is a set of specifications and standards

[18] http://www.dbpedia.org
[19] http://www.freebase.com
[20] http://www.mpi-inf.mpg.de/yago-naga/yago/
[21] http://www.geonames.org/

that support the use of relatively simple knowledge organization systems such as such as thesauri, classification schemes, subject heading lists and taxonomies. The RDF property `skos:prefLabel`[22] is a good candidate for the representation of a preferred label. In addition the reference concept may be enriched with alternative labels for synonyms. jargon terms, etc. which will all help to link the concept to its use in text or other metadata.

3. Definition or description. Every reference concept should have a description or definition in order to make its meaning unambiguous for human beings.

4. Embedded in a coherent structure - Embedding a reference concept in a lightweight conceptual structure, i.e. 'lightweight knowledge' (d'Aquin et al., 2008) or 'lightweight faceted ontology' (Giunchiglia et al., 2009), of some sort allows for automated discovery of super concepts, instances of concepts or otherwise related concepts and properties. These small bits of local structure would only have to be *locally consistent* and should not have to constitute an actual ontology which is usually enforced to be *globally consistent*. The difference between locally and globally consistent will be explained in more detail below.

Global consistency and local consistency (Bao et al., 2006) can be used to illustrate the difference between domain ontologies and reference repositories. Imagine a domain ontology about cars which has a deep hierarchical structure. This domain ontology would support making deep inferences such as that 'Ford T' is a type of motorized vehicle. Now imagine a reference repository containing similar information. This reference repository consists of several 'clumps' of information centered around concrete instances. For example, that 'Ford T' is a car and that 'Henry Ford' was its inventor, but not necessarily that a car is a motorized vehicle, i.e. it lacks a comprehensive conceptual structure. The notion of locally consistent, as used by Bao et al. (2006), applies to individual ontologies, but I extend its use towards fragments related to a particular concept from a reference repository. Each of these fragments can be viewed as a 'lightweight faceted ontology' (Giunchiglia et al., 2009).

A globally consistent domain ontology is able to make inferences about concrete instances and abstract concepts by virtue of its rigid well defined structure. It is easy to identify the common abstract concept of two concrete instances in an ontology manually designed in the form of a hierarchical scheme. Such inferences, however, are a lot more prone to errors in a reference repository due to the inclusion of relations with a more auto-associative character[23]. The reference repository on the other hand is not able to make such inferences (reliably) (Ji et al., 2011; Töpper et al., 2012), but, because of its simpler structure, is easier to automatically generate and maintain thereby decreasing the effort needed to create and maintain it. This allows reference repositories to be far larger and broader in scope than domain ontologies which have to be globally consistent.

In summary, reference repositories contain large amounts of reference concepts and associated metadata such as lexicalizations. They play an important role with respect to Linked

[22] `http://www.w3.org/TR/skos-reference/`

[23] Auto-associative connections refer to relations which have been established by an automated process. These usually do not correspond to a specific conceptual scheme and frequently arise from co-occurrence patterns in a (domain) corpus.

Open Data, i.e. reference repositories are hubs that connect diverse datasets. The large amount of structured information that is provided by reference repositories can be harvested for many applications. It is for this reason that reference repositories play a central role in chapters 3 to 5 in this dissertation. The next section highlights one specific reference repository, DBpedia, which is the actual reference repository that is used in this dissertation.

2.3.1 DBpedia

Different reference repositories are available in the Semantic Web. This section will focus on DBpedia in particular because of its size, broad scope and the fact that it is an important integration point for the Semantic Web as previously shown in figure 2.3. DBpedia is used in most chapters of this dissertation. It is used in chapter 3 for ontology enrichment, in chapter 4 for disambiguation and in chapter 5 for semantic search.

DBpedia (Bizer et al., 2009b) is a well-known example of a reference repository that follows all of the aforementioned guidelines. It is automatically extracted from the collaboratively created online encyclopedia Wikipedia using information extraction. The DBpedia extraction software defines a large amount of templates which map a supported table or list of items (in Wikipedia jargon referred to as 'infoboxes') to specific RDF triples. The Wikipedia community has standardized certain types of information such as demographics of countries or biographical information on notable persons, etc. This allows the information extraction software from DBpedia to generate large amounts of structured information on these domains.

Every Wikipedia article results in a so called *DBpedia resource* and the metadata that the extraction software was able to gather from the Wikipedia article. A DBpedia resource is a specific type of reference concept. The metadata of a DBpedia resource includes, but is not limited to:

- The title of the resource and alternative lexicalizations (generated from Wikipedia redirects) in all Wikipedia-supported languages where the same article is available.
- Disambiguation links linking ambiguous terms to possible articles
- Partial ontology mappings from and to DBpedia resources. This includes links to YAGO, OpenCyc, etc.
- Category membership of individual articles and hierarchical information on nested categories
- Local links between DBpedia resources (wikilinks) and links to external resources
- Ontological properties between DBpedia resources if the particular infobox extractor supports it.

A DBpedia reference concept usually has multiple lexicalizations (labels) attached to it which have been derived from the Wikipedia encyclopedia. This information includes *disambiguation pages* which list possible interpretations for a given term and *redirects*, which are terms that automatically redirect to some other preferred term. Redirects usually cover spelling

Type	# Statements
Disambiguation	551726
Category	13653684
Portal-related	1420415
Meta-page	29507
Redirects	5074113

Table 2.5: Occurrences of different types of relations between DBpedia resources derived from Wikipedia articles and other types of resources in the DBpedia version 3.7 Pagelinks dataset.

variations and abbreviations. Disambiguation pages provide additional (ambiguous) lexicalizations for the reference concept which are not preferred lexicalizations. In order to get a sense of the scope of the DBpedia reference repository table 2.5 includes some statistics of its size.

The large amount of reference concepts makes DBpedia suitable for the enrichment of other ontologies. More specifically, in chapter 3 the DBpedia reference repository is used as a source of new domain concepts, lexicalizations and relations. The DBpedia reference repository, thanks to its large amount of disambiguation links, redirects and pagelinks between DBpedia resources, is ideally suited for unsupervised disambiguation (for details see chapter 4). Reference repositories also play a central role in semantic search, presented in chapter 5, more specifically in the semantic analysis of keyword-based search.

Graph density The DBpedia reference repository is an important resource that is used throughout this dissertation. In this paragraph I will look at one specific property of DBpedia: its *graph density*. The notion of *graph density* will re-appear in chapter 4 in order to substantiate some of the design choices for a disambiguation algorithm.

DBpedia contains a large amount of untyped relations between its reference concepts referred to as 'wikilinks'. These wikilinks together establish a large graph of interconnected concepts. A direct wikilink indicates a relationship of some kind between two reference concepts. However, it is not necessarily the case that an indirect wikilink, i.e. a path of several wikilinks, also conveys information about the relatedness of two reference concepts. I will illustrate the issue of establishing an accidental associative relation in DBpedia using the notion of *graph density*.

Graph density can be calculated using eq. (2.1) where C are the concepts (nodes) in the graph and E the set of edges. This measure of graph density ranges between 0 and 1. A density of 1 means that a graph has an edge between every conceivable pair of nodes, i.e. the graph is complete. An undirected graph can have at most $|C|(|C|-1)/2$ edges where $|C|$ is the number of concepts (nodes) in the graph. Dividing the number of edges E in the graph by the

theoretical maximal amount of edges yields eq. (2.1):

$$\frac{2|E|}{|C|\,(|C|-1)} \tag{2.1}$$

The density of the DBpedia version 3.7 wikilink data is 0.000002707 which may not seem dense, but does mean that a DBpedia resource on average has about 13 wikilinks to other DBpedia resources. This density makes it relatively simple to find a short path in the DBpedia graph that connects two random, unrelated, DBpedia resources. Table 2.6 shows the distribution of shortest paths for a sample of 1000 randomly generated concept pairs.

Path length	Percentage
1	0%
2	0,8%
3	4,9%
4	23,3%
5	43,5%
6	25,7%
7	1,7%
8	0,1%
9	0%

Table 2.6: Shortest path distribution for 1000 randomly generated concepts pairs for undirected edges in DBpedia wikilinks. Edges related to external files, portals, categories, disambiguation pages, users and talk-pages have been pruned from the graph beforehand.

The density of the wikilink edges between concepts in DBpedia makes it trivial to establish sparse links between concepts which are only remotely related or not related at all. According to table 2.6 it is meaningless to consider paths longer than six wikilinks, because in that case every concept can be related to any other concept. This characteristic of the DBpedia reference repository is important to note, because it, in part, invalidates the use of path-based measures of word semantic similarity (Sinha and Mihalcea, 2007). The issue of DBpedia's graph density is revisited in chapter 4 section 4.2.

In summary, DBpedia is a large reference repository automatically extracted from Wikipedia. It contains various metadata about its reference concepts including lexicalizations, associative links and concept properties. DBpedia is used in most of the chapters in this dissertation and plays a central role in ontology enrichment, disambiguation and semantic search.

2.4 Conclusion

This concludes the overview of the Semantic Web and those resources and concepts that are relevant in the context of this dissertation. More specifically, domain ontologies and

the DBpedia reference repository will play a central role in chapters 3 to 5. RDF is the primary language used to store and access information in domain ontologies and reference repositories. Subsequent chapters will assume that the reader now has a basic understanding of RDF and ontologies.

Vocabularies and reference repositories do not impose much conceptual structure and are usually expressed using RDF-Schema. Upper level and domain ontologies do enforce a conceptual structure and are commonly expressed using OWL, because it allows for more constraints to be specified on concepts and relations. Domain ontologies frequently incorporate other established RDF vocabularies such as FOAF or SKOS for describing simple domain invariant properties and relations.

The various types of ontologies discussed in this section each have different applications. Reference repositories provide semi-structured metadata on a concept, but cannot match the high quality taxonomic structure of a domain ontology. Domain ontologies are great tools for getting a domain overview and insight into how domain concepts and jargon are related, improve information retrieval and support reasoning by computer agents. Manually designed ontologies can disclose information agreed upon within an expert community and serve as excellent support for users as will be shown in chapter 3.

Reference repositories frequently originate from dynamic socially maintained resources such as Wikipedia. Domain ontologies also have a social origin, the knowledge experts of a Community of Practice, but lose this social link as the community progresses and the ontology is not updated to reflect this. Chapter 3 describes the process of *Social Ontology Enrichment* which re-establishes the link between the community vocabulary and respective domain ontology. A domain ontology is enriched by adding new relevant concepts and alternative terms for concept by identifying them in a community's vocabulary in Social Media. Social Ontology Enrichment thus re-establishes the social context of knowledge as formalized by a domain ontology.

3 Social Web

3.1 Introduction

The previous section about the Semantic Web has emphasized the technical infrastructure required to capture semantics. However, the Web is not just a place for the exchange of objective data, it is also used by people to communicate with each other, share information and collaborate on projects. This section will focus on these social aspects of the Web and explore their origin and current usage trends. More specifically, it focuses on the collaborative behavior of users and how it affects the design of information systems.

Tim Berners-Lee stated early on about the Internet that "the original driving force was collaboration"[24]. Early versions of web browsers, such as Nexus/WorldWideWeb from 1990[25], had editors built into them to facilitate the collaborative creation of content. However, content

[24] http://www.w3.org/1998/02/Potential.html

[25] http://www.w3.org/People/Berners-Lee/WorldWideWeb.html

on the Internet in the early days up to the around 1998 largely relied on websites providing content and surfers consuming it[26]. The so-called Web 1.0 was largely content-driven with little to no interaction through the use of websites. Around 1998 websites progressed to better web frameworks and moved towards participative applications.

This transition to participative web applications transformed the way users and websites interacted. Internet users are no longer mere consumers of information. They increasingly create content themselves and influence the way content is managed in large scale information systems. This transition reintroduced some of the original vision of a social interactive web where surfers are not mere consumers, but also creators of content. The transition was coined 'Web 2.0' in popular media. During the transition to Web 2.0, websites shifted from a content-driven approach towards a symbiotic relation between the users of a social website and the site itself. The famous computer book publisher and early Internet entrepreneur Tim O'Reilly has an insightful comparison of this transition[27] as illustrated in table 2.7. It lists a number of websites and techniques and specifies which collaborative-focused competitor replaced it. Many of these websites and techniques are still popular today (2013).

Web 1.0	Web 2.0
DoubleClick	Google AdSense
Ofoto	Flickr
Akamai	BitTorrent
mp3.com	Napster
Britannica Online	Wikipedia
personal websites	blogging
evite	upcoming.org and EVDB
domain name speculation	search engine optimization
page views	cost per click
screen scraping	web services
publishing	participation
content management systems	wikis
directories (taxonomy)	tagging ("folksonomy")
stickiness	syndication

Table 2.7: Transition examples of Web 1.0 to Web 2.0. Table taken from `http://oreilly.com/web2/archive/what-is-web-20.html` , accessed 20-06-2011

3.2 Types of Social Media

The notions of Web 2.0 and Social Media are frequently conflated. For example, some of the websites listed in table 2.7 are considered to be examples of Social Media, whereas others are

[26] `http://www.w3.org/People/Berners-Lee/1996/ppf.html`
[27] `http://oreilly.com/web2/archive/what-is-web-20.html`

not. In order to understand what Social Media is and how it relates to Web 2.0 it is relevant to identify its defining properties. "Social Media is a group of Internet-based applications that build on the ideological and technological foundations of Web 2.0, and that allow the creation and exchange of User Generated Content" (Kaplan and Haenlein, 2010, p.61). "User Generated Content can be seen as the sum of all ways in which people make use of Social Media." (Kaplan and Haenlein, 2010, p.61)

The 'socialization' of the Web has three important aspects ranging from content sharing to complex social networks. Each of these three aspects of Social Media is still in active use today and boundaries between them are sometimes blurred:

1. Sharing and recommendation. Users share and rate content either on the website itself or via hyperlinks to other sites. Users come across new content through recommendations based on previous preferences and search. Users annotate the resources that they wish to share with short terms (tags) that fill the role of the primary keywords. We will use the word 'term' in a social media context to refer to a broad category that includes words, abbreviations, slang and misspelled words. A term is referred to as a 'tag' when it has been used by one or more users to annotate a resource. Sharing and recommendation systems are relatively easy to create and maintain. Conflicts between users do not have to be managed, because no agreement over tags or content is required. Aggregation over many users will yield popular resources and keywords. Prime examples are social bookmarking websites such as Delicious[28] and Bibsonomy[29], news aggregation websites such as Digg[30] and Reddit[31], and video sharing sites such as YouTube[32] and Vimeo[33].

2. Collaborative content creation: Users cooperate towards a single shared goal. This requires collaboration and conflict resolution using a 'wiki'. Content is usually versioned allowing each individual change to be stored and reversed. A wiki depends on strong social control on content contributions. Wikipedia is perhaps the best example of the potential of wikis for collaboratively creating content. Tutors and learners are quite skeptical of the quality of collaboratively created resources such as Wikipedia (Lim, 2009), but some researchers argue that collaborative resources can actually achieve an acceptable quality level that rivals more established encyclopedias and other sources (Simon, 2010). Moki from Aposdl (Ghidini et al., 2009) and 'Semantic Mediawiki'-derived projects (Krötzsch et al., 2006) are extensions of current wiki-based approaches towards the collaborative creation and maintenance of ontologies and structured data. These applications are largely oriented around the idea of collaborative editing of ontologies using easy to use HTML-based interfaces.

3. Social networks: Each individual is able to create his or her own group of peers in order to create a heavily personalized information sharing and selection system. Social network-based systems are currently mostly used for distributing small amounts

[28] http://delicious.com
[29] http://www.bibsonomy.org/
[30] http://digg.com
[31] http://reddit.com
[32] http://youtube.com
[33] http://vimeo.com

> of information at a time called either ‘tweets’ or ‘status updates’ in natural language resulting in a time ordered ’stream of content’. These status updates are frequently very short, but often contain pointers to external websites for a more extensive elaboration. Well-known examples of social networks with this type of functionality are Facebook, Hyves and Twitter. Although most social networks offer additional functionality like sharing photo albums or discussion forums their core functionality is still personalized information dissemination through status updates (Naaman et al., 2010).

Kaplan and Haenlein (2010) propose a more detailed categorization of Social Media. More specifically, their categorization distinguishes “collaborative projects, blogs, content communities, social networking sites, virtual game worlds, and virtual social worlds.” (Kaplan and Haenlein, 2010, p.59).

Social Media applications today are accessed by billions of users that actively participate in the creation of textual and visual content, providing tags to describe the resources they have contributed. Learners are also starting to use Social Media applications for learning purposes and consequently are gaining relevance in the educational field. For example, the Massachusetts Institute of Technology has a channel for posting videos on YouTube that has 94,564 subscribers and the channel page has been viewed 2,850,311 times[34]. On YouTube, there are videos of lectures given at top universities that have more than 50,000 views. These figures suggest a wealth of information present in modern Social Media that can be exploited for educational purposes. In chapter 3, the community vocabulary extracted from social media is used to enrich ontologies and is subsequently used in chapter 5 to improve information retrieval.

In summary, Social Media employs the concepts and technologies associated with Web 2.0 in order to create and exchange User Generated Content. The next section will focus on one particular aspect of Social Media that plays a central role in this dissertation: the use of social tagging to share and retrieve content.

3.3 Social tagging

Every modern web browser supports storing bookmarks of websites such that the user can easily return to the content directly. Bookmarks are frequently used to build a list of useful pages that the user would like to read at a later time. These are either large sites that the user needs to consult frequently or shortcuts to frequently accessed pages. Social bookmarking allows users to share bookmarks with others, annotate resources with free-form keywords, called tags, and search within their own bookmark collection, or those of others. Tags are the backbone of social tagging:

> Social tagging is fundamentally a method of organizing objects for later use. It is a process of encoding objects with keywords so as to later retrieve those very same documents. This retrieval could be done by the same person that encoded the object, or could be done by other users of the system. (Chi and Mytkowicz, 2008)

[34] http://www.youtube.com/user/MIT, accessed: 06-04-2011

A Collaborative Tagging System (CTS) is a platform where users are free to annotate (tag) resources. In most cases multiple users are allowed to tag the same resource. Part of the appeal of Social Media is the fact that it is very lean, pervasive and the amount of data is enormous with hundreds of millions of users. It is possible to reuse existing user contributed metadata as available in CTS or social bookmarking systems for ontology enrichment (for details see chapter 3). Another advantage of using user contributed metadata is that the community vocabulary may actually be detached from the resources that it describes, i.e. analysis of user contributed metadata can uncover terminology and concepts not present in the resources themselves. Through the analysis of tags a community vocabulary emerges. This vocabulary can be linked to concepts (chapter 4) and these concepts can then be used to enrich ontologies (chapter 3) and improve search (chapter 5).

The tags that are added to a specific resource by a specific agent are referred to as a *Tagging activity* or Tagging action (Kim et al., 2008b). Tags have been determined to better represent users' interests than keywords extracted from the web page (Szomszor et al., 2008) and search and recommendation systems that exploit social bookmarking can achieve higher user satisfaction and precision (Hotho et al., 2006a). In this dissertation I use 'tag' to refer to a *term in a social context*. Once a tag is removed from its social context it is referred to as a *term*.

Social bookmarking websites, a prominent example of the use of social tagging, have been surprisingly popular with examples like StumbleUpon and Delicious, but also implicitly include newer micro-blogging systems like Twitter. An important characteristic of the use of tags in Social Media is their ability to categorize content. Tags are keywords attributed to resources such as documents[35], videos[36], presentations[37] or descriptions of people[38]. It depends on the platform whether the choice of which tags to use is completely free or is limited to some prescribed list. A user wanting to 'tag' a web page about cooking pasta will bookmark the web page and then add one or more tags. The web page about cooking pasta might receive tags such as: 'cooking', 'pasta', 'kitchen', 'water' and 'basics'. It is interesting to see different types of tags linked to the resource. These range from keywords present in the resource ('cooking', 'pasta', 'kitchen', 'water') to user specific categorizations ('basics'). This same trend was observed by Golder and Huberman (2006); Szomszor et al. (2008); Bischoff et al. (2008).

> As the size of our information systems have expanded, there has been a gradual trend from centrally organized systems based on controlled vocabularies (i.e., a library model) to chaotic, ad-hoc distributed systems with many cooperating participants. This spectrum represents a trade-off that must be made between how much effort is required to make a single annotation and how much of the data is annotated. (Heymann and Garcia-Molina, 2006, p. 1)

The metadata that gets added to the resources is different for each social network, because each has its own target audience and protocols. For example, in the social bookmarking

[35] http://www.scribd.com/
[36] http://youtube.com
[37] http://www.slideshare.net
[38] http://www.linkedin.com

network Delicious, members can attach tags, a description and a title to their bookmarks, whereas in YouTube, which is primarily a video-sharing website, users can only (dis)like videos or add a comment, without having the possibility to add additional metadata to the resource using tags[39]. The tags and other metadata are in this case solely added by the uploader of the video. For non-textual documents, like videos on YouTube or Vimeo and images on Flickr, searching entirely depends on the metadata that have been entered, i.e. tag(s), description and title. The metadata attached to the resources for non-textual content is of vital importance for the retrievability of that content at a later time, since the metadata constitute the only textual information about the resource. Bad quality or no metadata, results in a low retrievability of content. Voss (2007) interestingly conceptualizes the popularity of tagging as an indicator of the revival of manual annotation in an era of largely automated text analysis. This observation was later further substantiated by Szomszor et al. (2008):

> "In a study from Yahoo! on the del.icio.us data, Li and colleagues found that tags are better representatives of users' interests than the keywords of the tagged Web pages, because (a) they offer a higher level of content abstraction, and (b) they are better representations of the user's perception of that content." (Szomszor et al., 2008)

The changes in the tags used to annotate resources also depend on a user's background knowledge and the vocabulary associated with the community that he or she is part of. Interestingly, although tagging capabilities were present in software for an extended period of time, tagging mostly gained popularity in the context of the social web due to social pressure and the advantages gained from community processes (Ames and Naaman, 2007). Another important characteristic of online tagging is that through although individual keywords are used to describe personal objects, a community vocabulary appears over time. This 'shared language' arises "organically through the efforts of many diverse users" (Chi and Mytkowicz, 2008, p. 7) instead of being enforced from the start. Heymann et al. (2008) gathered a corpus of 40 million tagged bookmarks from del.icio.us and observed that the "tags were overwhelmingly relevant and objective" (Heymann et al., 2008, p.205).

In order to analyze the results from social tagging and Collaborative Bookmarking Systems a more precise treatment of social tagging is required, i.e. an unambiguous mathematical conceptualization of the domain. Formally, a tag based social network can be modeled as a graph consisting of users (actors), resources (bookmarks) and tags. This model can be extended to include semantics (concepts). For example, by means of clustering the tags (Mika, 2005). Such a model of users, resources, tags and optionally concepts is referred to as a *folksonomy*:

> "The word 'folksonomy' is a blend of the words 'taxonomy' and 'folk', and stands for conceptual structures created by the people. Folksonomies are thus a bottom-up complement to more formalized Semantic Web technologies, as they rely on emergent semantics which result from the converging use of the same vocabulary. The main difference to 'classical' ontology engineering approaches

[39] up to at least September 2012

Figure 2.4: Example of a tag cloud. Image taken from (Sinclair and Cardew-Hall, 2008)

> is their aim to respect to the largest possible extent the request of non-expert users not to be bothered with any formal modeling overhead." (Hotho et al., 2006a)

In short, a folksonomy is the "emergent labeling of lots of things by people in a social context." (Gruber, 2007, p.2). It is not to be confused with "a taxonomy or even a collaborative categorization"(Gruber, 2007, p.2). The totality of tags from a specific user or group of users is referred to as a *tag space* or *folksonomy tag space* and can be used to summarize content. A popular visualization of a tag space is the 'tag cloud' which visualizes the tags as a cloud where tags with a higher frequency are displayed with a larger font size. Figure 2.4 shows an example of such a tag cloud visualization.

A tag cloud can alternatively be seen as a summary of the subjects of the dialogues within a community or a 'speech genre'[40]. The structure of a folksonomy can be used to improve personalization of content, e.g. provide personal recommendations of new resources and people based on a user profile (Ley et al., 2010; Siersdorfer and Sizov, 2009). Search and recommendation systems that exploit the social structure of a folksonomy can achieve higher user satisfaction and precision than conventional means (Hotho et al., 2006a).

[40] A speech genre is a stable category of utterances that correspond to a typical situation. It includes daily activities as greetings, military commands, conversational norms within (knowledge) communities, etc. Loosely based on http://prelimsandbeyond.wordpress.com/2009/01/02/bakhtin/ accessed: 28-06-2012

3.4 Conclusion

Social Media have transformed the way users interact with content on the Web. Collaborative tagging, annotating and social networks play an important role in the Social Web. They allow a community of users to structure information using a self-constructed vocabulary, i.e. a folksonomy. The adoption of a common vocabulary is attested by the shared use of tags. Information is more accessible and easier to retrieve when a community of users adopts a shared vocabulary. Additionally, the explicit social structure of their respective community can be used for search and recommendation. Social sharing and recommendation, collaborative content creation and social networks have re-established the role of social processes in information exchange.

Another important social phenomenon related to sharing and recommending information is *learning*. Learning concerns the exchange of knowledge between peers, Communities of Practice, teachers and/or tutors. The next section will introduce the subject of learning and the crucial role that social processes play in the learning and development process.

4 Learning and E-learning

The Semantic Web has transformed the way in which machine interpretable knowledge is stored and exchanged. The Social Web has enabled scalable social interaction and supports collaborative problem solving. Both the Semantic and Social Web have several applications. Sections 2 and 3 have illustrated how computer and communication technology have transformed knowledge representation, information exchange and social interaction. This section will focus on one application in particular: learning. In order to understand how the Social and Semantic Web relate to the subject of learning and education it is important to firstly define what learning (section 4.1) is, and secondly how e-learning, i.e. technologically assisted learning (section 4.2), functions. Problems related to current approaches to e-learning will be discussed in section 4.3. The lack of integration with the Social and Semantic Web will get attention in particular. The proposed solution to the problems described in section 4.3 will be discussed in section 5.

4.1 Learning

One could think that someone learns when he or she can reproduce the multiplication table of 4. Learning in this context seems to mean precisely storing certain information with the aim of reproducing this information at a later time. However, learning just the multiplication table of 4 or the names of dinosaurs, without social context or understanding, does not fully constitute 'learning', but only reproduction. It is the ability to abstract from the initial learning situation, to use knowledge correctly in collaboration with peers and to transform and adapt insights to new ones that are essential. Learning can be viewed as a 'transformative process' in the *constructivist* tradition (DeVries, 2000; Palincsar, 1998). The 'transformative process' encompasses the building, revision and use of 'cognitive structures' by individuals. This process is unique for each individual and thus leads to differences in understanding.

Postmodern constructivism moves away from the viewpoint that the individual should sit at the center of this structure building process. Instead, the *social environment* is assumed to provide the *foundation* for individual structure building. A community of knowledgeable peers expects its members to have specific knowledge and the ability to communicate about it. An individual that wants to participate needs to acquire the conceptual knowledge and the vocabulary used by that community. For example, a community about dinosaurs will expect its members to know what a 'Paraceratherium' is and presupposes the ability to explain, using proper terms, how it relates to 'Hyracodontidae'. Learning always takes place within the context of a community, i.e. individuals are always in a 'social context' when learning. The combination of a constructivist perspective on learning and the paramount importance of the social environment is commonly referred to as 'social constructivism'.

Lev Vygotsky, an important contributor to the main concepts of social constructivism made an insightful distinction between the terms 'learning' and 'development' (Palincsar, 1998; Glassman, 1995; John-Steiner and Mahn, 1996; Chaiklin, 2003). This distinction stems from the difficulty when applying the term 'learning' without having a social context or taking individual cognitive structures into account. For example, it does not make sense to 'learn' about dinosaurs without having the right background knowledge or the intention to act in the respective community of paleontologists, i.e. dinosaur experts. Vygotsky defines *development* as the acquisition of skills and knowledge specific to an individual's maturing cognitive structures in the context of a relevant social environment. This distinction allows learning a certain skill or piece of information, without entailing that someone develops further as a knowledgeable individual. According to Vygotsky "the only good learning is that which is in advance of development" (Palincsar, 1998, quoted on p. 353).

Wenger and Snyder (2000) elaborated on the importance of the social environment that Lev Vygotksy stressed. Wenger and Snyder (2000) coined the term *Community of Practice* (CoP) which refers to "groups of people who share a concern or a passion for something they do and learn how to do it better as they interact regularly."[41]. Communities of Practice have three important characteristics: (1) members have a shared domain of interest, (2) members in the community interact and collaborate, (3) members are actual practitioners instead of only having an interest in the domain. Learners (implicitly) belong to a host of CoPs whether in the context of their personal hobbies, work or education. In some CoPs learners are core members whereas in others their role is only peripheral. Over time the relation between certain CoPs and the learner may change as the result of development. A learner that was initially in the peripheral of a CoP may become a core member as the result of increased knowledge. CoPs are increasingly embracing the capabilities of social media in order to support their community's activities. In effect, the level of integration of a particular individual within a specific CoP is also expressed via the social connections to, and content accessed from that online community.

Learning, within the paradigm of social constructivism, is the individual process of structure building mediated by the social environment. The only learning that is worthwhile is that which results in *development* of the individual. Learning is always embedded in a *Community of Practice*, a social environment, that determines what is learned, how the information is

[41] http://www.ewenger.com/theory/ accessed on 2011-11-02. Although the same ideas are presented in a paper (Wenger and Snyder, 2000) the content of the web page expresses the ideas more clearly in my opinion.

presented and how cognitive structures are built.

4.2 E-learning

Books in a library are for many no longer the primary source of knowledge. Books are increasingly replaced by alternatives such as blog articles, online encyclopedia and tweets. Information is also increasingly accessed, shared and filtered through social media (for details see section 3). Important aspects of learning have transitioned to Internet and computer-based technology. These computer-supported approaches to learning are referred to as *e-learning*. E-learning concerns the design and use of computational tools that support the learning process on both the collaborative and individual levels. Computer technology supports novel ways of storing and using knowledge that are not possible with static books. Knowledge models, like ontologies (see section 2 and chapters 3 and 5 for details), help learners access content and support the acquisition of domain knowledge (Westerhout et al., 2010, 2011; Monachesi et al., 2009).

There are various ways of making use of modern technology to support the learning and development process of individuals. Many institutions currently employ simple e-learning tools for their existing courses. Up until at least 2012 these so called *Learning Management Systems* (LMS) are deployed to provide the very basics of online course support. They usually provide functionalities such as a discussion forum, assignment distribution, collection and grading and organizing course content and announcements. Teachers use the LMS to provide students with a learning path and structure information within their course on the Web (Kirkwood, 2009). The support provided by an LMS consists of learning support for an individual student. Examples of an LMS include BlackBoard[42], Moodle[43] and ILIAS[44]. An LMS provides a platform for a community of peers to access a limited set of resources selected by a teacher or tutor. However, most if not all LMS-like systems lack a transition path towards a Community of Practice and are thus, by design, limited to supporting a learner's development as part of their courses but not beyond that limited context.

Computer supported approaches and paradigms which focus on the collaborative aspects of learning are referred to as *Computer Supported Collaborative Learning* (CSCL) (Stahl et al., 2006). Learning in a CSCL-perspective is argued to be primarily a collaborative process between students and tutors instead of a more traditional teacher centered practice, i.e. direct instruction. Collaborative processes are argued to lead to better learning outcomes when compared to isolated learning activities (Cho et al., 2007). CSCL-approaches are becoming increasingly relevant for distance education and learning outside of an institutional context such as a university. In those situations the computer is the *primary* method for collaborating between peers, collaborative problem solving, mediating between tutors and accessing learning objects. Stahl et al. (2006) argues against an exclusive focus on learning with no regard for development or social context. "E-learning is too often motivated by a naive belief that classroom content can be digitized and disseminated to large numbers of students with little continuing involvement of teachers or other costs ..." (Stahl et al., 2006, p. 2).

[42] http://www.blackboard.com
[43] http://www.moodle.org
[44] https://www.ilias.de

The model proposed by Stahl (2006) constitutes a CSCL-based social constructivist learning process (Stahl, 2006). The CSCL-based approach from Stahl et al. (2006) accounts for the different phases of the internalization of collaborative structure building and a model on how one phase leads to another. This model includes the transition from 'personal knowing' to 'collaborative knowing' via individual attention, public utterances, community discourse and shared understanding. Well designed technological support tools that support learning are argued to offer three important types of advantages (Stahl et al., 2006):

1. new forms of communication - Examples include, Instant Messaging, Online Social Networks, online discussion forums and collaborative editing. Collaborative editing mixes the traditionally distinct phases of discussion and content creation. Stahl et al. (2006) emphasizes that "we should exploit technology for its potential to make new interactions possible, not try to force it to replicate face-to-face interaction".

2. activity records - The ability to analyze discussions between learners and monitor development in order to generate feedback that supports the collaborative process. This feedback can be used by students for reflection purposes and meta-cognitive skills. Tutors can employ them to get an overview of student activity. (Rebedea et al., 2010b; Janssen, 2008).

3. adaptive learning objects - Learning texts and objects that adapt their content based on some model of the learner or personalized learning object recommendation (Brusilovsky, 2001; Klasnja-Milicevic et al., 2011)

For collaborative learning to be successful the tools that are used should be familiar to students. The tools also need to be continuously accessible outside of the physical classroom. Collaborative learning logically extends to the increasingly important online world as well. Not only the learning resources themselves can originate on the Internet, but the teaching practice itself is shifting towards online environments. The availability of online social networks, which are continuously available outside of the classroom, has led students to exploit the Internet for learning purposes in a way that is familiar to them. In practice, this means that students share links and documents using social media, mobile applications and instant messaging. There is, however, currently a disconnect between these online activities, collaborative processes and the highly structured learning processes used in the classroom and the institutional LMS.

Historically, people like teachers, tutors and librarians have performed the role of intermediating between high quality expert approved content and the information needs of novices. These intermediates make decisions with regard to what resources should be considered that support the learning and development process of a learner. In informal online learning situations teachers are either unavailable, because a learner is not affiliated with a formal institution (a university) or the learner requires 24/7 assistance that a teacher is unable to provide. Currently, it is unclear how and to what extent the online social and collaborative learning activities should be interwoven with established educational practices (Hung and Chen, 2001).

The integration of the teacher practice of mediating between knowledge sources and peers needs to take the leap towards the dominant online environment used by students; the social

networks and online search and recommendation systems (Dalsgaard, 2006; Marenzi et al., 2008). Integrating the teaching practice with online social networks allows teachers to re-establish their role of knowledge mediators, both in the classroom as well as in the increasingly important online social world. This was an important objective of the iFLSS system (Westerhout et al., 2010, 2011) developed within the LTfLL-project(Berlanga et al., 2009). Additionally, the integration of expert-validated knowledge models with the community's understanding of a domain is the primary motivation for the approach to ontology enrichment (Monachesi et al., 2009) presented in chapter 3.

4.3 Problems with e-learning using Social Media

The previous sections have illustrated the advantages of using computer technology to support learning. Social context is crucial for learning to succeed. Social Media websites are a likely candidate for providing the social context of learning within an online environment. They virtually guarantee a social context for learning, but a fundamental problem remains. It is the crucial difference in the vocabulary used by experts and novices. Novices trying to find information created by experts will fail, because they do not know the right terms to search with. In effect, material which might actually be useful and of high quality is not accessible, because a different vocabulary is used by the respective Community of Practice. Additionally, personalization can artificially constrain the relevant resources that one can access.

Individuals employing the wrong vocabulary may get relevant resources, such as a tutorial, for accomplishing a task. However, these resources only require the ability to follow a number of clearly defined steps. Proper understanding of each step is not required. These types of resources are certainly useful for accomplishing a simple task, but from a learning perspective these are detrimental in the long run. The reason is that individuals do not acquire much conceptual knowledge using such resources, but only specific skills. Individuals may be given a false sense of security and they may overestimate their conceptual knowledge and abilities. It is therefore important to transition from step-by-step tutorials to resources if the individual has a long-term interest in the subject matter.

An example of the difficulties that learners face is when searching for 'create website'. The goal of the learner is to acquire abstract concepts like 'markup' and 'stylesheet languages', such as HTML and CSS, in order to gain conceptual knowledge instead of 'mere' skills or 'surface knowledge' (Beattie IV et al., 1997). A query like 'create website' on a popular social bookmarking website, such as Delicious, will yield resources about step-by-step wizards for creating a rudimentary website. Interestingly none of these results are about learning the languages used to actually create web pages (e.g. HTML and CSS). People familiar with the art of creating websites often include terms like 'html' and 'css' in their queries, because that is the established vocabulary in their community. They would not label resources about HTML as suitable for 'creating a website', because this is, from their perspective, trivial and redundant. A query using the terms 'html learn' yields an entirely different set of resources which are more useful from a conceptual learning point of view.

There is thus a problem with users searching for 'create website', because this query does not use the 'correct' vocabulary and thereby excludes high quality resources from the expert community. The abundance of resources in social media drives learners towards fact finding

and surface learning (Beattie IV et al., 1997) with little motivation towards deep learning (understanding, reflection and abstraction). Deep learning in a search context is also referred to as *informational search* (Jansen et al., 2008). Expert communities in Social Media share resources using a self constructed vocabulary which might be inaccessible for learners with insufficient knowledge of this vocabulary (for details see chapter 3). In addition, terminology can have conflicting meanings in different communities thus leading to confusing search results (for details see chapter 5). A learner that does not know the right vocabulary cannot improve his or her search request by revising or adding additional terms.

Although the vocabulary mismatch between learners and communities is important, there is another problem related to the use of personalization and recommendation. In a learning context the focus should be on the acquisition of conceptual knowledge that leads to development. Personalization and recommendation of content based on past interests can be effective for getting additional related resources, but it can offer only limited assistance to learners starting in a new domain of interest. Greedy personalization makes locally optimal choices for recommendation and personalization. It has an important downside, especially in a learning context. Imagine a learner starting in a new domain. Initially, this learner will select low quality/simple resources, because he or she has not acquired the proper vocabulary and domain knowledge yet. A system performing greedy personalization will exploit the content preferences of this new domain in order to recommend and 'improve' search results towards 'more of the same' (low quality/simple). However, this is not always in the best interest of the learner, especially when the learner has progressed beyond his or her initial understanding of the domain.

Development of learners is also obstructed by something closely related to recommendation and personalization. Learners cannot access all relevant resources, because of a self imposed *filter bubble* (Bozdag and Timmermans, 2011). A filter bubble refers to the concept that an individual gets only part of the relevant resources, based on a personal profile created by computer algorithms. Other resources that are outside of the filter bubble may actually be important and relevant, but will not be available, because of our reliance on computer based search and filtering. In a learning context, a filter bubble is the result of past actions during which the learner had less domain knowledge than the learner has now. As a result, content is recommended that overlaps with past preferences. However, resources that lead to conceptual development of the individual should be recommended in a learning context instead of resources that match already acquired concepts.

The effects of a filter bubble can be mitigated by using an enriched ontology (presented in chapter 3) to bridge the lexical and conceptual gap between the expert community and novices. Chapter 6 will show the impact of taking the conceptual development of learners into account when classifying and recommending resources using machine learning. Additionally, chapter 5 provides an approach based on semantic search that can retrieve relevant resources in spite of terminological differences between a user and a resource collection.

5 Integrating the Social Semantic Web and e-Learning

The previous sections have introduced the Semantic Web, The Social Web and E-learning. This section explores the roles that the Semantic and Social web play in supporting learning from a theoretical and practical perspective. In summary this suggests that e-Learning can benefit from the knowledge representation capabilities of ontologies and the social context that social media provide.

An increasing amount of learning activities is taking place outside of the classroom. It is important to understand how these new learning activities are performed and how collaborative processes function on the Internet. A good way to understand these phenomena is to study the use of the Social Web in its different incarnations. How do people form groups on the Internet? How do they exchange resources?

What semiotic means are used to communicate about resources and people? In the light of social constructivism, ontologies can be viewed as the formalization of the conceptualization of the experts within a community. Ontologies can also incorporate the end products of community agreement using the ongoing dialogue within that community as represented by a community's salient tags. The community dialogue and learning process is increasingly shifting towards online Social Media and the exploitation of structured information sources as available within the Semantic Web.

Hung and Chen (2001) identified four important aspects that a web-based learning community needs in the context of situated cognition (Brown et al., 1989) and Vygotsky's role of the social environment in learning and development:

> (1) Situatedness, fostered by: contextualized activities, e.g. tasks and projects based on demand and needs; and implicit and explicit knowledge, e.g. ways of seeing such as beliefs and norms.
> (2) Commonality, fostered by: shared interests, e.g. in books; and shared problems, e.g. in solving programming problems.
> (3) Interdependency, fostered by: varying expertise e.g. differences in knowledge and skills; varying perspectives or opinions, e.g. differences in views on current issues; varying needs, e.g. those who want to gain a reputation and those who want answers; mutual benefits, e.g. to complete a task that is not manageable by any one individual; and complementary motives, e.g. novices get answers from the experts and experts gain reputation from the novices.
> (4) Infrastructure, fostered by: rules, e.g. ratings or points system to motivate participation; accountability mechanisms, e.g. credibility of a contributor's review which is appraised by other members; and facilitating structures, e.g. information architecture facilitating the interdependencies.
>
> (Hung and Chen, 2001)

These four aspects highlight the importance of the community in which the learning activity needs to be embedded. Embedding the community makes the learning process meaningful and exploits learners' internalization of social constructs, through which learners achieve

their potential level of development. It is these same four aspects that are well supported by modern Social Media. Learners are able to address real needs and communicate with potential adopters through Social Media, which lowers the boundary of contacting experts (Chi and Yang, 2010). Social Media make it easier to find peers with shared interests, even for niche areas, and make expert dissemination of information accessible to novices. This again underlines the importance of situatedness when designing systems and especially e-learning systems (Hung and Chen, 2001; Lewis et al., 2010). The processes which determine the development of learners all come down to complex interactions inside and between Communities of Practice. Individuals always play the roles of consumers *and* producers of the content simultaneously through their interaction with each community.

It is important not to underestimate the impact of the unassisted appearance of lightweight community vocabularies in Social Media. Apparently, annotations such as tags in Social Media allow scattered Communities of Practice (knowledge communities) to emerge and align their respective vocabularies. The assimilation of the community vocabulary by individuals is actually an important component of learning and development, where individuals learn to "master a speech genre" (Bakhtin et al., 1986) and integrate within the associated Community of Practice (Wenger and Snyder, 2000) in a situated environment (Hung and Chen, 2001).

The solutions that will be presented in this dissertation for the problems identified with learning and social media are centered around the idea that the vocabulary employed by individuals depends on their level of expertise. The mediation between knowledge sources is historically the domain of teachers, tutors and librarians, but automated systems can provide immediate 24/7 feedback and support, which better fits with an online environment. I have carried out research into automated methods that can (in part) fulfill the role of a teacher. These methods require two important types of information:

1. High quality information regarding proper terminology and or conceptual structure.
2. Availability of an accessible digital social environment.

Regarding the first requirement; ontologies in the Semantic Web are excellent tools for succinctly describing the important characteristics and vocabulary of a domain. Ontologies can help identify whether two terms, although different in form, actually refer to the same or related concept and explain how they are related (for details see chapter 3). As such, a domain ontology can fulfill the role of a teacher or tutor in the sense that it can mediate between a novice and the domain structure as viewed by experts. In order for ontologies to succeed in fulfilling this role it is vital to identify the vocabulary that novices employ and relate it to the concepts that a domain ontology contains (for details see chapter 3). Bridging the gap between the vocabulary of novices and experts using ontologies also addresses problems novices experience using keyword-based search (see chapter 5).

The second requirement refers to the availability of a digital environment that can be accessed and interacted with by automated systems. The digital environment is frequently an online Social Media site. Many learners are already familiar with them and their popularity and scalability makes them suitable for sharing content. Learners can use these tools to search for information using specific terms, a process which can be enhanced using ontologies (for details see chapter 5). Alternatively, content is automatically recommended to them based

on past preferences and or social connections. An important characteristic of many of these social networks is that they enable users to use short terms, i.e. *tags*, for labeling resources of interest.

The first and second requirement interlink at the level of the vocabulary. It is thus important to establish a robust relation between the language employed by novices, the expert vocabulary captured by domain ontologies and the language employed by online Communities of Practice. This can be achieved by embedding knowledge in a social environment and, vice versa, recommending appropriate knowledge in social contexts. Chapter 3 investigates a methodology that achieves these goals of integrating novice and expert domain vocabularies using ontology enrichment. Similarly, automated analysis of the vocabulary employed by learners can provide insights into their learning and development process (for details see chapter 6).

It makes sense for the Social and Semantic Web to converge to a single integrated whole, because knowledge itself is fundamentally social. Just as the semiotic tools, social and individual factors within Vygotsky's theoretical framework form one integrated whole. Ontologies, for example, can be effective semiotic means in the broad sense of Vygotsky that interlink information on the Web in an abstract and useful manner and aid development (Stojanovic et al., 2001). An integration of the Social and Semantic Web leads to a myriad of mutual benefits. This integration ranges from the modeling of the Social Web using Semantic Web techniques (Specia and Motta, 2007; Angeletou et al., 2007; Andrews et al., 2010), extracting ontologies from folksonomies (Damme et al., 2007; Monachesi et al., 2009), identifying social relations and communities (Diederich and Iofciu, 2006) and folksonomy based recommendation of content (Gemmell et al., 2008; Jäschke et al., 2007; Siersdorfer and Sizov, 2009). The aforementioned related work suggests that is thus a lot of potential in the integration of the Semantic and Social Web. This thesis follows this research direction and evaluates it in the context of e-learning.

6 Conclusion

The Semantic Web plays a largely facilitative role in automating the tedious tasks of information management and the maintenance of knowledge models. It also supports the deep integration of various types of information for use both by computer algorithms and users. In parallel, the Social Web situates the community knowledge via a flexible categorization scheme using tags and social recommendation and filtering. Many of the information gathering activities of users shift towards the Social and Semantic Web. This includes search, sharing and recommendation in the pursuit of learning. Integration of the tools and techniques of the Social and Semantic Web re-establishes the inherently social nature of learning and development of learners. The integration of the Semantic and Social Web will result in high quality, user friendly, collaborative tools that assist the development of learners.

Social constructivism stresses the importance of the collaborative nature of learning and development. Social learning environments lead to the collaborative construction of semiotic tools such as folksonomies and ontologies. In order to bridge the gap from collaboratively constructed folksonomies to formal domain ontologies, ontologies need to incorporate novel concepts into their existing conceptual structure and incorporate new lexicalizations that ap-

pear in the community vocabulary for existing concepts. A methodology for accomplishing these goals is presented in chapter 3. Vice versa, tags used in the Social Web are open to additional semantic analysis if appropriate links can be established between ontological concepts and tags as presented in chapter 4.

When integration between the Social and Semantic Web has been achieved, common tasks such as searching for information, presented in chapter 5, can be significantly improved. The integration of the Social and Semantic Web also has important ramifications for (e-)learning and development, in the sense of Vygotksy's social constructivism, because it is important to situate knowledge, formalized through ontologies, in its proper Community of Practice.

Quantitative analysis of the differences, both conceptual and lexical, between novice learners and experts provides insight into an individual's rate of development and learning path.Chapter 6 presents an approach that uses topic models and takes both the lexical and conceptual aspects into account using a unified lexical model.

Chapter 3

Ontology Enrichment

1 Introduction

Improved methods for information retrieval have become crucial in order to locate information. Some of these methodologies make use of ontologies to provide structured access to knowledge and resources. An ontology for a specific domain (a domain ontology) is a formalization of a piece of knowledge of one or more experts of this domain. Because of their clear formal structure, domain ontologies have the potential to facilitate access to information. However, in order to accomplish that, an ontology must be compatible with the resources of a Community of Practice (CoP). At the moment of its creation, a domain ontology[1] is a proper reflection of a community's understanding of a domain and the vocabulary used to communicate about it. The concepts and relations represent the community's understanding and the ontology's lexicalizations represent the vocabulary. With time, the community will develop and new concepts and terms will either augment or replace existing ones. A community's development and change is reflected in its discourse, which is increasingly taking place in Social Media.

The domain ontology, however, does not automatically incorporate the conceptual and lexical changes as they appear in the community's discourse. Without *ontology maintenance* the lexical and conceptual gap of the domain ontology with the community will increase over time. In order to prevent this, *enrichment* of the domain ontology's conceptual and lexical structure is required. Ontology maintenance can be performed manually, but automatic ontology maintenance based on a community's discourse, as it is expressed in Social Media, is preferable, especially because ontology maintenance is a continuous process.

This chapter presents a new approach to automatic ontology maintenance called *Social Ontology Enrichment* (SOE)) (Monachesi and Markus, 2010b) that exploits social media for ontology enrichment. SOE automatically adds new concepts, relations and lexicalizations

[1] Recall that in chapter 2 it was explained that the term 'domain ontology' stands for 'lexicalized domain ontology', unless otherwise mentioned. This is in order to improve the readability of the text and limit the amount of repetitiveness for complex terminology.

to an existing domain ontology. Social media, more specifically, tag-based Collaborative Tagging Systems (CTS), are used as a corpus in order to determine the currently relevant community vocabulary and to relate it to the domain ontology concepts. The community vocabulary is linked to appropriate concepts from a *reference repository*, such as DBpedia. The metadata of a reference repository is used to enrich the domain ontology. New information is embedded in the existing structure of the domain ontology with an appropriate ontological relation extracted from a reference repository. The lexical and conceptual gap, between the ontology and its associated the Community of Practice (CoP) is thus reduced.

The SOE-approach is innovative, because it makes exclusive use of data from Social Media in combination with reference repositories to determine the (new) relevant concepts and their lexicalizations. As a result, SOE is able to automatically add new concepts and ontological relations beyond semantic similarities or cooccurrence rates to a domain ontology. It also exploits *reference repositories* to infer relations and extract metadata about concepts, whereas related approaches use either formal resources such as WordNet or large document collections for relation extraction. SOE achieves high quality ontology enrichment with an accuracy of 0.94 in some conditions. It only adds knowledge that has been validated by an online community.

Section 2 contains a survey of state of the art techniques related to the SOE approach and its components. An overview of ontology enrichment in general and a motivation of the SOE approach is provided in section 3. The full methodology and its components are presented in section 4 in detail. The SOE ontology enrichment process is illustrated with a few examples in section 5. The performance of SOE will be evaluated in section 6. An existing domain ontology about computing is enriched using social data extracted from the social bookmarking site `Delicious.com` and is compared to the original unenriched ontology. Manual evaluation of the enrichment results shows a promising accuracy of up to 0.94. Section 7 provides a summary of the chapter, the main conclusions and the results that have been achieved.

2 State of the art

Ontology enrichment is a rather broad area by itself. It spans a wide range of domains ranging from named entity recognition (Borthwick, 1999), ontology mapping (Noy, 2009; Bouma, 2010), NLP (Faatz and Steinmetz, 2002; Navigli and Velardi, 2006), graph theory and manual ontology maintenance (Damme et al., 2007; Alves and Santanche, 2011).

Ontologies can be enriched using a variety of sources. This chapter focuses one such source in particular: folksonomies that originate in Social Media. This approach by itself makes some strong methodological choices with respect to ontology enrichment. The most important one is that social networks are good sources of domain knowledge. More specifically, social networks capture the current vocabulary and domain concepts of a Community of Practice (CoP) (Wenger and Snyder, 2000).

It is for this reason that the SOE-approach described in this section uses Collaborative Tagging Systems (see chapter 2 section section 3) as the primary source for the identification of new relevant concepts and lexicalizations (Monachesi and Markus, 2010b). This section will

focus on techniques related to social media analysis in the context of ontology enrichment. More specifically, this involves three related areas of research:

(1) tag recommendation (section 2.1)

(2) learning taxonomies from folksonomies (section 2.2)

(3) integration of folksonomies and ontologies (section 2.3).

Methodologies and algorithms from each of these research areas will be used as the basis for the proposed SOE approach to ontology enrichment presented in section 3. The following sections discuss the state of the art with respect to each of these research areas. Section 2.4 describes how SOE compares to the state-of-the-art social media analysis and ontology enrichment techniques and approaches.

2.1 Tag recommendation

In the context of ontology enrichment *tag recommendation* is relevant, because it can be used to extract the most relevant tags from a larger set of tags in a folksonomy. Tag recommendation-techniques allow one to extract tags that are relevant with respect to a specific domain in spite of other domains, and their associated tags, co-existing in the same folksonomy.

The reason why tag recommendation is relevant for ontology enrichment is that Social Media are considered to be a generic dataset spanning an arbitrary number of domains. The domain to which each tag belongs is not clearly labeled. A social media corpus might contain tags related to different topics such as tourism, computing, cooking and automobiles. It is not possible to directly extract the relevant tags for a topic, because the topics are not explicitly represented in the corpus. However, it is possible to start with one or more tags known to be specific to a particular topic and then identify other tags that, for example, co-occur or are used by similar users. For example, tag recommendation applied to the tag 'HTML', related to computing, yields only related tags for the same topic. Tag recommendation thus identifies new and relevant community tags, starting from known terms. Using tag recommendation, new relevant domain terms can be extracted using other known terms. These known terms can originate from a domain ontology's lexicon and are sometimes referred to as 'seed terms'.

Wu et al. (2006) implemented a search process using the folksonomy structure that allows retrieval of resources which have only been annotated with tags that are strongly related to the input query, but have no lexical overlap with the input query. This problem is more commonly known as the "vocabulary mismatch problem" (Furnas et al., 1987; Siersdorfer and Sizov, 2009). Wu et al. (2006) used probabilistic clustering on folksonomies to generate term-clusters. These were used to investigate the ambiguity of tags via entropy measurement, to generate resource and tag recommendations for users using different recommendation strategies and to support a type of semantic search based on co-occurrence patterns of tags. Tag-resource-user relations were represented as multidimensional vectors to probabilistically cluster tags. Clusters of synonymous tags were argued to represent concepts. This tag-cluster representation was used to investigate whether the emergent semantics of

folksonomies could be leveraged to provide functionality thus far limited to semantic search systems based on ontologies.

Another way to perform tag recommendation is to employ topic modeling ((Blei et al., 2003; Blei, 2011), covered more extensively in chapter 6), to identify subjects of interest within a community. In this approach, tags are clustered in a fixed number of latent topics. The system operates on specific distributions of topics instead of individual tags. Harvey et al. (2010) proposes an extended LDA-based model for tag recommendation. The model allows for estimation of topic distributions over users and documents, and of term distributions over topics and was applied to data from Bibsonomy. Tag recommendation was performed based on a small number of tags that users entered while tagging a resource as well as a user's past annotations. Harvey et al. (2010) shows that this LDA-based model recommends more relevant tags than other tag recommendation methods such as *TopSys*, *TopUser* or *CoTag*. See table 3.1 for definitions of these methods.

Abbreviation	Description
TopSys	The tags most frequently used in the system.
TopUser	The most frequently used tags by the users who tagged the resource.
CoTag	Tag co-occurence using asymmetric normalization (Schmitz, 2006).

Table 3.1: Three basic tag recommendation strategies discussed in Harvey et al. (2010).

The extended LDA-based model from Harvey et al. (2010) improves on the traditional LDA model for tag recommendation from Krestel et al. (2009). Krestel et al. (2009) achieved a best F-score of 0.40 (0.37 precision and 0.44 recall) on a Delicious dataset. The tags from the five most recent bookmarks of a user were used for prediction. Harvey et al. (2010) reported that the CoTag method from Schmitz (2006) outperformed a much more computationally expensive LDA-based model, similar to that of Krestel et al. (2009). The extended LDA-based model of Harvey et al. (2010) also makes small, but statistically significant, improvements on CoTag in terms of precision achieving 0.43 (+1.49%) for the top 5 tags and 0.22 (+7.9%) for the top 20 tags respectively. The performance of *TopSys* and *TopUser* was very poor in comparison to *CoTag* (Harvey et al., 2010).

The comparison between *TopSys*, *TopUser*, *CoTag*, listed in table 3.1, and the two LDA-based models by Krestel et al. (2009); Harvey et al. (2010) shows that CoTag performs only marginally worse than the extended LDA-based model (Harvey et al., 2010). It is for this reason that *CoTag*, because of its effectiveness, efficiency and simplicity, is used in the ontology enrichment approach presented in this chapter. *CoTag* is described in more detail in section 4.3.2 and its role in the overall methodology is explained in section 4.3 .

2.2 Learning taxonomies from folksonomies

It can be relevant to make use of the structure that is implicitly available in folksonomies to derive simple taxonomies. Extracting the tags, concepts and relationships from a folksonomy enables more substantial ontology enrichment, because a taxonomy extracted from a

folksonomy is easier to employ for ontology enrichment than the original folksonomy itself. This is due to the fact that in many domain ontologies the concepts are structured using some hierarchical or taxonomic structure; it is often mandated by the upper-level ontology. The fact that one can extract a taxonomy from a folksonomy makes it easier to integrate it within an existing taxonomic structure of an ontology.

Folksonomies consist of a flat vocabulary, i.e. terms without relations or hierarchical structure, that is moderated by a community process. Various approaches construct rudimentary taxonomies based on clustering or co-occurrence analysis of tags in folksonomies. These taxonomies often take the form of a hierarchy of tags where generic or abstract tags are at the top of the hierarchy and very specific tags are near the bottom. The flat lexical tag-space of a folksonomy is thus converted into a hierarchy of tags, but the exact nature of relationships between tags in this hierarchy remains vague.

Mika (2005) proposes the Actor-Concept-Instance model, which describes how graph-like structures can be extracted from folksonomies, thereby reducing the gap between folksonomies and the Semantic Web. His landmark paper provides a model for mathematically describing users, resources and concepts in relation to each other. Some disciplines restrict the term 'concept' to concepts in the Semantic Web sense, whereas others will also refer to a single tag or groups of tags as a concept. Overall, in this dissertation the term 'concept' is reserved for concepts from an ontology or reference repository with a URI. The inclusion of users in the regular bipartite model (concepts, resources) allows for additional integration of Social Web data and Semantic Web content.

However, the lack of explicit semantics in folksonomies makes additional methods for automatically establishing ontological relations between concepts necessary. Tags in Mika's framework are only known to be 'related' in some sense and thus constitute a type of association thesaurus. Betweenness centrality (covered in more detail in chapter 4), clustering coefficients and network constraint are used in Mika (2005) to identify sets of synonyms and establish networks of concepts. Set theory is used to distinguish between broader and narrower relations between concepts.

Schmitz (2006) created a method using a probabilistic model, tree pruning and reinforcement in order to infer hierarchical structure from conditional tag co-occurrences in Flickr. This is an improvement from purely associative connections, as in Mika (2005), and is more informative in an ontological sense. Schmitz (2006) reported an accuracy of 0.51 on his dataset for properly identifying a subsumption-relation between two or more tags.

Heymann and Garcia-Molina (2006) also created a hierarchical model, by building a graph of tags where the similarity between tags is defined by the cosine of two tag vectors. Each of the elements of a tag vector contains the number of times that the respective tag has been used to annotate the resource by zero or more users. Heymann and Garcia-Molina (2006) then continue by building a hierarchical tree in decreasing order of the closeness centrality value calculated from the tag-similarity graph. They validated their method using both Delicious and CiteULike datasets, but reported no actual performance figures.

Various measures for semantically grounding the 'relatedness' of tags are compared in Cattuto et al. (2010). Their aim was to come to a "more systematic characterization and validation of tag similarity in terms of formal representations of knowledge" (Cattuto et al.,

2010, p. 615). Five measures of tag relatedness were compared; co-occurrence, three cosine distributional metrics (tag, resource and user context) and FolkRank (Hotho et al., 2006b). All the tag pairs were mapped to WordNet (Miller, 1995), in order to determine the type of semantic relationship between the generated pairs. Cattuto et al. (2010) conclude that tag or resource based cosine distributional metrics appear to yield more synonyms and spelling variants, whereas the co-occurrence and FolkRank measures rather yield more general tags, among which parent/super concepts, which, they suggest, makes them suitable for learning taxonomic relationships.

Tang et al. (2009) propose a three-stage approach for learning ontologies from folksonomies. Tagging behavior is modeled by a generative probabilistic model and four divergence methods are employed to identify the type of relationship between two tags. Tang et al. (2009) further propose an algorithm for deriving a hierarchical structure from this data. Tags themselves are modeled as topic distributions in order to assess the specificity of individual tags. A tag that is not clearly indicative of a specific topic will have a topic distribution where multiple topics will have similar values. In contrast, a tag that is specifically about a singular topic will have a topic distribution where only one topic stands out. The divergence measures identify the proper relations between tag-topic distribution probabilities. The model was trained and applied to data from CiteULike and IMDB. The accuracy of the resulting ontologies was found to be 0.62 for CiteULike and 0.66 for IMDB when compared to a large manually created web directory from the Open Directory Project[2].

There is a downside to the approaches from Mika (2005); Schmitz (2006); Heymann and Garcia-Molina (2006); Tang et al. (2009): It is very difficult to extend these approaches towards the extraction of relations other than hierarchical ones (Schutz and Buitelaar, 2005). However, these methods can still be used in an unsupervised manner to extract information that is useful to ontology enrichment such as a hierarchy of tags.

2.3 Integrating folksonomies with formal ontologies

The previous section has surveyed methodologies and techniques aimed at extracting a taxonomy from a folksonomy. This also includes identifying clusters of synonymous tags and treating these as concepts (Mika, 2005). This section surveys related work that aims to explicitly link tags to URI-based concepts and relations from formal ontologies and reference repositories. This constitutes an integration of folksonomies and formal ontologies. The previous section has only discussed the extraction of taxonomies from folksonomies without establishing an explicit link to concepts and relations from other ontologies. Explicitly interlinking tags with concepts from formal ontologies allows one to go beyond simple taxonomies, because formal domain ontologies contain a wealth of highly specific relations that are impossible to infer from a folksonomy alone.

Establishing a robust relation between tags and concepts is vital for applications that want to make use of an ontology in combination with the large amount of data available in folksonomies. It enables one to move from the lexical to the semantic level (Buitelaar et al., 2006; Buitelaar and Cimiano, 2008). Tags are treated as potential lexicalizations of concepts.

[2] http://www.dmoz.org/

Another issue related to treating individual tags as concepts is that it raises questions on how to conceptually distinguish synonymous tags. A semantic interpretation of tags, using concepts, opens up possibilities for smarter systems that can automatically reason about domain entities as they are used in an online Community of Practice using their own continuously evolving terminology (for details see chapter 5 and (Weller et al., 2007)).

Specia and Motta (2007) describe an approach using tag preprocessing (morphologic similarity, exclusion of isolated tags), statistical tag clustering based on co-occurrence and relation identification by looking up terms in other online ontologies. Specia and Motta (2007) add semantics to tag associations by identifying the concepts that the tags express and by determining relationships via lookup in external semantic resources. Tag co-occurrence is used to perform conglomerative clustering which results in a subsumption hierarchy of tags. Tags that frequently co-occur together on a resource end up close together in the subsumption hierarchy. The tags in this hierarchy form clusters of increasing size and generality. The subsumption hierarchy of tags is used to perform disambiguation, visualization and tag recommendation. Each tag cluster is mapped to concepts or instances of existing ontologies available on the Web.

After establishing a link between tags and concepts, Specia and Motta (2007) check for possible relationships among pairs of concepts. Problems with ambiguity of tags are prevented by only considering external ontologies that contain both tags. The end result of Specia and Motta (2007)'s approach is a number of concepts with meaningful semantic relations derived from existing ontologies within an overall hierarchical structure based on the folksonomy characteristics.

A similar approach is described by Damme et al. (2007) who focuses on how an actual ontology can be generated on the basis of a folksonomy. Damme et al. (2007) propose that, in addition to providing the tags, the community must also directly help to identify, judge and/or approve relations in the ontology.

Lin et al. (2009) have a related approach for extracting ontological structure from folksonomies. They deviate from clustering-based approaches, such as the one by Specia and Motta (2007) and Damme et al. (2007), and instead perform association rule mining and token based similarity. Association rule mining extracts statistical patterns of the form $t_1, \ldots, t_m \Rightarrow t_n$, where t_i is an attribute from a large database. Applied to the context of tagging, an association rule captures the pattern that given tags $t_1 \ldots t_m$ it is likely that tag t_n will also be added to a resource. Such association rules can clearly be used for tag recommendation. What is elegant about this approach is that it is not limited to pairs of tags, but can accommodate more complicated tag co-occurrence patterns.

Lin et al. (2009) investigate the type of semantic relations that the extracted association rules express, using Wordnet. Lin et al. (2009) note that Wordnet by itself is insufficient to properly deal with domain specific concepts, jargon and discovering relationships between domain specific concepts. This is an important observation, because their approach is completely reliant on WordNet for discovering relations and linking terms to concepts.

Related approaches of establishing a link between folksonomy tag spaces and reference concepts exist that seek to improve precision and applicability through structural inference. Angeletou (2010) enriches the tags of an existing folksonomy with a layer of semantic content.

This approach includes a component for mapping a tag to a concept from an ontology. Once the corresponding concept has been determined the various typed relations can be retrieved. For example, in (Angeletou, 2010) the tag 'building' is automatically linked not only to more general ontological concepts such as 'Infrastructure' and 'Manmade Structure', but also to more specific ones such as 'Restaurant' and 'RailroadStation'. This approach has a reported recall of 0.49 and precision of 0.93 on a FlickR-based tag set when using WordNet as the source of semantic content.

All of the former approaches have focused on scaffolding or generating ontologies from folksonomies, but little attention has been paid towards the enrichment of pre-existing domain ontologies using folksonomies. Although ontology enrichment based on domain corpora has been previously investigated (Faatz and Steinmetz, 2002; Buitelaar and Cimiano, 2008) and ontology enrichment as part of the ontology life cycle are active areas of research (Castano et al., 2006; Khattak et al., 2009, 2010), ontology enrichment based on folksonomy data is less well established. However, recently Alves and Santanche (2012) also identified the potential of enriching ontologies with folksonomies. "A folksonomy can represent a perspective of a wider group, but the semantics extracted from the implicit relations among tags are rather simple. An ontology is usually built by a more restricted group, but has the richness of an engineered product." (Alves and Santanche, 2012, p.19). Instead of using the word 'enrichment' they refer to the combination of folksonomies and ontologies as 'fusion' (Alves and Santanche, 2011, p.58). They discard path-based metrics for similarity due to theoretical issues brought forth by Resnik (1993) and opt for depth-based category hierarchies and a content-based measure. Alves and Santanche (2011, 2012) used WordNet to map tags to concepts. They do not explicitly deal with ambiguity of the individual tags, but deal properly with morphologic variety in a group of tags that are likely to represent the same concept, i.e. synonyms and spelling variants. Alves and Santanche (2012) build on Alves and Santanche (2011) and offers minor improvements. They do apply a disambiguation approach to deal with ambiguous tagsets when linking it to the right ontology concept by creating a graph of the tagsets and their relations. The paper suggests that no new concepts are introduced in the domain ontology, but that existing ontological relations are enriched with coocurrence rates and IC-based information. They do not consider lexical enrichment of the ontology explicitly in their approach. In Alves and Santanche (2012) the main application is a support-tool for ontology designers, not the automatic enrichment of an ontology using information extracted from a folksonomy. The main focus of their work is to improve semantic similarity by integrating the semantic similarity in folksonomies with that of WordNet.

This concludes the discussion of the state of the art with respect to techniques related to extracting structure from folksonomies in the context of ontology enrichment. The following section will propose a new approach to ontology enrichment and relate that approach to the work just presented.

2.4 Related work

The work presented in this chapter combines some of the aforementioned techniques and combines them in an integrated process. While other approaches attempt to develop (light) ontologies from sets of tags (Mika, 2005; Schmitz, 2006; Heymann and Garcia-Molina, 2006;

Tang et al., 2009), the SOE approach is different, because it relies on existing domain ontologies and uses external data to enrich them. Extraction methods exclusively focused on deriving taxonomies from tag systems cannot achieve high quality results due to the unavailability of explicit structural information in folksonomies. It is for this reason that reference repositories are used to establish proper ontological relations for ontology enrichment.

SOE uses a simple method for tag recommendation and filtering based on resource co-occurrence with Jaccard normalization, because more sophisticated approaches (Harvey et al., 2010) only had marginal improvements over the simpler method (Schmitz, 2006) and are much slower and time consuming to implement (see section 2.1 for details).

SOE builds on related work that aims to reuse existing domain ontologies within the Semantic Web towards ontology enrichment (Specia and Motta, 2007; Angeletou, 2010). However, instead of relying on domain ontologies for enrichment SOE relies on reference repositories. Reference repositories, such as DBpedia (Bizer et al., 2009b), suffer from fewer coverage issues (Lin et al., 2009) than for example WordNet (Miller, 1995), i.e. more terms can be resolved to concepts. In particular, domain specific terms and concepts are much more readily available in reference repositories than in lexical resources such as WordNet. Reference repositories are used in SOE to identify new concepts and relations.

SOE also takes the lexical enrichment of domain ontologies into account, because these are crucial for end user applications (Monachesi et al., 2008, 2009). This aspect is absent from most related approaches, e.g. that of Alves and Santanche (2011, 2012). In an abstract sense the proposed ontology enrichment methodology extends the tripartite model from Mika (2005) specifically with regard to concepts and their lexicalizations.

Whereas Mika (2005) basically equated tags and concepts, SOE links tags to corresponding reference concepts. Once a tag is linked to a reference concept integration with any ontology that reuses the reference concept directly or indirectly is automatically realized. It thus constitutes an explicit integration of a folksonomy with formal ontologies and reference repositories, i.e. the folksonomy has become a type of resource within Linked Open Data. A tag is treated as a lexicalization of a concept in the strict Semantic Web sense; i.e. a concept with a stable URI from either an existing domain ontology or a reference repository. There is some methodological overlap between Alves and Santanche (2011, 2012) and SOE with respect to the integration of folksonomies and ontologies. However, SOE is designed to apply ontology enrichment entirely automatically, whereas Alves and Santanche (2011, 2012) only visualize the information and require a human ontology engineer to manually perform the actual ontology enrichment.

3 Ontology Enrichment

3.1 Overview

This section introduces the main characteristics of ontology enrichment in order to focus on the specific approach presented in this chapter, i.e. SOE. First, a generic overview is presented followed by a description of the two primary components of ontology enrichment, that is *lexical enrichment* (section 3.2) and *conceptual and relational enrichment* (section 3.3).

Firstly, it is important to distinguish between *ontology learning* and *ontology enrichment*. The former concerns the creation of a new conceptual structure, whereas the latter pertains to the revision of concepts and relations of an existing ontology. A domain ontology is either bootstrapped from scratch by mining a domain corpus (Fortuna et al., 2008) or an existing ontology is *enriched* based on the information extracted from a domain corpus (Buitelaar and Cimiano, 2008). The ontology that is used for the evaluation and experiments in this chapter is the manually constructed LT4eL lexicalized ontology on computing with the associated English lexicon of about 1200 concepts (Lemnitzer et al., 2008).

Ontology enrichment concerns the revision of concepts, lexicalizations and relations that **improve** the pre-existing structure of an ontology. Automatic ontology enrichment is a specific methodology for *ontology maintenance*. Ontology maintenance can also be achieved through manual revision of an ontology. Large scale revision of the primary conceptual structure of a domain ontology is outside of the scope of the type of automatic ontology enrichment that will be discussed in this chapter.

Secondly, it may appear as if ontology development is a linear process which moves from creation to maintenance whereas, as with software, this is actually an iterative process (Boehm, 1988). Ontologies need to be continuously updated and revised to reflect the development of their respective community. Each application of the ontology enrichment process is expected to improve an existing domain ontology. It achieves this fact by adding and removing new domain concepts and relations and by updating the lexicalizations in the ontology such that they reflect the community vocabulary that is in active use. In effect, each iteration of ontology enrichment reduces the differences between a community's understanding of a domain and how the domain is formalized in an ontology.

Ontology enrichment is not necessarily an additive process where only new information is added and old information is forever retained. Ontology enrichment might actually remove parts of a domain ontology, because the corpus suggests that it is no longer used, relevant and/or accurate. An ontology enrichment process can include human input about such decisions, because the members of a CoP might not always make all facts about their domain explicit. Human input can be used to manually override the ontology enrichment's decision to remove information from the domain ontology. For example, because it reflects some invariant base characteristic of the domain (Damme et al., 2007) that is not made explicit in the corpus. It is becoming clear that ontology maintenance is a frequent and incremental process. There is thus a case to be made for adopting *agile* methods (Fowler and Highsmith, 2001) to maintain existing ontologies (Luczak-Rösch, 2009; Auer and Herre, 2007). Automatic incremental ontology enrichment and ontology learning integrate well with an agile approach to ontology maintenance.

Recall that ontologies are stored as sets of triples in RDF stores (see chapter 2 section 2.1.1). This implies that repeated applications of ontology enrichment do not lead to duplicate enrichment results, because a repeatedly added triple is stored only once.

Ontology enrichment in general covers all aspects of revision, which includes adding, removing and updating concepts, relations and lexicalizations. The SOE approach, presented in section 3, is however limited to adding new concepts, relations and lexicalizations. The removal of outdated information is not part of the proposed automatic SOE approach, although it is possible and may be beneficial to include it (see chapter 7 section 3).

The initial step of any automatic ontology enrichment process is to identify which parts of the ontology are eligible for enrichment and with what concepts. These so called *candidate concepts* may become part of the domain ontology and can, for example, originate in a domain corpus, another domain ontology or a reference repository. A candidate concept might not be known immediately, for example when only terms are identified in a corpus. In that case, an additional step is required where the relevant term is resolved to a candidate concept, i.e. the term is treated as a lexicalization of a concept.

After identifying which parts of the domain ontology should be enriched and with which concepts, the ontology enrichment process branches into two different types of enrichment, depending on what data is available:

1. *Lexical enrichment* (section 3.2) - Add a newly identified related term as a new lexicalization to an existing concept in the domain ontology, i.e. integrate the vocabulary of a Community of Practice with an existing ontology.
2. *Conceptual and relational enrichment* (section 3.3) - Create or identify a new candidate concept, add the candidate concept's related terms as lexicalizations and integrate the new candidate concept using a proper relation with an existing concept in the domain ontology. If the candidate concept is already part of the domain ontology, but not directly connected to another concept, add a new relation directly connecting the two concepts.

Which of these two branches is chosen depends on additional information. Each of the two types of ontology enrichment is described in more detail in the following sections.

3.2 Lexical enrichment

Lexical enrichment focuses on the revision of the lexical items associated with domain concepts. Lexical enrichment is required when a CoP has deviated from the vocabulary that a domain ontology has captured at a certain time, but the concepts are still the same as those modeled by the domain ontology.

The use of incorrect terms from the original ontology for resource retrieval and other tasks will yield less recent resources or resources of poorer quality. The phrase 'incorrect terms' in this case refers to the situation where a user intends to access resources from a specific CoP, but unintentionally accesses resources from another, because they do not use the matching vocabulary. The fact that some resources are not using the correct vocabulary indicates that the authors or taggers of those resources are less well integrated in their respective CoP. This can either be caused by evolving terminology in the CoP, topic drift (Kirshenblatt-Gimblett, 1996) or the changing structure and division of one or more CoPs.

A motivation for including more lexicalizations is to increase the chances of users finding concepts that match a search query. For example, assume that the concept `lt4el:JavaScript` has the preferred lexicalization 'JavaScript'. Adding the additional lexicalizations 'jscript' and 'js' for the concept `lt4el:JavaScript` allows a user to find the concept using a common synonym or abbreviation. Lexicalizations that are uncommon for a specific CoP, and thus not

part of the ontology's lexicon, should still be included for usability reasons (Spitkovsky and Chang, 2012). This is only possible if the lexical layer accommodates these use cases by distinguishing between various types of lexicalizations, e.g. preferred, alternative and hidden terms (for details see chapter 2 section 2.1.1).

From a more theoretical perspective, the addition of lexicalizations to a domain ontology also serves another purpose. The lexicalizations that have their origin in social media and collaborative systems reflect the actual community vocabulary. Adding the community vocabulary to a domain ontology achieves integration with the community vocabulary and the conceptual structure of domain ontology. The dynamic community vocabulary is thus aligned with the domain ontology's lexicon thereby increasing its use and scope. Chapter 5 includes an evaluation showing the impact of integrating the community vocabulary with a domain ontology in the context of a search task.

Using the appropriate vocabulary enables one to access more relevant information in search-based systems than arcane, but technically correct, terms. Access to information is mediated by lexical competence (Marconi, 1995), the ability to use language effectively for a discourse domain. In a learning scenario, not having access to high quality learning material from experts may hinder the learning progress. Enrichment of existing domain ontologies with the current domain vocabulary from the CoP was found to increase the applicability and success of an enriched domain ontology (Monachesi et al., 2009). Chapter 5 describes an evaluation of semantic search using ontology enrichment that proves this point. Ontology enrichment adds relevant terms to the lexicon of the domain ontology thereby aligning the vocabulary of the ontology with that of the community, while retaining the conceptual structure of the domain ontology.

3.3 Conceptual and relational enrichment

Conceptual enrichment focuses on the identification of new domain concepts and instances that are not yet part of the domain ontology's lexicon and concepts. In the case of SOE the source of new domain concepts is the community vocabulary extracted from social media.

In order to decide whether a term constitutes a new domain concept, both its lexical form and the ontological structure of the new concept need to be considered. The lexical form might need to be stemmed and ambiguity has to be resolved in order to determine this reliably. Analysis of the lexical form is followed by semantic analysis in which the meaning of the lexical form has to be determined. The meaning of the lexical form is either a formal concept with a URI or a word sense identifier. In the case of SOE, only a concept with a URI can represent the meaning of a lexical form. This process includes disambiguation in cases where the lexical form can refer to more than one concept.

For example, the reference concept `dbpedia:Java_(programming_language)` is a *candidate concept* for the meaning of the term 'java' after the application of a disambiguation algorithm. This concept could already be part of the domain ontology explicitly, or ontology reasoning tools (Sirin et al., 2007) can be applied in order to infer that the candidate concept is actually already part of the domain ontology. When the concept is already part of the domain ontology, conceptual enrichment of the domain ontology with this concept is unnecessary, but it may still be relationally and lexically enriched. In the case of social networks

the identification of new domain concepts is based on the tags users attach to resources, such as bookmarks, videos, slides, etcetera.

When the candidate concept constitutes a new concept not currently present in the domain ontology, the domain ontology needs to be enriched with the candidate concept by integrating it. Integration is accomplished by adding the candidate concept with an appropriate relation to an existing domain ontology concept. The relation with which the two concepts are connected meaningfully explains why the candidate concept is relevant in the context of the domain ontology. The identification of the proper relation with which two concepts should be connected is referred to as *relational enrichment*. Relational enrichment is described in more detail in the following paragraphs.

Relational enrichment There are two situations where relational enrichment is performed. The first is when the candidate concept is not currently present in the domain ontology and needs to be integrated with an appropriate relation. The second situation is when the candidate concept is already present in the domain ontology, but no direct relation exists between domain ontology concept and the candidate concept. The former situation has been previously explained. I will now discuss the second situation concerning the relational enrichment of concepts that are already part of the domain ontology.

Relational enrichment may be beneficial when two concepts, that are both part of the domain ontology, are not connected directly. However, the terms associated with these two concepts frequently co-occur in social media. This indicates that the addition of a direct relation between the two concepts in the domain ontology would reflect the community's understanding of the co-occurrence of the two concepts. As a result, the conceptual gap between the domain ontology and the CoP is reduced.

The ontological relation could be as simple as a subsumption relation, but can also be more elaborate and domain specific (e.g. `writtenBy` or `actedIn`). There are various ways of extracting useful information for ontology enrichment from data sources. One is to reuse existing ontological relations from existing domain ontologies or reference repositories (Angeletou et al., 2008) . Another is the extraction of ontological relation from documents which contain the two domain terms of interest (Schutz and Buitelaar, 2005).

Relationships indicated by a folksonomy via co-occurrence may also represent sets of terms which are frequently associated together, but whose relation is not actually part of the conceptual domain structure. For example, the domain of defeasible reasoning (Pollock, 1987) contains examples about penguins which are frequently used to apply abstract logic to concrete problems (Prakken and Vreeswijk, 2002). A naive approach to ontology enrichment might actually create a relation between 'penguins' and 'formal proofs' which makes little sense for an ontology designed to describe the domain of formal logic. A much safer route is to build on an existing validated domain structure, as is available through reference repositories.

Let me illustrate relational enrichment with a fragment of a taxonomy of the domain of computing, shown in fig. 3.1. Let there be two concepts whose tags appear frequently together (co-occur) in social media, `lt4el:HTML` & `lt4el:JavaScript`, but which are far apart in the ontology's taxonomy. Relational enrichment adds a new relation that connects the two con-

cepts directly. This makes the co-occurrence of the two terms and, optionally, their relation explicit whereas it was first obscured in the original taxonomic structure[3]. The new relation reflects an important piece of domain knowledge that was expressed in a folksonomy and has now become part of the domain ontology[4]. Section 4.7 explains how the information in the reference repository determines whether a specific ontological relation can be identified. Alternatively, the enrichment is limited to the coocurrence between the two concepts in the folksonomy (Alves and Santanche, 2012).

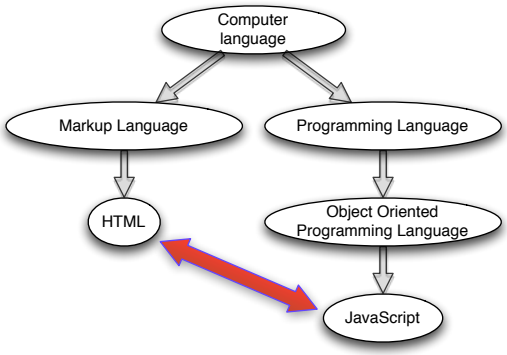


Figure 3.1: The two concepts *JavaScript* and *HTML* are embedded in a (hypothetical) taxonomy. The red arrow indicates a new relation that is added through ontology enrichment. It has been added because of the frequent co-occurrence of the two concepts in the domain corpus.

Adding relations to an ontology yields a denser web of interconnected concepts giving users a more elaborate overview of how domain concepts interrelate. The addition of new relations introduces a struggle between the ontological structure created by the original ontology designer and the new relations. There is a point, though, where too many relations between concepts might actually hinder comprehension by users, but computer software is not necessarily affected. The amount of relations and new concepts that is added is regulated by the ontology enrichment process. Details about how this trade-off is managed within SOE specifically are presented in section 4.3.

4 Social Ontology Enrichment

The previous sections have described the generic characteristics and aspects of ontology enrichment. This section provides an overview of the *Social Ontology Enrichment* (SOE)-approach and the subsequent sections will go into the implementation and design details.

[3]These links between concepts from different 'domains' are referred to as 'cross-links' in the context of concept maps (Novak and Cañas, 2006)

[4]Alves and Santanche (2012) describe a related example regarding the WordNet synsets `bible.n.01` and `christian.n.01`. These synsets have a WordNet path distance of 11, i.e. they are connected by a shortest path of length 11 having to go all the way up to the generic synset of `entity.n.01` in the hierarchy.

The SOE-approach starts with a dataset extracted from Social Media, more specifically *Collaborative Tagging Systems* (see chapter 2, section 3.3 for details), that reflect the community's understanding and vocabulary of a domain. The dataset is gathered with the help of a domain ontology and is referred to as a *seeded dataset*.

Domain ontology concepts are linked to corresponding reference concepts from a reference repository using *ontology mapping*. A reference repository is also used to *disambiguate* a tag by means of linking it to a reference concept. A tag is treated as the potential lexicalization of a reference concept. Specific tags are selected from the seeded dataset using the domain ontology's lexicon and a *similarity measure*. Only these selected tags are considered for further analysis and enrichment.

The combination of *ontology mapping* and *disambiguation* enables SOE to determine whether a tag's corresponding concept is already available in the ontology or if a new concept needs to be integrated. Figure 3.2 illustrates how *tags* are linked to *domain concepts* using *reference concepts* identified with *ontology mapping* and *disambiguation*. SOE actively uses information from reference repositories to determine the most appropriate relation with which a new or existing concept is integrated in the domain ontology. The use of reference repositories also helps to identify alternative lexicalizations of a concept.

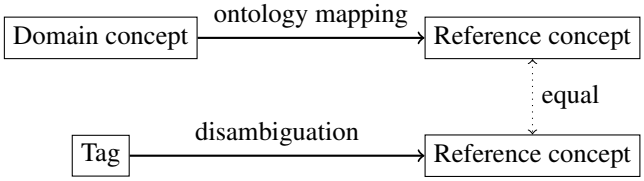


Figure 3.2: Schematic overview of how disambiguation and ontology mapping interrelate.

The SOE approach to ontology enrichment using social media consists of a number of components. These components interact and constitute an ontology enrichment pipeline. This section will give a broad overall overview of what these components are and in what order they are executed. This overview is also depicted graphically in figure 3.3.

The SOE approach to ontology enrichment consists of 5 steps:

1. *Create a seeded dataset.* (section 4.1)
   A dataset is created by gathering data from *social media* using a *domain ontology*.

2. *Perform ontology mapping.* (section 4.2)
   Concepts from a *domain ontology* are linked to corresponding *reference concepts* from a *reference repository*.

3. *Generate related terms.* (section 4.3)
   The lexicalizations of domain ontology concepts are used to generate new relevant domain terms from social media contained in the seeded dataset.

4. *Link related terms.* (section 4.4)
   Each related term is linked to a *reference concept* using information from the *domain ontology concept* whose lexicalization was used to generate the term.

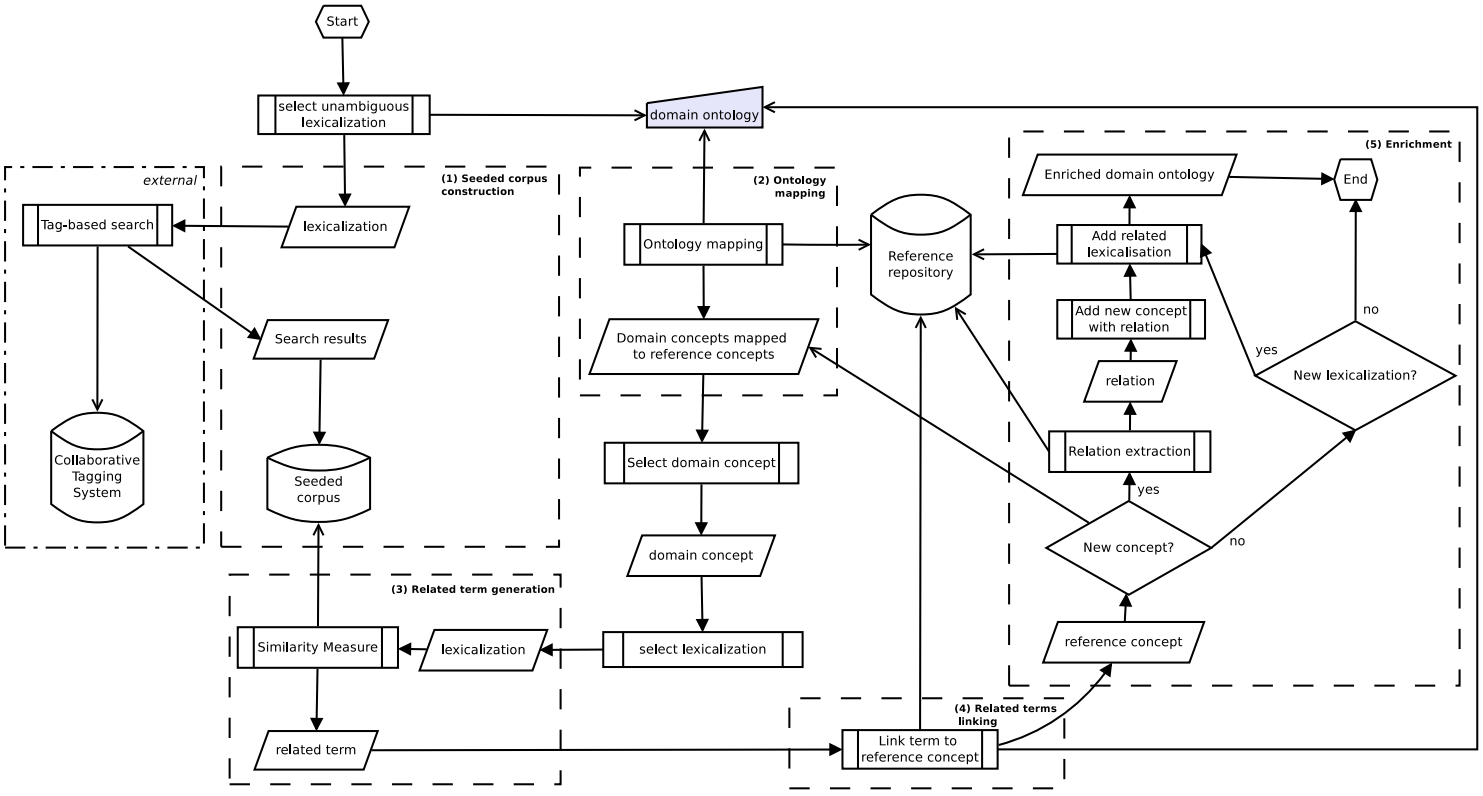


Figure 3.3: An overview of the Social Ontology Enrichment approach. The process of selecting a domain concept is repeated for each of the selected concepts in the ontology. Similarly, the process of selecting a lexicalization is repeated for each of the concepts' lexicalizations.

5. *Enrichment.* (section 4.5)
   Extract information from the reference ontology and enrich the domain ontology lexically, conceptually and relationally. The reference concept that corresponds to each generated term is considered for enrichment. Information from the reference repository is used to establish the appropriate ontological relation and additional lexicalizations which are to be added to the domain ontology's lexicon.

The following sections will now describe each of these steps in detail.

4.1 Seeded dataset

The first step of the SOE-approach is to construct a dataset using "ontology based focused crawling" (Luong et al., 2009) in order to construct a *seeded corpus*. A seeded corpus is a corpus that has been gathered for a specific domain from a larger collection of information using a form of domain knowledge, either manually or automatically. Only unambiguous terms from the ontology, e.g. as evidenced by a reference repository, are used. For example, the term 'java' would not be suitable for creating a domain specific seeded corpus about computing, because its use may include information about an Indonesian island or a type of coffee. A term such as 'HTML' is perfectly suited, because it has no other known interpretation other than that of an Internet markup language. The exclusive use of unambiguous terms, such as 'HTML', restricts the seeded corpus to those terms that are known to correspond to domain ontology concepts. This limits the amount of topic drift (Hobbs, 1990; Kirshenblatt-Gimblett, 1996), i.e. the tendency to stray from the initial topic(s).

In the context of SOE the seeded corpus consists of data from Social Media gathered with the help of a domain ontology. This corpus does not consist of actual texts, but constitutes a folksonomy with a focus on co-occurring terms. It is for this reason that the 'corpus' will be referred to as a *seeded dataset* as opposed to a *seeded corpus*. More specifically, the data is gathered using keyword-based search requests generated using a lexicalized domain ontology. The seeded dataset of Social Media data consists of a collection of resources that have been tagged by users. The seeded dataset will therefore be treated as a set of resource-tag bag-of-words pairs, i.e. a resource with one or more tags and for each tag a number indicating how many times it was added to the resource by different users.

The constructed seeded dataset is thus a collection of <resource,tag>-pairs extracted from Social Media. The seeded dataset only includes resources that are relevant in the context of the domain ontology. Subsequent operations that use the seeded corpus (see section 4.3 for details) suffer less from topic drift, because of this method.

4.2 Ontology mapping

The goal of ontology mapping is to identify for a domain ontology concept what its equivalent reference concept is using the DBpedia reference repository, a domain ontology and the disambiguation algorithm (for details see chapter 4). *Ontology mapping* as a generic concept was introduced in chapter 2, section 2.1.4. This section will describe how ontology mapping

is performed in the context of SOE. . The same method will also be applied to the SOSEM system in chapter 5. Ontology mapping, as part of SOE, is performed in order to link a *domain ontology concept* to a *reference concept*. Once linked, it can be used to decide whether a *candidate concept* is already included in the domain ontology or not.

Recall that concepts are identified via a URI, a unique identifier (see chapter 2 section 2.1.1 for details). It is possible for two concepts, from two different ontologies, to have the same meaning, but to differ with respect to the URIs used to refer to them. Ontology mapping is required in order to be able to compare reference repository concepts to domain ontology concepts.

For example, the LT4eL domain ontology (Lemnitzer et al., 2008) contains the concept `lt4el:Java` and the DBpedia reference repository contains the concept `dbpedia:Java_(programming_language)`. Both are roughly about the same concept and share one or more lexicalizations. However, the URI of both concepts are different and it is nontrivial to determine whether they are the same without reintroducing ambiguity. In this specific instance of ontology mapping the disambiguation algorithm, presented in chapter 4, can be reused to address the issue of reconciling the two concept URIs.

It is assumed that the domain ontology's concepts are a subset of the concepts available in the reference repository. The objective of ontology mapping is thus to find the corresponding reference concept for every domain concept. We know that the URIs of the concepts are different, but the terms are not. Recall that it is not possible to simply retrieve the concept for a term from the domain ontology in the reference repository due to ambiguity (chapter 4 section 4). When we consider the term from a domain ontology in the context of related terms from the same domain ontology disambiguation becomes possible. This is due to the fact that this set of terms provides the necessary *context* for successful disambiguation of the original term.

The disambiguation algorithm will determine the most appropriate reference repository concept for all the terms in the list extracted from the domain ontology. In effect, the following transitive relation is established: "*domain ontology concept* $\rightsquigarrow$ *terms* $\rightsquigarrow$ *reference concept*". After this ontology mapping operation, two comparable sets of concepts are available from the same sense repository. This allows one to identify whether an arbitrary reference concept is already available in a domain ontology by checking whether there is a domain concept that has the same corresponding reference concept.

I will illustrate the lightweight ontology mapping employing disambiguation using an example. The domain ontology concept `lt4el:JavaScript` has a number of direct ontological relations[5] to other concepts that all have a preferred lexicalisation. The preferred term is collected for each of these concepts. For the concept `lt4el:JavaScript` and all its directly connected concepts, such as `lt4el:HTML`, `lt4el:CSS` and `lt4el:DOM`, this results in the following set of terms: {`javascript`, `html`, `scripting language`, `json`, `dom`, `ajax`, `javascript library`, `css`}

Each of these terms is disambiguated if possible, as discussed in chapter 4, resulting in a reference concept for each disambiguated term. For example, in the case of the term 'javascript'

[5]Explicitly stated relations in the ontology that are not the result of semantic reasoning software taking the transitive closure of such relations.

the disambiguation algorithm links it to the reference concept `dbpedia:JavaScript`. The ontology mapping operation determines which reference concept (`dbpedia:JavaScript`) is equal to the domain ontology concept (`lt4el:JavaScript`) even if the term used for the concept is ambiguous.

Ontology mapping compensates for possible issues related to ambiguity in the concept's lexicalizations. It also enables access to additional relations and metadata available in a reference repository and make it possible to identify whether a reference concept is already included in a domain ontology. The reference repository is not guaranteed to include every abstract domain ontology-specific concepts. Similarly, the domain ontology can only map to a subset of the reference repository, because the reference repository covers a large number of domains. An evaluation has been performed that quantifies whether this happens in practice and up to what extent.

Evaluation In order to determine whether the proposed method of ontology mapping is feasible, an evaluation has been performed. It is expected that concepts that are increasingly abstract in the domain ontology are more likely to fail to be mapped to a corresponding reference concept. The reasoning behind this is that such abstract concepts are required for the conceptual structure of the domain ontology. For example, to bridge the gap between highly abstract upper level ontology concepts and concrete domain concepts. However, such intermediate concepts do not always clearly correspond to concepts outside of the domain ontology's specific conceptualization.

In this evaluation a number of domain concepts from the LT4eL domain ontology on computing, see section 3 for details, has been mapped to the DBpedia reference repository. The ontology mapping algorithm has been executed on 200 random concepts from the domain ontology. A domain expert has judged the quality of each domain concept and its reference concept. The result of this manual evaluation is shown in table 3.2.

Mapping result	#	%
Correct	112	56.0%
Acceptable	9	4.5%
Incorrect	15	7.5%
Unmapped	64	32.0%

Table 3.2: Results of ontology mapping applied to the LT4eL ontology on computing (Lemnitzer et al., 2008) using the DBpedia reference repository (Bizer et al., 2009b).

The overall performance of ontology mapping approach is summarized in table 3.2. 111 (56%) concepts are correctly linked to a corresponding reference concept. The evaluation has also identified the reasons of failure to identify a domain concept. 64 (32%) domain concepts could not be mapped to a corresponding reference concept (referred to as 'unmapped') and constitute a large part of the failures.

The reasons for failing to find a reference concept ('Unmapped' in table 3.2) for a domain concept have also been determined. Two types of failures have been distinguished: (1) absent

reference concept and (2) local domain ontology concept. (1) refers to the situation where a corresponding reference concept is missing in the reference repository. (2) refers to concepts that have been devised for the particular domain structure of the domain ontology, but have no clear interpretation outside of it. 25 (39%) of the concept mapping failures are caused by absent reference concepts. The remaining 39 (61%) failures are due to local domain ontology concepts. The primary reason for failing to identify reference concepts is thus the fact that local domain concepts are not present in the reference repository at all, rather than failure to identify the corresponding reference concept using ontology mapping. In virtually all cases this is due to very specific lexicalizations for the corresponding local domain ontology concepts. Without taking unmapped concepts into account from table 3.2, 89% of the identified reference concepts are either correct or acceptable. This is reliable enough for the purposes of ontology enrichment.

In summary, concepts from a domain ontology are linked to a corresponding reference concept in order to decide whether a candidate reference concept is already included in the domain ontology or not. This step affects the application of the similarity measure in the next step, because only domain concepts with a corresponding reference concept are to be considered in subsequent parts of the SOE approach.

4.3 Related term generation

The third step is the generation of *related terms* using the lexicalization of a given *domain concept*. The *domain ontology's lexicalizations* are used as the input. The output is a set of *related terms* extracted from the *seeded dataset*.

The generation of related terms is important for ontology enrichment, because it selects terms from a corpus that are related to the input terms. These terms and their associated concepts are used to extend a domain ontology with concepts that are highly relevant, i.e. have a large degree of semantic similarity.

The input terms are provided by a domain ontology in the form of concept lexicalizations. For each lexicalization a list of terms is selected with the help of a *similarity measure*. There are many different similarity measures, each with different characteristics. These will be properly introduced in sections 4.3.1 and 4.3.2. The type of similarity measure influences the terms that are selected for ontology enrichment, i.e. the output of a similarity measure can help select terms from a corpus. The *related terms*, obtained using a similarity measure, are likely to be relevant for the domain ontology and the input lexicalization specifically. This is because the seeded dataset is a domain-specific selection of content in social media.

In the SOE-approach, concept lexicalizations are used to select terms that co-occur in a seeded dataset. A similarity measure is then applied to sort these terms by their degree of 'semantic similarity' to the lexicalization. The first N results of this sorted set are used in subsequent steps of the SOE-approach to discover new relevant concepts, relations and lexicalizations. In effect, the similarity measure, through its selectivity, reduces the set of all possible ontology expansions to only those which are relevant in a certain domain. The concept lexicalizations determine the domain for which the expansions are triggered.

The value of N determines the impact of the enrichment process on the domain ontology.

- *String similarity measures* (e.g. Levenshtein distance);
- *Distributional metrics* (e.g. co-occurrence)
- *Vector space metrics* (e.g. cosine similarity, frequently used in Latent Semantic Analysis (LSA) based approaches (Landauer et al., 1998).)
- *Path-based metrics* (e.g. minimum number of vertices between two synsets in Wordnet)

Table 3.3: Four generic classes of similarity measures.

A large value of N leads to the introduction of many new concepts and relations between pre-existing and new concepts. A small value for N leads to reduced domain coverage, but simultaneously improves the chances of a concept being considered relevant to the community as a whole. For example, if a domain ontology is enriched with concepts and relations for the 5 most relevant terms, it is very likely that the CoP as a whole will agree on their relevance. It is much less likely that the full CoP will agree on the relevance of the 50 most relevant terms. Choosing a higher value for N necessarily means slowly moving towards the boundaries of the CoP. Thus the higher the number of concepts considered for enrichment, the larger the domain coverage, but the smaller the community agreement concerning the enrichment results.

The following sections are about similarity measures in the context of ontology enrichment. The concept of similarity measures and their applications is introduced in section 4.3.1. Section 4.3.2 describes a number of similarity measures in the context of ontology enrichment. The performance of the similarity measures, with respect to identifying relations between terms, is evaluated in section 4.3.3.

4.3.1 Introduction to similarity measures

Similarity measures are functions that compute the degree of similarity between two objects. A lack of similarity between two objects can also be interpreted as distance. Similarity is usually expressed as a real number to allow for an easy comparison of multiple pairs of objects. Similarity measures have a wide range of applications ranging from duplicate detection to spell checking amongst other things (Cha, 2007).

There are roughly four classes of similarity measures (see table 3.3). These four classes will be discussed in the rest of this section. The subsequent sections will focus on distributional and vector space metrics in the context of finding semantically related terms.

The distance between the string “mouse” and another string “mice” would be 5.0 if the Levenshtein string similarity measure was used. However, it depends on the task whether this is a satisfactory result. Take for example, two other strings such as “mouse” and “goose” which have a Levenshtein distance of 4. If the similarity measure is supposed to generate terms which are similar in meaning to some term, then the fact that “goose” gets a lower score than “mice” could be considered erroneous. This means that it is impossible to say whether a

specific similarity measure is either good or bad without specifying the task which it should solve.

The Levenshtein similarity measure is **the** prime example of a class of measures called *string similarity measures*. The more general class of 'similarity measures' can be applied to any vector, but string similarity is only concerned with the form of strings themselves. String similarity measures are fast and simple, but only take two strings into consideration. This limits the range of problems they are able to solve or contribute to. Distributional similarity measures take information into account, other than the two terms themselves, such as the overall context in which terms appear (Lee, 2001). The context in which terms occur can either be n-grams, sentences, paragraphs or even whole documents.

For example, "mouse" and "mice" occur quite often together in documents whereas "mouse" and "goose" do not. The co-occurrence similarity measure could, for example, yield a similarity score of 20 for <mouse, mice>, but a much lower score of 2 for < mouse, goose >in this hypothetical corpus.

The co-occurrence score however has a downside when applied to frequently used terms. Consider the hypothetical corpus again with the following co-occurrence scores for the next few word pairs:

- <mouse, mice>, 20
- <mouse, goose>, 2
- <mouse, a>, 35
- <mouse, the>, 45

<mouse, the>actually co-occurs more often than <mouse, mice>which would make the co-occurrence measure unsuitable for generating synonyms. We do however observe that "the" is extremely common throughout the entire corpus, but this information is not yet part of the similarity measure. Instead of using the basic co-occurrence measure as described above, Cattuto et al. (2010); Sigurbjörnsson and Van Zwol (2008) point out that this measure should be normalized in order to make co-occurrence suitable for retrieving terms with some taxonomic relationship to the input term. The normalization can be performed using the symmetric (eq. (3.1)) or asymmetric normalization methods (eq. (3.2)):

Symmetric normalization according to the Jaccard coefficient:

$$F(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{3.1}$$

Asymmetric normalization:

$$F(A,B) = \frac{|A \cap B|}{|B|} \tag{3.2}$$

F is a binary function and it takes two sets of elements: A and B. The sets A and B could be sets of documents that include a specific term. For example, a term that always co-occurs in

the same documents with another term will get a significantly lower co-occurrence value after it has been normalized using asymmetric normalization. Both symmetric and asymmetric normalization compensate for uninformative terms by dividing by the number of occurrences of both terms or one of the terms in the corpus. These methods address the issue of uninformative common words co-occurring with specific informative ones.

Vector space models work by building a vector representation of a term and then apply some similarity measure to compare the vectors (Van de Cruys, 2010). A popular similarity metric is the cosine similarity which expresses the angle of two multi-dimensional vectors. The cosine value decreases when the vectors are less alike. A simple application of this measure would be to create a vector for each term in our hypothetical corpus ('mouse', 'mice', 'goose', 'a', 'the'). Every element in the vector would be the number of occurrences of the term in a document. For example, the vector for the term 'mouse' for an imaginary corpus which consists of five different documents would be: $[2,0,0,7,1]$. This means that the term 'mouse' occurs twice in the first document, does not occur in the second document, etcetera. The vector for the term 'mice' could be $[1,0,0,6,2]$ and that for 'goose' could be $[0,3,4,2,0]$. The cosine angle as per equation 3.3 would then approximately be 0.98 for 'mouse' and 'mice' and 0.35 for 'mouse' and 'goose'. The '$\cdot$'-operator used in eq. (3.3) is the dot product operation, also known as the inner product.

$$\cos(A,B) = \frac{A \cdot B}{\|A\|\|B\|} \tag{3.3}$$

The final category of similarity measures we will consider are path-based metrics. These measures operate on a graph where each term is a vertex and every relation between two terms is an edge. Two terms are highly similar if the shortest path that connects the two terms is very short. Vice versa, terms which are dissimilar will result in a longer path through the graph. The famous Wordnet dataset (Miller, 1995) is essentially a graph which connects sets of synonymous terms (synsets) using specific semantic relations. Similar terms are connected via a short path, whereas those that are not will be connected via a longer path (Van de Cruys, 2010)[6].

As part of the search for an appropriate similarity measure for ontology enrichment only two classes of similarity measures have been considered; distributional metrics and cosine-based vector space metrics. String similarity measures were discarded, because were are only interested in semantically related terms, not just terms whose lexical form is similar. It is essential to go beyond mere direct lexical overlap. Path-based measures were also discarded, because the application needs to be data-driven and cannot presuppose other pre-existing knowledge-rich structures such as WordNet. The following section will define specific variants of similarity measures in the context of tag analysis.

[6]However, Resnik in (Resnik, 1993, p.107) questions this idea, because it is risky due to differences in the conceptual granularity of WordNet version 1.2. The argument likely extends to more recent versions as well.

$$\text{tf-idf}(t,R) = |\{r \in R : t \in r\}| \times log \frac{|R|}{1+|\{r \in R : t \in r\}|}$$

Figure 3.4: tf-idf(t,R), where t is a tag/term and R is a set of resources

4.3.2 Similarity measures

Various similarity measures have been investigated in order to assess their contribution to automatic ontology enrichment. The objective is to determine whether specific similarity measures correspond to specific ontological relations. More specifically, we want to test which measures allow for identification of more specific or common terms, alternative lexicalizations of pre-existing concepts and identification of the relevant domain in case of ambiguity.

The following similarity measures have been assessed for ontology enrichment:

- Resource co-occurrence: tags are defined to co-occur when they have been added to the same resource, potentially by different users.
- User co-occurrence: the individual users are taken into account when calculating the co-occurrence scores. A tag only co-occurs with another tag if that specific user actually added the two tags. This is the type of co-occurrence that is defined by Cattuto et al. (2010).
- Resource Cosine Similarity: For each tag, a vector is constructed with as length the number of resources. The number of times the tag is used to annotate a resource (tf) determines the content of each element in the vector. In Cattuto et al. (2010), it is shown that this method is suitable for finding synonyms. In Gemmell et al. (2008), TF-IDF (equation in fig. 3.4) is used in addition to the original method (tf). In all cases, they found TF-IDF to be superior. They evaluated three different algorithms using this measure to cluster tags: hierarchical clustering, maximal complete link clustering and k-means clustering. The latter ones yielded poor results, whereas hierarchical clustering had the best performance.
- User Cosine Similarity: For each tag, a vector is constructed with as length the number of users. The number of times the tag is used by a specific user, determines the weight.

The various similarity measures will be evaluated in the context of Social Media. They are evaluated on the basis of two properties. First, their applicability for the reliable identification of a specific ontological relation between two terms will be assessed. Second, their ability to generate appropriately related terms[7].

4.3.3 Evaluation

The similarity measures presented in the previous section have been evaluated via manual inspection of their output by domain experts. Each of the similarity measures has been provided with a list of domain terms for which each generates a sorted list of related terms

[7] What 'appropriately related terms' are will be further substantiated in future sections.

Type	Count
Users	134905
Resources	549682
Tags	195433

Table 3.4: Statistics of the RDF dataset used for ontology enrichment and similarity measure evaluation. Each figure refers to the number of unique occurrences in the data set.

from a dataset. The similarity measures have been executed against a *seeded dataset* from `delicious.com` using a specific *set of evaluation terms*. The output of each similarity measure has been manually rated by domain experts on a number of *predefined criteria*.

A dataset has been gathered from the social bookmarking site `Delicious.com`[8]. The relevant statistics of this dataset are presented in table 3.4. It has been collected between February and May 2010. It consists of tagging data represented as <user, tag, resource>-triples and formally constitutes a folksonomy (see chapter 2 section 3.3 for details).

The dataset has been aggregated using a custom crawler and contains tags on a wide range of subjects, but with an emphasis on computer related terminology. The dataset has been constructed using an "ontology based focused crawling" (Luong et al., 2009) inspired approach in order to create a *seeded dataset* (previously presented in section 4.1). More specifically, the crawler started gathering information based on unambiguous lexicalizations from the LT4eL domain ontology (Lemnitzer et al., 2008), such as 'HTML'.

The dataset is considered to be large enough to apply techniques to filter out the common vocabulary. For example, is it possible to differentiate between domain specific tags and generic ones like 'toread' or 'cool' based on the way the tags co-occur in the corpus, i.e. 'toread' and 'cool' do not clearly differentiate between different domains, whereas other tags, such as 'html' or 'javascript' do (Golder and Huberman, 2006). The choice in dataset size introduces a trade-off between reliability and latency of the identification of popular terms in social media. A dataset that only includes recent resources is up-to-date, but there is less data available for analysis. A larger dataset, gathered over an extended period of time, may include terminology which has been superseded, outdated or conflicting. However, the quantity of information supports statistical analysis. I submit that the corpus that has been gathered from `Delicious.com` is both large enough for statistical analysis and appropriately temporally constrained.

Various similarity measures have been evaluated in the context of the SOE-approach. A separate study has taken into account how many users and resources are necessary to obtain satisfactory results using similarity measures. Detailed results of these experiments are discussed in Monachesi et al. (2009).

A standard set of evaluation terms has been created for which it has been verified that the dataset contains enough information to apply statistical analysis. This set contains 12 terms within the computing domain with different levels of abstraction. The 12 standard test terms

[8]The dataset can be downloaded at no cost as RDF at `http://www.phil.uu.nl/~tmarkus/datasets.shtml`

Similarity method	Similar concepts	Synonyms	Tail usable	Close in hierarchy
Resource Co-occurrence (Jaccard)	5	1	1	4
Resource Co-occurrence (Asymmetric)	3	1	1	1
User Co-occurrence (Jaccard)	5	1	1	5
User Co-occurrence (Asymmetric)	3	1	1	1
Resource Cosine Similarity	5	3	1	4
User Cosine Similarity	3	1	1	3

Table 3.5: Rounded average results of the evaluation of similarity measures measured using a 5-point Likert-scale.

were: *java, docbook, xml, xhtml, css, tex, standards, linux, design, blog, tools* and *software*. These terms have been chosen because they were likely to be encountered during the ontology enrichment process described in section 4.

Since some measures can return thousands of results, manual evaluation of all results was not considered feasible. The manual analysis therefore focused on the first 20 items retrieved by each measure. In total 12*20=240 results were analyzed by two domain experts. The choice for 20 was an arbitrary trade-off between coverage and man-power available at the time. The output of each similarity measure was rated on a 5-point Likert-scale.

All the measures were applied to this standard list and their results were analyzed. The evaluation of SOE in section 6 also considers the first 20 related terms for enrichment. This guarantees that the evaluation results in section 4.3.3 can be reasonably transfered to the overall SOE process. The following four criteria were taken into account during the evaluation:

- *Similar concepts*: Does the measure return concepts which are similar[9] to the input term (e.g. ‘java’ and ‘jre’)?
- *Synonyms*: Does it reliably list synonyms at the top of the result list (e.g. ‘cpu’ and ‘processor’, ‘javascript’ and ‘ecmascript’)?
- *Tail usable*: Is it possible to find a pattern such as spelling errors or unrelated terms within the tail (terms with the lowest score) of the sorted results?
- *Close in hierarchy*: Are the related tags close to each other in the ontological hierarchy. The existing LT4eL domain ontology was used as a point of reference (e.g. ‘xhtml’ and ‘xml’)?

Table 3.5 provides an overview of how the similarity measures perform with respect to each of the criteria mentioned above.

The normalization method for co-occurrence greatly influences the results returned. A detailed analysis of the results in table 3.5 indicates that they can be useful for manual ontology enrichment, e.g. the similarity measures can support a human by identifying candidate concepts for enrichment. However, the results are less useful when the goal is to perform

[9] As intuitively judged by a domain expert without the use of a knowledge model, such as an ontology.

ontology enrichment automatically. For example, the first hits for *asymmetric resource co-occurrence* are very generic and thereby of little value, because the relation to the input term is too trivial. The results from the asymmetric co-occurrence measure are much more specific, i.e. lower in an imaginary conceptual hierarchy. The highly ranked terms for asymmetric resource co-occurrence are closer to the seed tag which confirms the observations from Cattuto et al. (2010).

User co-occurrence was found to be roughly equivalent to *resource co-occurrence* for larger number of resources and users (Monachesi et al., 2009). The results suggest that a small number of users does not need to be a problem with respect to the representativeness of the results as long as enough resources are tagged. When using the *user co-occurrence* similarity measure a sample size between 10-15 users and about 200 resources seems to be sufficient for a precision of about 0.75 when compared to the results from resource co-occurrence. A more extensive description of these results can be found in (Monachesi et al., 2009).

Table 3.5 shows that none of the similarity measures was able to reliably discover synonyms automatically. I hypothesize that this is due to the fact that the test set did not contain enough terms which have widely used synonyms. Computer Science terms are relatively standardized, although important exceptions exist. Another set of five terms[10] was created that did have clear synonyms or spelling variants , e.g. CPU and processor. These additional terms were then used to re-evaluate the cosine-based measures to see whether they reliably return synonyms, because the literature strongly suggests that they are suited for this task. In some cases, cosine-based measures can be used to identify synonyms. However, their ranking in the result list is unreliable. For example, if we consider the differences in lowercase/uppercase or singular/plural. The same terms appear in different forms in the dataset, e.g. 'java', 'Java' etc. We expect that given a tag, such as 'java', the other form, 'Java', will also appear high in the list. In exactly 50% of the cases, the term that only differs in case is in the first 20 elements of the sorted list. Overall, the position in the top-20 for synonyms ranged from 3rd to 20th and is thus unreliable.

In short, all of the evaluated similarity measures did not reliably correspond to a specific ontological relation in spite of individual differences. The evaluation shows that all measures suffer from too much variation in the results that they produce. Cosine measures do correspond to the results suggested by the literature (Cattuto et al., 2010), i.e. they often give synonyms a high rank. However, in the context of ontology enrichment, semantically related terms other than synonyms are much more useful. In addition, synonyms, i.e. alternative lexicalizations of a concept, can be retrieved using more reliable strategies (for details see section 4.6).

Our exploratory evaluation thus shows that none of the similarity measures can be reliably interpreted as an ontological relation. Although we failed to establish a stable one-to-one correspondence between a similarity measures and an ontological relationship, the various similarity measures do exhibit important differences. These differences are still relevant to consider when employing similarity measures to a corpus for generating related terms. SOE employs a similarity measure that highly ranks terms that are semantically related to the input term that are not synonyms. An increase in the number of synonyms decreases the domain coverage of the coocurring tags, because only the first ten tags are considered for enrichment.

[10] humor/humour, weblog/blog, processor/cpu, howto/tutorial, java/Java

For example, two synonyms refer to one concept which means that only one concept is added to the domain ontology instead of two. The *resource co-occurrence measure with jaccard normalization* fits this objective, is computationally efficient and is reported in the literature as being suitable for this task (Cattuto et al., 2010). It is employed in the SOE-approach as part of a larger process to select relevant terms from the seeded dataset in response to terms coming from a domain ontology. It is not clearly defined how the semantically related terms are precisely related to the input term, but this shortcoming is addressed in another step (see section 4.7) of the SOE-approach.

In summary, folksonomies provide us with a domain vocabulary which is validated as common knowledge by the community that has produced it. Processes using similarity measures allow us to select possible lexicalizations of concepts which are related to the existing domain vocabulary in the ontology. Similarity measures can be used to identify *semantically related terms* in a corpus[11]. New domain terms, extracted from folksonomies using a variant of resource co-occurrence, are considered to be 'socially relevant' with respect to the input lexicalization and useful for ontology enrichment.

4.4 Link related terms

In this step a term is linked to an appropriate reference concept. As inputs this step requires a domain concept and a *new domain term* that has been generated using the process described in the previous section. The disambiguation algorithm, presented in chapter 4, is employed to find the most likely *reference concept* for the term from a *reference repository*. If the term is unambiguous, as attested by the reference repository, then the reference concept is linked immediately. A disambiguation step is required if the reference repository indicates that the term is ambiguous. The disambiguation step uses terms other than those related to the domain concept. This disambiguation step is relevant for later parts of the ontology enrichment process, because the identification of the appropriate reference concept allows for retrieval of properties such as additional lexicalizations, definitions and relations for ambiguous terms.

Recall that the unambiguous lexicalizations of domain concepts are used to generate new domain terms extracted from social media. The domain ontology concept that is used to generate new domain terms via its lexicalizations is also known. Subsequent steps make use of the metadata available in the reference repository. In order to access the appropriate metadata a reference concept that represents the meaning of the domain term needs to be identified. A simple lookup in the reference repository may be sufficient to retrieve the appropriate reference concept when the new domain term is not ambiguous. For example, the term 'HTML' can only yield one reference concept (`dbpedia:HTML`). However, the term 'java' can yield many reference concepts (e.g. `dbpedia:Java_coffee` or `dbpedia:Java_class_cruiser`), because of its ambiguity. Disambiguation is required to select the appropriate reference concept.

A comprehensive description of the disambiguation algorithm used in this section is presented in chapter 4. Disambiguation and ontology enrichment are co-dependent in this dissertation

[11] Some of the literature suggests that similarity measures can also be used to reliably determine an ontological relation between two terms and by extension two concepts. However, as the evaluation in section 4.3.3 has shown, our own experiments do not confirm this to a satisfactory degree.

as will be shown in chapter 5. However, the need for disambiguation becomes clearer, in my opinion, after having understood the ontology enrichment process presented currently. In brief, the disambiguation algorithm requires a *context*, a set of mutually disambiguating terms to link each term to an appropriate reference concept. A term's ambiguousness cannot be resolved in isolation. This context is constructed by combining the domain concept's preferred lexicalization, the preferred lexicalizations of concepts directly connected to the domain concept (see section 4.2 for details) and the new domain term. The combination of these elements constitutes a set of terms, most of which originate from the domain ontology. This enables the disambiguation algorithm to identify the corresponding reference concept for all the terms, including the new domain term. As a result, the new domain term is linked to a corresponding reference concept and thereby disambiguated. The appropriate reference concept enables access to the additional metadata associated with it which will be used in subsequent steps of the ontology enrichment process.

In summary, a list of terms, generated with the help of a similarity measure, is transformed into a list of reference concepts using disambiguation. Subsequent steps exploit the reference concepts to perform ontology enrichment using the metadata available in the reference repository.

4.5 Enrichment

The final step integrates the new candidate concept, its lexicalizations, or both, into the domain ontology, thus enriching it. The steps previously described have been preparatory steps for the actual ontology enrichment where all of the information is combined. The actual enrichment of the domain ontology is described in this section.

Alves and Santanche (2012) give an excellent summary of which types of information present in a folksonomy are relevant and how they can be used to enrich an ontology:

> "*A popular tagset without a respective concept in the ontology*. It can indicate a candidate to a new concept to be added in the ontology.
> *A strong relation between two tagsets that has no correspondent relation between the respective concepts.* It can indicate some important relation not represented in the ontology.
> *Tagsets embed rich information about relations among tags and concepts.* A tagset aggregates many tags around a meaning. Its internal network of relations and the connections they have with the concepts in the ontology are rich sources for the analysis of how words are related to the meaning of concepts." (Alves and Santanche, 2012, p.24)

The first two of these observations will be referred to as 'conceptual enrichment' in this section. The last observation about the rich information of tagsets can be interpreted as a type of lexical enrichment and is also explicitly addressed in SOSEM.

The disambiguation step has identified a reference concept for each of the related terms that have been generated (for details see section 4.3). The ontology mapping step has identified a

corresponding reference concept for each domain concept when possible. The combination of these allows SOE to compare the domain ontology concept with the various reference concepts and determine how ontology enrichment will be performed.

In SOSEM each domain ontology concept is linked to a corresponding reference concept (see section 4.2 for details). Additionally, terms have been generated in the term generation step (section 4.3) and each term is linked to a corresponding reference concept. Recall that a reference concept that is linked to a term is referred to as a *candidate concept*.

A candidate concept is considered to be part of the domain ontology if a mapping exists (for details see section 4.2) from a domain concept to the same candidate reference concept. In that case, integration of the candidate concept in the domain ontology is not required, because it is already part of the domain ontology. The outcome of this comparison determines how the SOE-process should proceed with enrichment:

1. *The candidate concept is not yet part of the domain ontology.* $\rightarrow$ The new candidate concept is integrated in the ontology using conceptual and relational enrichment (section 4.7) and all relevant lexicalizations are added to the ontology's lexicon by means of lexical enrichment (section 4.6).
2. *The candidate concept is already part of the domain ontology.* $\rightarrow$ only relational (section 4.7) and lexical enrichment (section 4.6) are performed on the corresponding domain ontology concept.

The next two sections will describe lexical, conceptual and relational enrichment in greater detail.

4.6 Lexical enrichment

Lexical enrichment is about the addition of new lexical entries for a concept that is in a domain ontology. The various types of metadata available in reference repositories can be exploited to perform lexical enrichment of a domain ontology concept. No new relations or concepts are added to the domain ontology as part of this process.

Recall that lexical enrichment applies to lexicalized ontologies, i.e. ontologies that make an explicit distinction between the terms that represent a concept and the identifier used to refer to the concept. A lexicalized ontology can also be represented using RDF. As previously shown in chapter 2 listing 2.1 the use of language tags allows lexicalized ontologies to distinguish between terms from different languages. However, not all terms in a lexicalized ontology necessarily have the same status. The SKOS vocabulary (Miles et al., 2005), allows one to differentiate between types of lexicalizations for the same concept.

SKOS allows one to distinguish between a preferred lexicalization (e.g. the head term), alternative lexicalizations (e.g. popular and alternative terms for the same concept) and additional lexicalizations (e.g. spelling errors, slang). Preferred lexicalizations can be stored using the SKOS *prefLabel* property, alternative lexicalizations use the *altLabel* property and additional lexicalizations will use the *hiddenLabel* property. Listing 3.1 illustrates how these properties can be used in an RDF example.

```
1   lt4el:ComputerMouse    skos:prefLabel      "mouse"@en .
2   lt4el:ComputerMouse    skos:prefLabel      "muis"@nl .
3   lt4el:ComputerMouse    skos:altLabel       "computermuis"@nl .
4   lt4el:ComputerMouse    skos:hiddenLabel    "mius"@nl .
```

Listing 3.1: "Example of the use of different SKOS properties to distinguish between different types of lexicalizations"

Recall that in chapter 2 section 2.3 various types of lexical information have been presented that are available within the DBpedia reference repository. Lexical enrichment uses three specific types of metadata associated with every reference concept in DBpedia:

1. Resource titles
2. Redirects
3. Disambiguation links

The resource title is used as the preferred lexicalization for a concept when no preferred lexicalization is already present. This is for example the case for new candidate concepts. Pre-existing domain concepts with a preferred lexicalization will use the resource title as an alternative lexicalization, i.e. a synonym.

Terms associated with a disambiguation link are added as *hidden lexicalizations*. Hidden lexicalizations should only be used to improve concept retrieval when an uncommon concept lexicalization is used by a user. However, their use is discouraged with respect to the accepted vocabulary of a CoP as evidenced by the preferred lexicalization of the domain concept.

Redirects are added as *alternative lexicalizations*. Alternative lexicalizations are synonyms of a concept that have comparable status in a CoP, e.g. the terms ‘CPU’ and ‘processor’.

Reference concepts in DBpedia are also annotated with a language tag which also allows for enriching mono- and multilingual domain ontologies, i.e. domain ontologies originally equipped with lexicalizations for only one language can be turned into multilingual ontologies. Once a reference concept has been identified for a concept in the domain ontology, all possible lexicalizations for all languages supported by the reference repository can be added without being subject to translation errors or introducing problems related to ambiguity.

4.7 Conceptual and relational enrichment

This section describes the procedure followed by the enrichment algorithm to actually add new concepts, relations and lexicalizations to the domain ontology. It consists of two main components: *conceptual enrichment* and *relational enrichment*.

4.7.1 Conceptual enrichment

New concepts are integrated within an existing domain ontology by means of conceptual enrichment. It requires the candidate concept as an input and uses the domain ontology and a reference repository. Conceptual enrichment always implies relational enrichment, because a new concept needs to be linked to some other concept using a relation. Relational enrichment identifies the most appropriate relation with which the candidate concept should be linked to the domain ontology.

Conceptual enrichment is performed by first identifying whether the candidate reference concept is included in the domain ontology. This is accomplished by comparing the candidate reference concept to the set of reference concepts that correspond to a domain ontology concept. Recall that for each domain ontology concept, zero or more reference concepts have been identified with ontology mapping (section 4.2). If a candidate concept is already part of the ontology, only relational enrichment is required. If it is not yet part of the domain ontology, the concept and its lexicalizations, as identified via lexical enrichment, are added to the domain ontology, followed by relational enrichment.

In all cases of conceptual enrichment a reference concept is added to a domain ontology using the exact same identifier as the reference concept. For example, in the case of the DBpedia reference repository this involves a DBpedia URL. The use of the same identifier establishes an explicit link between the reference concept and the domain ontology, thereby integrating the domain ontology with other ontologies, reference repositories and the Linked Open Data cloud.

Ontologies are frequently interlinked on the level of an upper level ontology. However, the reuse of reference concepts is valuable, because they interlink the ontology with other datasets on a much more detailed level. For example, two ontologies might both acknowledge that `org1:Java` and `org2:Java` are instances of the `upper-ontology:computer-based-language`-concept via a shared upper-level ontology. However, when `org1:Java` and `org2:Java` are both linked to the reference concept `dbpedia:Java` concept all information about both concepts can be unambiguously integrated at the specificity of `dbpedia:Java` as opposed to the more general `upper-ontology:computer-based-language`-concept. This allows for more expressive inferences to be made in the larger Semantic Web in which the two ontologies are subsequently embedded via their use of shared identifiers.

In summary, conceptual enrichment identifies the appropriate reference concepts for semantically related terms. Relational enrichment always follows conceptual enrichment, irrespective of whether conceptual enrichment leads to the addition of a new concept to the domain ontology.

4.7.2 Relational enrichment

A new candidate concept is integrated with an existing concept from the domain ontology by connecting the two with an appropriate relation. This process is referred to as *relational enrichment*. Relational enrichment is also required to identify a relation between two exist-

ing domain concepts that frequently co-occur in social media (see section 3.3 for details). Appropriate relations are identified with the help of a reference repository.

Reference repositories are a good source of relations between concepts. However, they often lack the detailed formal descriptions that characterize domain ontologies. Although reference repositories contain vast amounts of useful information, it is nonetheless important to minimize the amount of information that is used, because they are only locally consistent (see chapter 2 section 2.3 for details). Limiting the amount of information used from a reference repository reduces the risk of introducing noise and errors in the pre-existing formal structure of the domain ontology.

There is a limited amount of concepts that are directly connected through an ontological relation in the reference repository itself. More specifically, for the reference repository this consists of so called 'infobox data', i.e. relationships between concepts that have been extracted from tabular information. In these cases the available relation is directly used for enrichment and relational enrichment is completed. In situations where no direct ontological relation exists between two concepts in the reference repository, several heuristics are employed to determine an appropriate indirect relation. Most of these heuristics rely on the use of a broad reference repository, such as DBpedia (Bizer et al., 2009b), that is interconnected to other datasets.

The SOE process uses three heuristics to link a new reference concept to a domain ontology concept: *shared categories*, *ontology mapping datasets* and a *default case*-heuristic that triggers if the two former ones fail. Each of these will now be described in more detail.

Shared categories The first heuristic that identifies a relation from the reference repository is based on the category-based Wikipedia classification system.

Recall that the DBpedia reference repository is automatically extracted from Wikipedia (in realtime (Hellmann et al., 2009)) (for details see chapter 2 section 2.3.1). As a result, DBpedia resources are structured according to different classification schemata available in Wikipedia. One of these classification schemata is Wikipedia categories. Wikipedia has an actively used category system which is used to group articles and by extension DBpedia resources. These categories are contained in other categories, resulting in a hierarchical structure. The category structure is formally a cyclic graph, because there is no restriction on what sub-categories a category contains, but it is possible to automatically calculate the *closest shared category* for two concepts.

For example, using the category hierarchy it is possible to identify which categories are shared by two reference concepts. It can be the case that the two reference concepts are not directly related, but only indirectly through some shared category higher up in the hierarchy. It is possible to determine whether such an indirect shared category exists by iteratively extending the categories of two reference concepts towards more generic categories until the two sets overlap.

If the two iteratively expanded sets of categories overlap then the two reference concepts share a category indirectly. The result of this process is a tree with the two reference concepts as the leaves and the category which contains the leaves at the top as the 'root concept'. Because we assume that the category structure may be interpreted as a transitive subsumption relation,

the tree fragment can be used to infer a hierarchical relation. This heuristic thus identifies a hierarchical relation similar to SKOS's skos:broader or skos:narrower relations.

When extending the set of categories in search of a shared category by two concepts, in effect, a deep tree is constructed with potentially many intermediate categories connecting the root node to the two leaves at the bottom. This tree is reduced to a tree with only 3 nodes: the shared category as the root and the two reference concepts as leaves. This is technically allowed, because the subsumption relation is assumed to be transitive. This also prevents one from having to include all of the intermediate categories that link the two reference concepts to the shared category, thus limiting the impact on the pre-existing domain ontology. By assuming transitivity, the domain ontology can be enriched, while introducing only a minimal amount of information from the reference repository.

The available information about the shared category determines the subsequent step:

1. Only the shared category is available in the target ontology → Add the candidate concept with a subsumption relation to the domain ontology's equivalent of the shared category.

2. Only the domain concept that is equivalent to the reference concept is available in the target ontology → Add the candidate concept and the shared category to the target ontology along with the proper relations.

This concludes the discussion of the use of the DBpedia categorization scheme to infer hierarchical relations. I will now discuss a secondary source of hierarchical relations that consists of datasets that express equivalences between different concept identifiers.

Ontology mapping datasets The second heuristic involves ontology mapping and reuses existing datasets that connect ontologies and reference repositories on different levels of abstraction. Recall that ontology mapping in general was introduced in chapter 2 section 2.1.4. Some effort has been devoted to mapping other ontologies such as openCyc (Matuszek et al., 2006), Yago (Suchanek et al., 2007), Freebase (Bollacker et al., 2008) and Wordnet (Miller, 1995)) onto DBpedia in order to improve its scope and semantic interoperability. These datasets consist of large collections of either concept equivalence statements (e.g. `owl:sameAs`) or `rdf:type` relations. The `rdf:type` relation is particularly useful, because it provides taxonomic information.

For example, the reference concept `dbpedia:JavaScript` is attested in DBpedia to be of `rdf:type` `dbpedia-owl:ProgrammingLanguage` and `yago:ScriptingLanguAges`[12]. Either `dbpedia-owl:ProgrammingLanguage` or `yago:ScriptingLanguAges` may be already available in the domain ontology. The candidate concept `dbpedia:JavaScript`, identified by ontology mapping of the domain concept `lt4el:JavaScript`, can thus be integrated in the domain ontology with the proper `rdf:type` relation.

[12]The uppercase 'A' is intentional, because the URI is case sensitive and actually contains this inconsistency. The reason for this inconsistency is unclear.

Default case If no taxonomic relation between the seed and the candidate concept can be determined then the candidate concept is added to the ontology using the `ltfll:related` relation. This relation asserts that there is some associative relation similar to that of the `skos:related` property. There is however a difference between the `ltfll:related` predicate and the `skos:related` property. Which I will try to explain using the following example.

Both relations are not transitive. Which means that `dbpedia:JavaScript` $\xrightarrow{\texttt{skos:related}}$ `dbpedia:HTML` and `dbpedia:HTML` $\xrightarrow{\texttt{skos:related}}$ `dbpedia:XHTML` does not imply `dbpedia:JavaScript` $\xrightarrow{\texttt{skos:related}}$ `dbpedia:XHTML`. Additionally, only the relation `skos:related` enforces symmetry and thus mandates that `dbpedia:JavaScript` $\xrightarrow{\texttt{skos:related}}$ `dbpedia:XHTML` implies `dbpedia:JavaScript` $\xleftarrow{\texttt{skos:related}}$ `dbpedia:XHTML`.

However, the resource co-occurrence similarity measure is certainly not symmetric and thus `skos:related` cannot be used. This is due to the fact that when generating a term *B* in response to some term *A* it is not guaranteed that term *A* is included within the first *N* related terms for term *B*. It is for this reason that the `lftll:related` relation has been included. It acts as a default strategy for socially relevant concepts which have failed to be included with a clear (hierarchical) property into the target ontology. It is conceptually similar to `skos:related`, but it is not symmetric.

In summary, several heuristics are used to infer relations from the DBpedia reference repository. In many cases an appropriate relation is identified with which a new reference concept can be integrated with existing domain concepts. In all situations the impact on the existing domain ontology structure is limited.

5 Example

I use a concrete example to illustrate the entire SOE process: the enrichment process starts with the LT4eL domain ontology on computing. First, using its unambiguous lexicalizations a seeded dataset is gathered from social media.

Second, SOE iterates over all the available domain concepts and their preferred lexicalizations. For each domain concept a reference concept is identified by means of ontology mapping. The enrichment process continues by iterating over each domain concept with its associated reference concept. Now, let us assume that at some point the ontology enrichment process has arrived at the `lt4el:XHTML`-concept which has been linked with the reference concept `dbpedia:XHTML`. The preferred lexicalization of `lt4el:XHTML` is extracted from the lexicalized domain ontology. Let 'xhtml' be the preferred lexicalization. It is passed on to the resource coocurrence similarity measure which generates a sorted set of tags in associated with 'xhtml'. Now, let the term 'xslt' be one of these generated tags. The term 'xslt' is attested, unambiguously, in the reference repository as the preferred term for the reference concept `dbpedia:XSLT`. The ontology mapping performed previously did not identify a domain concept that matches the candidate reference `dbpedia:XSLT` and as a result conceptual and relational enrichment is required.

There is no direct ontological relation attested in DBpedia between dbpedia:XHTML and dbpedia:XSLT. Therefore, the use of a heuristic is required in order to identify an appropriate relation connecting lt4el:XHTML and the candidate reference concept dbpedia:XSLT. A search for a shared category determines that dbpedia:XSLT shares the category 'XML' with the dbpedia:XHTML concept. Given that the category 'XML' is already present as a concept in the domain ontology (lt4el:XML) the new concept dbpedia:XSLT can be added as a subclass of it.

A secondary term 'dom' is also generated by the similarity measure in response to the preferred lexicalization of lt4el:XHTML. The term 'dom' is determined to be ambiguous in the reference repository. The disambiguation algorithm is applied to the term 'dom', the preferred lexicalizations of lt4el:XHTML and the lexicalizations of its linked concepts in the domain ontology up to a distance of two. The disambiguation algorithm determines that dbpedia:Document_Object_Model is the most likely reference concept for the ambiguous term 'dom'. Again, there is no direct relation attested in DBpedia between dbpedia:Document_Object_Model and dbpedia:XHTML. Therefore, SOE selects the shared categories-heuristic in order to determine a meaningful relation. dbpedia:Document_Object_Model and dbpedia:XHTML share the category 'HTML' which is already present as a concept in the domain ontology as lt4el:HTML. As a result, dbpedia:Document_Object_Model is linked to HTML using the skos:broader relation.

All available lexicalizations for dbpedia:XSLT and dbpedia:Document_Object_Model are transferred to the ontology's lexicon using lexical enrichment. Similarly, additional lexicalizations attested in the reference repository for lt4el:XML, lt4el:HTML and lt4el:XHTML are also added to the lexicon of the domain ontology.

6 Evaluation

The previous sections have described the various stages of the SOE approach. This section will evaluate the performance of the approach on the enrichment of an existing domain ontology using data gathered from a Collaborative Tagging System.

A folksonomy dataset has been acquired by crawling the collaborative tagging site delicious.com. Information was gathered on users, resources and tags. The resulting data set contains 598379 resources, 154476 users and 221796 tags on a wide range of subjects, but with an emphasis on computer related terminology in the context of LT4eL ontology using the seeded dataset approach. This is the same dataset as was previously mentioned in table 3.4. This data set is used to generate semantically related terms in this evaluation. As a reference repository, DBpedia version 3.8 is used. The relevant data sets for DBpedia version 3.8 were downloaded and queried by loading them in a local instance of a triple store.

We want to enrich the LT4eL domain ontology on computing that was developed in the Language Technology for eLearning project[13]. It contains 1002 domain concepts, 169 concepts from OntoWordNet (Gangemi et al., 2003) and 105 concepts from DOLCE Ultralite[14]. The

[13] http://www.lt4el.eu

[14] A lightweight upper ontology for grounding domain ontologies in a basic set of concepts. It is available at: http://www.loa-cnr.it/ontologies/DUL.owl

connection between terms and concepts is established by means of language-specific lexicons, where each lexicon specifies one or more lexicalizations for each concept (Lemnitzer et al., 2007).

The first question that we wish to answer given these data sets is whether the enrichment results are of sufficient quality (section 6.1). In order to do this we need to compare the end result of the enrichment process, the enriched ontology, with the original unenriched domain ontology. This allows us to determine what parts of the ontology are the result of the enrichment process and which belong to the pre-existing domain ontology.

The second question is whether there is any overlap between a manually enriched ontology and an automatically enriched one (section 6.3). Some overlap between the two ontologies is to be expected, but we also expect interesting differences to appear.

6.1 Enrichment quality

The quality of the ontology enrichment process has been determined via manual inspection of the results. Reference concepts that are similar to upper ontology concepts have been discarded in this evaluation[15]. These were discarded because they were too generic to be considered for inclusion in the domain ontology. Inclusion of those resources would skew the results too much in the direction of these very general categories as they do not contribute much to the overall quality of the ontology. Cucerzan (2011, 2012) also reports suffering from "spurious information", i.e. the inclusion of poor concepts for common phrases that reduce the accuracy of the system. He addresses this by “employing a logistic regression classifier trained on 1,000 manually labeled Wikipedia pages” and reports an accuracy of 99% using that approach. A similar methodology can be employed to replace the manually constructed list of concepts with an automatic procedure as part of SOE. The results with and without the manual concept filtering are presented in table 3.6 Additionally, a new relation is not introduced for a concept if the two candidate concepts are already in the domain ontology and they are connected by a path of length two.

As part of this evaluation only unambiguous terms were eligible for enrichment. This allows us to evaluate the performance of the ontology enrichment pipeline in isolation and thus independently from the performance of the disambiguation algorithm itself. Chapter 4 contains both an evaluation of the disambiguation in isolation and an extended ontology enrichment experiment, similar to the one presented here, that includes the disambiguation algorithm in the overall SOE-process. We expect the extended ontology enrichment evaluation in chapter 4 to include more concepts and terms, because it allows for the enrichment with ambiguous tags and by extension concepts, such as `dbpedia:Mouse_(computing)` and `dbpedia:Java_(programming_language)`.

The application of SOE has led to a significant amount of new concepts and relations related to the domain ontology. All of the resulting RDF-triples that either integrate new reference concepts or establish new relations between domain concepts have been manually verified by a domain expert. The most important statistics that resulted from this manual verification are shown in table 3.6

[15] The exact list of concepts that has been discarded is listed in appendix A

	With filtering				Without filtering			
	Category distance 2		Category distance 3		Category distance 2		Category distance 3	
	Associative	Ontological	Associative	Ontological	Associative	Ontological	Associative	Ontological
Acceptable	470 (77%)	230 (98%)	444 (62%)	342 (75%)	503 (59%)	234 (91%)	446 (59%)	344 (62%)
Unacceptable	133 (23%)	4 (2%)	267 (38%)	112 (25%)	348 (41%)	23 (9%)	309 (41%)	215 (38%)
Total	603	234	711	454	851	257	755	559

Table 3.6: Enrichment results from the LT4eL-ontology for new concepts and relations. The numbers refer to the number of unique RDF-triples that have been added to the ontology. *Associative* refers to the use of the `ltfll:related` relation. *Ontological* refers to the use of either a shared category or a DBpedia specific property. These results have been obtained with DBpedia version 3.8 and only the first 10 cooccurring tags for a concept lexicalization using a variable category distance.

The results in table 3.6 are limited to the first 10 that were generated through the resource coocurrence similarity measure. Increasing this number was not feasible at the time due to time constraints. As a result, the total number of concepts decreases, but the relevance of those concepts as indicated by the online community should be higher. This is a concrete implementation of the idea that the coocurrence rate of a pair of tags gives an impression on the importance of the inclusion of the concept related to the tag in the domain ontology. Two runs of the ontology enrichment have been evaluated that differ with respect to the maximally allowed category distance.

Table 3.6 shows that the proportion of acceptable ontology enrichment results is up to 98% for the ontological relations. This is an excellent result. The enrichment results are split into those that had a hierarchical relation of some sort derived from DBpedia, i.e. the header 'Ontological' in table 3.6, and those that did not, i.e. the 'Associative' header. Recall that failure to identify a hierarchical relation will still lead to inclusion of the concept, but using the `ltfll:related` relation instead of a more specific one. There is thus an interaction between the associative and ontological relations; the number of associative relations will increase as the ontological relations decrease and vice versa. The accuracy of both classes of relations in table 3.6 is very different which is surprising given that the `ltfll:related` is more flexible in its interpretation.

The ontological relations are more reliable and it is tempting to increase their number by increasing the category distance. However, this comes at a cost. A larger allowed category distance increases the abstractness of the shared category, i.e. recall the similarity with upper-ontology level concepts as presented in chapter 2. Some of these shared categories are rather similar to upper-level ontology concepts and contribute little to the domain ontology itself. This fact is also clear in table 3.6, because a higher category distance, although increasing the number of statements, is associated with a decrease in accuracy. It is possible to artificially increase the accuracy by filtering certain concepts, i.e. concepts similar to upper-ontology level concepts, from the results. Results with and without filtering are included and show that increased filtering for spurious categories becomes more important as the category distance increases.

It is clear that the category distance has a big impact on the appropriateness of the ontology enrichment results. When decreasing the category distance the number of associative relations increases. This is to be expected, because the associative relations are only triggered if SOE fails to identify either a direct relation in the reference repository and a shared category. There is probably a 'sweet spot' with respect to the category distance between the 2 and 4 displayed in table 3.6. However, the quality deteriorates rapidly and additional filtering of enrichment results becomes increasingly important.

The data presented in table 3.6 presents the number of unique triples added to the ontology that establish new relations between new and existing concepts. However, these numbers do not represent the number of new concepts, i.e. a single triple can introduce up to two new concepts in the ontology. In total, 289 new concepts have been added to the domain ontology from the reference repository for a category distance of 3 with filtering and 317 concepts without filtering. 251 new reference concepts have been integrated with the ontology when the category distance is limited to 2 when filtering is applied and 255 new concepts without filtering. This constitutes a considerable expansion of the

domain coverage of the domain ontology. This includes important new concepts such as dbpedia:Apache_Hadoop, dbpedia:BibTeX, dbpedia:Cloud_infrastructure, dbpedia:C_programming_language, dbpedia:E-book, dbpedia:Emulator, dbpedia:OS_X, dbpedia:Xbox_360, dbpedia:Facebook and many others. Recall that the results reported in table 3.6 are without the use of disambiguation.

The overall results are satisfactory and in some cases, i.e. ontological relations with a category distance of 2, excellent. However, there are no clear guidelines in the literature as to what actually constitutes high quality with respect to this type of ontology enrichment. In part, this depends on the application of the resulting enriched ontology which includes visualization (Westerhout et al., 2010; Alves and Santanche, 2012) and semantic search (see chapter 5). Although ontology evaluation metrics exist (Orme et al., 2007) their applicability is limited. Alternatively, one could argue that the required quality level of an ontology is a function of the application of the ontology in a certain context and not an inherent property of the conceptual enrichment results itself. The enriched ontology has been evaluated in an undergraduate university course and the enrichment results were appropriate as judged by students. For details on this user evaluation in a learning context see (Westerhout et al., 2010, 2011).

6.2 Lexical enrichment

The ontology enrichment pipeline also performs lexical enrichment on concepts which are already part of the domain ontology and new concepts that are introduced. This is accomplished through the use of information available in the reference repository. More specifically, the DBpedia reference repository which is rich in concept lexicalisations in the form of disambiguation links and redirects. Listing 3.2 shows a typical example of the lexical enrichment performed using the reference repository on pre-existing domain ontology concepts.

```
1  lt4el:CPlusPlus skos:altLabel "ansi␣c++"
2  lt4el:CPlusPlus skos:altLabel "iso␣c++␣programming␣language"
3  lt4el:CPlusPlus skos:altLabel "iso␣14882"
4  lt4el:CPlusPlus skos:altLabel "c+=1"
5  lt4el:CPlusPlus skos:altLabel "gxavo"
6  lt4el:CPlusPlus skos:altLabel "c++␣language"
7  lt4el:CPlusPlus skos:altLabel "c++␣programming␣language"
8  lt4el:CPlusPlus skos:altLabel "c+++"
9  lt4el:CPlusPlus skos:altLabel "cxx"
10 lt4el:CPlusPlus skos:altLabel "c=="
11 lt4el:CPlusPlus skos:altLabel "c␣with␣classes"
12 lt4el:CPlusPlus skos:altLabel "c++98"
13 lt4el:CPlusPlus skos:altLabel "export"
14 lt4el:CPlusPlus skos:altLabel "c-plus-plus␣programming␣language"
15 lt4el:CPlusPlus skos:altLabel "iso/iec␣14882:2003"
16 lt4el:CPlusPlus skos:altLabel ".cxx"
17 lt4el:CPlusPlus skos:altLabel "iso␣c++"
18 lt4el:CPlusPlus skos:altLabel "c++␣standard"
19 lt4el:CPlusPlus skos:altLabel "cee␣plus␣plus"
20 lt4el:CPlusPlus skos:altLabel "c++␣program"
21 lt4el:CPlusPlus skos:altLabel "iso/iec␣14882"
22 lt4el:CPlusPlus skos:altLabel "++c"
23 lt4el:CPlusPlus skos:altLabel "c␣plus␣plus␣programming␣language"
```

Category distance 2			Category distance 3		
Preferred	Alternative	Hidden	Preferred	Alternative	Hidden
251 (0%)	9361 (67%)	784 (61%)	289 (0%)	9715 (64%)	831 (58%)

Table 3.7: Enrichment results from the LT4eL-ontology for new lexicalizations using DBpedia version 3.8 and only the first 10 cooccuring tags for a concept lexicalisation and using a maximum category distance of 2 and 3. The number in parentheses behind each type of lexicalization refers to the relative amount of new lexicalizations that are added to existing domain ontology concepts.

```
lt4el:CPlusPlus skos:altLabel "x3j16"
lt4el:CPlusPlus skos:altLabel "criticism␣of␣c++"
lt4el:CPlusPlus skos:altLabel "c-plus-plus"
lt4el:CPlusPlus skos:altLabel "sepples"
lt4el:CPlusPlus skos:altLabel "c++␣syntax"
lt4el:CPlusPlus skos:hiddenLabel "export"
lt4el:CPlusPlus skos:hiddenLabel "c"
lt4el:CPlusPlus skos:hiddenLabel "msl"
lt4el:CPlusPlus skos:hiddenLabel "as␣if"
```

Listing 3.2: The lexical enrichment results using the DBpedia version 3.8 reference repository for the `lt4el:CPlusPlus` concept. This listing only includes lexicalizations which were not already part of the ontology's lexicon.

Table 3.7 presents the results from the lexical enrichment for the ontology enrichment pipeline with manual filtering applied. The numbers represent the amount of unique lexicalizations added to the ontology's lexicon. The percentage behind each figure is the relative amount of those new lexicalizations that have been added to pre-existing domain ontology concepts, i.e. concepts within the `lt4el` namespace.

The amount of new lexicalizations that has been transferred from the reference repository to the domain ontology is considerable. As expected the amount of lexicalizations for the lower category distance is correspondingly higher, because no lexicalizations for the newly introduced shared categories need to be added to the ontology. The amount of new lexicalizations for pre-existing domain ontology concepts is relatively high, steadily making up for about two thirds of all new lexical items. The amount of preferred terms for pre-existing domain concepts is always 0%, because the reference repository is not allowed to override the preferred term for a domain concept. The preferred term from the reference repository is therefore added as an alternative lexicalization to the ontology's lexicon.

Although not every lexicalization is guaranteed to be relevant, as evidenced by listing 3.2, the majority will help novices, that have not mastered the proper domain vocabulary yet, to access domain concepts via incorrect or uncommon terms. As a result, the usability of the domain ontology is expected to increase.

6.3 Ontology overlap

The second question is whether there is overlap between a manually enriched ontology and an automatically enriched ontology using social media as input.

During this evaluation three different domain ontologies have been compared:

1. The original LT4eL computing ontology with the related English lexicon (1274 concepts and instances and 1657 lexical entries);

2. A manually enriched ontology which takes the LT4eL one as basis (1323 concepts+instances and 1735 lexical entries). This is our gold standard;

3. An automatically enriched ontology, which takes the original LT4eL ontology as basis[16]. (1831 classes and 2753 lexical entries)

The automatically enriched ontology has been generated by considering each co-occurring tag in the Delicious data set as eligible for enrichment. Nevertheless the lexical overlap between the manually enriched ontology and the automatic one is minimal. No conceptual or lexical overlap was found between the ontology produced by means of a manual enrichment process carried out by an expert and the automatic SOE process. Regardless of the minimal lexical overlap between the manually and the automatically enriched ontology, it is not the case that the concepts that have been added automatically are not appropriate and are misplaced in the ontology, as the preceding evaluation shows.

An analysis of the differences between the original and manually enriched ontologies identified 78 new lexicalizations and 49 new concepts. Of all the terms that have been added to the lexicon of the manually enriched ontology 61 are multi-word units and are thus a-priori not attested in Delicious. The lack of multi-word units as lexicalizations is a limitation of the use of this particular Collaborative Tagging System. The concepts and lexicalizations that have been manually added to the domain ontology are representative of the expert view of the domain, given their level of specificity, and include terms such as: 'NMTOKEN attribute', 'XML element type declaration' and 'XML attribute list declaration'. Additionally, of these the remaining 17 lexicalizations that are not multi-word units. 1 term, i.e. 'Xlink', was already available in the original ontology, but associated with a different concept. 15 of the terms are actually attested in the seeded corpus, but only 4 of them were both generated by the similarity measures and present in DBpedia 3.7.

In summary, the automatically enriched ontology includes the vocabulary of the community of users, while the manually enriched ontology includes very specialized tags provided by an expert. It is exactly this complementarity that we wanted to achieve by embedding tags into an existing ontology. The intended consumer of this ontology is unlikely to start using very specific domain concepts, but will probably use generic or erroneous terms to learn about the domain. In such cases the community vocabulary improves the usability of the domain ontology (for details and additional experiments concerning this claim see chapter 5).

[16]This specific ontology has been enriched based on DBpedia 3.7 and used only tags for lexical enrichment.

7 Conclusion

I presented a Social Ontology Enrichment approach that uses data from online Social Media to enrich domain ontologies. In the approach, I retain the approved conceptual model of the domain by maintaining the ontology structure, but complement it with the 'wisdom of the crowd' in order to provide more dynamic ontologies that take into account the evolving vocabulary of the community.

In order to reach this goal I have used similarity measures to select the relevant community vocabulary in the context of a domain ontology from a seeded dataset. Ontology mapping is used to identify links between the domain ontology and the reference repository. Reference repositories are used to provide structured information that support linking the community vocabulary using appropriate relations to the existing domain ontology structure. The evaluation shows that this enrichment process yields an enriched ontology of sufficient quality. Up to 94% of the new concepts are added with an appropriate relation to the existing domain ontology.

The results also show that automatic ontology enrichment using social media can help reduce the gap between the vocabulary of a Community of Practice vocabulary and a domain ontology created by experts.

Chapter 4

Graph-based disambiguation

1 Introduction

The sharing of resources in Social Media has been greatly enhanced through the use of tag-based resource annotation. A large amount of information in Social Media has therefore been annotated with tags to make discovery of relevant material easier. Tags are popular and easy to use, but the ease of use of tagging comes at a cost; a resource tagged with, for example, 'mouse' does not tell us whether it's about a computer interface device or an animal.

At the same time there is the Semantic Web; a rapidly growing amount of structured data that computers can use to understand information. For example, in the Semantic Web there is no confusion as to what a 'mouse' is, because concepts with a unique URI are used as opposed to term-based annotations in Social Media. Having a method that can take a term from Social Media and identify the proper concept regardless of any ambiguity would allow for a semantic understanding of the content of tags. It would enable one to differentiate between the use of 'mouse' in its different contexts without having to require users to adopt a concept-based annotation scheme.

This transition from tags/terms (Social Media) to concepts (Semantic Web) is formally referred to as *disambiguation*. Through disambiguation, a robust link is established between the community vocabulary as it is used in Social Media and ontologies and reference repositories on the Semantic Web. Without disambiguation it is not possible to maintain the high precision and quality of the Semantic Web when working with data from Social Media.

In this chapter, I will present an unsupervised graph-based disambiguation approach that only relies on simple associative data. It is based on a reference repository, i.e. DBpedia, which makes it possible to accommodate the shifting vocabulary of online communities and automatically integrate new concepts as they appear in online discourse. The proposed approach supports disambiguation of incompatible sets of concepts, such as resources annotated with terms from multiple domains or mutually ambiguous terms. The evaluation (section 5) will show reasonable performance on tag disambiguation compared to well established methods based on lexical overlap such as LESK and 'most frequent sense'.

This chapter starts with a review of the state of the art and related work in section 2. This is followed in section 3 by a more conceptual overview of disambiguation and more specifically *graph-based disambiguation*. Section 4 presents a new approach to disambiguation that uses a *sense inventory* to construct an interrelated graph of concepts. The new algorithm is evaluated in section 5 and compared to several other disambiguation algorithms. Section 6 summarizes the overall chapter and presents the most important conclusions.

2 State of the art

This section reviews the current state of the art with respect to disambiguation. Disambiguation is a very broad area of research and it is for this reason that section 2.1 distinguishes between different approaches. More specifically, it focuses on types of disambiguation that make use of a *sense inventory* (section 2.2). Sense inventories play an important role in graph-based disambiguation algorithms (section 2.3) and can be fruitfully used to disambiguate tags (section 2.4). Finally, section 2.5, presents a new approach that uses reference repositories as sense inventories in combination with graph-based disambiguation in order to associate tags from Social Media with concepts from the Semantic Web.

2.1 Word Sense Disambiguation

Word Sense Disambiguation (WSD) is a set of techniques which, given a set of words (a sentence or a bag-of-words), make use of 'sources of knowledge' to associate the most appropriate senses (meaning) to words in context (Navigli, 2009). WSD is a well established discipline including unsupervised, supervised and knowledge-based approaches (Navigli, 2009). Unsupervised methods do not require a large number of examples to 'learn' how to identify the proper sense whereas supervised methods do. Knowledge-based approaches require a (large) knowledge model that is related to the ambiguous terms in order to determine the proper sense.

In order to determine the proper sense of an ambiguous word, natural language based approaches look at other words in the same sentence, paragraph or document. The words surrounding the ambiguous word are referred to as *context*. The use of a context when attempting to perform disambiguation is crucial, because a word on its own without context cannot be disambiguated.

There are roughly two different approaches as to how disambiguation can be performed. The first is by clustering the terms themselves. For example, unsupervised clustering based approaches (Tomuro and Shepitsen, 2009; Specia and Motta, 2007; Van de Cruys et al., 2011) have been shown to be effective in grouping terms with similar senses. In this approach, ambiguous terms occur in clusters that consist of groups of terms with similar senses. These clusters will be separated by a distance based on the semantic similarity of each cluster. This value derived from the distance measure can then be used for further processing to derive hierarchies or groups of related terms, for example by means of conglomerative clustering. These 'term clusters' represent the semantic representation of the terms' shared meaning or theme. A simple example of such a term cluster would be: {dog, canine, doggy}. The

advantage of this unsupervised method is that no external resources are required, but human interpretation of these term clusters can be difficult when the related terms in a cluster are not clear synonyms of each other.

The second approach is to assign each term a strict sense identifier from a *sense inventory* (covered in more detail in the following section) (Sussna, 1993; Angeletou et al., 2008; Garca-Silva et al., 2009; Tesconi et al., 2008). Synonymous terms, in this approach, are not clustered, but will each receive the same word sense identifier. For example, the two terms 'doggy' and 'canine' may each be assigned to the word sense `dbpedia:Dog`.

The disambiguation algorithm proposed in this chapter follows the second approach, i.e. assigns terms to word sense identifiers as opposed to clustering terms. It is for this reason that the next section will discuss the characteristics of various sense inventories, because they play a crucial role.

2.2 Sense inventories

A *sense inventory* is a big collection of concepts and the terms associated with each concept and, optionally, additional metadata. This metadata can be a definition, describe the relations between concepts, how often a concept or term is used, what language a term belongs to and so on. Because the sense comes from a sense inventory, the sense is clear and other information about it is available such as definitions and relations.

An example of a term mapped to a sense would be *dog* $\rightarrow$ `dbpedia:Dog`; the term 'dog' has been mapped to the sense `dbpedia:Dog` thereby grounding the term 'dog'. A disadvantage of this approach is the requirement that the senses need to be available in the sense inventory beforehand. This may be problematic with recently introduced terms or concepts or very specialized terminology. An advantage of using a sense inventory is that senses are stored as URIs which allows for integration with ontologies and other Semantic Web-based systems (see chapter 2, section 2 for details).

A commonly used sense inventory used by WSD-systems is WordNet (Miller, 1995; Agirre and Rigau, 1996; Angeletou et al., 2008; Fellbaum, 2010) :

> "WordNet groups synonyms into unordered sets, called synsets. Substitution of a synset member by another does not change the truth value of the context, though one synonym may be stylistically more felicitous than another in some contexts. A synset lexically expresses a concept. [...] WordNet's synsets further contain a brief definition, or "gloss," paraphrasing the meaning of the synset, and most synsets include one or more sentences illustrating the synonyms' usage." (Fellbaum, 2010, p.232)

At the time of writing, WordNet version 3.1 has 117,659 senses (synsets for nouns, verbs, adverbs and adjectives). It is a high quality curated sense inventory with a significant number of strict relations between synsets. In recent years, less traditional sense inventories such as Wikipedia, DBpedia (Bizer et al., 2009b) and Freebase (Bollacker et al., 2008) have become

popular due to their additional coverage in terms of the number of concepts compared to WordNet (Ponzetto and Navigli, 2010).

DBpedia is also an extensive sense inventory for named entities, historical facts and concepts. The English Wikipedia from which it is primarily derived currently has 3,729,572[1] articles mostly for nouns and named entities. The structured DBpedia dataset also contains several types of (linguistic) information such as: polysemy (through Wikipedia disambiguation pages), synonyms (redirect pages), associative relations (links between Wikipedia articles (wikilinks)) and a hierarchical category system. Although the specific number of Wikipedia articles is considerably larger than the number of synsets in Wordnet it is hard to qualitatively compare the two. WordNet for example has significantly better coverage of adjectives and verbs.

There is an important advantage in using a sense inventory such as DBpedia because of its scope and origin. It covers a huge range of concepts and its implicit metadata in the form of redirect pages, disambiguation pages and category labeling provides a great deal of information about the lexical form of concepts as they are used. The use of *socially derived sense inventories*, such as DBpedia, means that whenever someone creates a new article on Wikipedia the sense is immediately available for use in a disambiguation system.

In effect, socially derived sense inventories are able to provide senses for newly introduced concepts or terms on a huge range of topics (Andrews et al., 2010; Posea and Trausan-Matu, 2009) whereas WordNet is not (Schütze and Pedersen, 1995). In addition Andrews et al. (2010) reports that static vocabularies (such as Wordnet) severely limit the applicability of WSD-systems when disambiguating tag-based data from Social Media. Using socially derived sense inventories is vital given that otherwise "the user will be suggested the right sense for a given [tag] in much less than 60% of cases" (Andrews et al., 2010, p.15). Mendes et al. (2011) used DBpedia with TF-IDF-weighted coocurrence analysis on the content of Wikipedia articles to perform disambiguation on a collection of paragraphs from the New York Times newspaper. Assignment of a random sense results in a very poor accuracy of only 17.8% which is expected given the increased polysemy of DBpedia. The disambiguation approach of Mendes et al. (2011) achieves an accuracy of 80.5%. Rizzo (2011) evaluate a number of web-scale systems that disambiguate entities in news articles from various sources. Rizzo (2011) report an overall precision for the approach in (Mendes et al., 2011) of 78.3% in their first experiment and 79.5% in a second experiment on a different set of articles. Mihalcea and Csomai (2007) performs disambiguation of entities in text and evaluate a LESK-inspired and feature-vector machine learning based approach to disambiguation on Wikipedia text using Wikipedia as the sense inventory. Mihalcea and Csomai (2007) report a precision of 92.9% and a recall of 83.1%. However, Mendes et al. (2011) notes that "In Wikify!, ... [the selection strategy] ... yields surface forms with low ambiguity for which even a random disambiguator achieves an F1 score of 0.6". This observation is supported by the fact that selection of a random sense in Mihalcea and Csomai (2007) results in a precision of 63.8% and recall of 56.9%. The most frequent sense achieved a precision of 87.0% and a recall of 77.6%. This suggests that the evaluation corpus in Mihalcea and Csomai (2007) was a generic one that did not target jargon or highly domain specific information. This might have unintentionally inflated the reported performance of the system.

[1] The number of articles shown on: `http://en.wikipedia.org/wiki/Main_Page` accessed 05-09-2011

2.3 Graph-based disambiguation

The availability of large sense inventories has made new approaches to word sense disambiguation possible for many domains. On an abstract level, these sense inventories can be viewed as graphs. Concepts or synsets take the form of nodes and the relations between the concepts are edges. This has made it possible to apply algorithms for word sense disambiguation which effectively exploit the graph-like structure of these sense inventories. The algorithms work by assigning values to nodes, edges and/or groups of them. Graph-based algorithms for disambiguation are also theoretically attractive, because social networks themselves can also be modeled as graphs. Historically, Wordnet made the first large scale computational experiments using graph-based disambiguation possible (Sussna, 1993). Over the years, algorithms originally designed for Wordnet have been adapted to newer less structured sense inventories such as DBpedia.

There are roughly two categories of graph-based measures which determine the 'importance' of nodes or edges (Ratinov et al., 2011). Local measures use local features of graph elements such as "this node has four outgoing edges" or "this edge has weight 0.42". Global measures use *global* features which describe features of interconnected parts of the graph or even the graph in its entirety. In the context of a disambiguation task, the local and global graph measures may be interpreted differently. Global measures can be seen as enforcing *all* terms to be consistent with each other, because the overall structure of the graph is taken into account. More concretely this means that choosing a sense for a term is performed while taking all other possibilities for the other terms into account. Local measures only enforce partial consistency of the assigned word sense, e.g. when disambiguating a term only a few nearby terms and concepts will be taken into account. Although global measures appear to be theoretically superior, the performance of local measures is competitive if not better than with global measures (Ratinov et al., 2011).

A large range of global and local measures is available for graph-based disambiguation. There are significant performance differences between different measures when applied to graph-based disambiguation. All research mentioned in the rest of this section has been performed using Wordnet as the sense inventory unless mentioned otherwise. The following research will give an overview of graph-based disambiguation approaches and evaluation results.

Navigli and Lapata (2007) presents an evaluation of several graph measures for unsupervised graph-based disambiguation. Among the local measures are 'In-degree Centrality', 'Eigenvector Centrality', 'PageRank', 'HITS', 'KPP', 'Betweenness centrality' and 'Maximum Flow'. The global measures considered are 'Compactness', 'Graph Entropy' and 'Edge Density'. They report that "a relatively simple measure like InDegree performs as well as PageRank" (Navigli and Lapata, 2007, p.1398). The results suggest that local measures yield better performance than global ones. The best performing local measures are KPP, 'In-degree Centrality' and PageRank. KPP has a slight advantage over the other two measures, due to its better recall. Navigli and Lapata (2007) further report that the first-sense heuristic, which assigns all instances of an ambiguous word its most frequent sense, virtually always outperforms current (2007) unsupervised and supervised WSD methods.

Navigli (2009) reports that the most frequent sense still outperforms unsupervised WSD, but supervised methods actually surpass the most frequent sense baseline. Sinha and Mihal-

cea (2007) similarly evaluated graph measures for unsupervised graph-based disambiguation based on WordNet and settled on a voting strategy combining several measures. Tsatsaronis et al. (2010) also performed a detailed comparison of the performance of seven unsupervised graph-based disambiguation strategies and their performance on Senseval evaluation tests (Edmonds and Cotton, 2001; Mihalcea et al., 2004). The algorithms are used with the English Wordnet as a lexical database and the graph processing methods SAN (Crestani, 1997), PageRank (Brin and Page, 1998; Page et al., 1999), HITS (Kleinberg, 1999) and P-Rank (Zhao et al., 2009). HITS outperforms the others with a reported accuracy of 69.1% and 69.2% for Senseval 2 and 3 respectively.

Anaya-Sánchez et al. (2007) presents a WordNet sense clustering approach. Senses of target words are extracted from the Senseval "course grained"-task and are clustered using the Extended Star Clustering Algorithm (Gil-García et al., 2003). The resulting clusters are then sorted and the best cluster covering the largest number of terms is selected. Clustering is applied recursively with increasingly stricter requirements until only one target sense remains for the terms that are to be disambiguated. This process continues for each of the subsequent clusters until each word is assigned a sense.

Fogarolli (2009) disambiguates terms originating from Wikipedia by considering the link structure of pages that refer to each other. A term vector for an article is constructed using TF-IDF and it is compared through term overlap to the candidate senses which are retrieved via disambiguation pages similar to the LESK algorithm (Lesk, 1986). Fogarolli (2009) reports an accuracy of 90.01% when only considering articles with a symmetric wikilink to the other article and 100% when considering all links. Other experiments with more conventional terms such as those encountered when disambiguating keyphrases in Wikipedia articles have a reported accuracy of up to 94.19% (Li et al., 2011).

Han and Zhao (2010) disambiguate named entities by constructing a semantic graph using Wikipedia, Wordnet and Google search results. Semantic distances are calculated using semantic relatedness in WordNet (Lin, 1998), Wikipedia (Milne et al., 2006) and the Google Similarity Distance (Cilibrasi and Vitanyi, 2007) in order of preference. The normalized strength of the semantic relation is added as a weighted edge connecting the two concepts in the graph. A new semantic similarity measure is used that combines both the explicit edge weights and the overall structure of the graph. This new similarity measure is then employed to cluster the nodes using the hierarchical agglomerative clustering algorithm (Nanni, 2005) until some measure of dissimilarity is exceeded. Interestingly their method has some similarity to the P-Rank () structural similarity measure. State of the art term-vector similarity and Pagerank-based baselines are outperformed using the semantic similarity and graph structure between concepts by 8.7% and 14.7% (absolute) on the WePS1 and WePS2 datasets (Artiles et al., 2007, 2009).

Monachesi and Markus (2010b) present a system for "Social ontology enrichment" which enriches an existing domain ontology with the community vocabulary and concepts from Social Media. Ambiguous terms are disambiguated using a graph-based approach supported by DBpedia wikilinks and categories. DBpedia redirects and disambiguation links are employed to generate candidate DBpedia resources which function as word senses. The graph is first projected in a two-dimensional space using the Fruchterman-Reingold algorithm (Fruchterman and Reingold, 1991) and is then clustered using a Self Organising Map (Smith, 2002).

The senses from the terms are then selected from the generated clusters using the largest clusters first and associating each term with a DBpedia resource.

2.4 Tag disambiguation

The task of disambiguating tags in Social Media is closely aligned to the task of disambiguating words in natural language (Bontcheva and Rout, 2012). In the context of Social Media, tags are the object of study instead of words and the *context* does not consist of sentences or paragraphs, but collocated tags, i.e. tags added to the same resource optionally by the same user.

Tags mostly consist of English nouns (Dutta and Giunchiglia, 2009). Multiple tags can be added to a single resource, but they have no order and can thus be represented as a set of words. The primary goal of the tags is to retrieve, find or share the resource using simple, but effective terms. It is therefore the case that abbreviations of terms are quite common, because they reduce the time required to annotate a resource. However, the use of abbreviations also increases the ambiguity of the tags and thus forms an important aspect to consider when exploiting tag spaces (García-Silva et al., 2011). Andrews et al. (2010) report an average homography of the tag tokens in a large del.icio.us dataset of 4.68 and Li et al. (2011) report 4.22 candidate concepts for just terms in Wikipedia. These aforementioned terms from Wikipedia have been derived from Wikipedia article titles and the text of hyperlinks to other Wikipedia articles (wikilinks). The amount of homography is thus an important sign that any attempt at searching through Social Media needs to deal effectively with ambiguity.

Although graph-based approaches have been applied successfully to word sense disambiguation in natural language, the number of studies specifically targeting the disambiguation of tags using graph-based disambiguation is limited. It is worthwhile to consider alternatives to graph-based disambiguation in order to get a complete overview of the field of tag disambiguation.

Cucerzan (2007) extracts and disambiguates arbitrary entities in news articles. Computing the agreement between concepts directly is abandoned, due to efficiency reasons. Instead Cucerzan (2007) opts for calculating the agreement between the categories of the Wikipedia articles and a limited amount of keyphrases from the initial paragraph of the Wikipedia article, an approach similar in spirit to LESK (Lesk, 1986). In Cucerzan (2007), the evaluation is done manually against a set of twenty news stories with a reported accuracy of 91%. (Cucerzan, 2011, 2012) present improved versions of the system proposed in (Cucerzan, 2007). They are validated on the TAC's Knowledge Base Population English entity linking task from 2011 and 2012 and achieve accuracy rates of up to 0.87 and 0.77 respectively[2].

Tagpedia (Ronzano et al., 2008) is a system based on a Wikipedia corpus establishing a relation between tags and Wikipedia articles. The tag context is taken into account (i.e. the various tags attached to a resource) in order to develop a reliable method that can disambiguate a tag given one or more other tags. Tesconi et al. (2008) reuses the Tagpedia system

[2]I have tried to acquire the TAC KBP corpora for evaluation of the approach presented in this chapter, because mr. Cucerzan claims there are no other large carefully annotated sets out there (personal correspondence). However, at the time of writing, they were sadly not distributed outside of the TAC competition by the Linguistic Data Consortium.

and extends it with a disambiguation approach to link tags from del.icio.us to terms. They use Tagpedia to get the likely sense candidates for each tag and then apply a disambiguation method based on the tags associated with a user profile, popular tags for the same resource and the number of occurrences of the tag in Wikipedia article texts. These measures are combined and normalized to calculate a single sense rank which is to be optimized. They evaluated their method by considering the profiles and tags of 7 del.icio.us users and reported that "globally, the 2891 (91,52%) of the 3159 disambiguated tags had been correctly associated to the right meaning".

Garca-Silva et al. (2009) disambiguate terms as they occur in Social Media by associating each with a DBpedia resource. Word sense candidates are selected by building a vector space model of the important terms attached to each concept. The concept with the smallest cosine distance compared to all the terms in a Tagging action is chosen as the proper word sense.

2.5 Related work

The algorithm presented in this chapter builds on the previously described techniques and work presented in Monachesi and Markus (2010b). I propose a new graph-based disambiguation algorithm that exploits the community structure of graphs (Girvan and Newman, 2002). Community structure is a common property of networks of people, but, as this chapter will show, is also useful for sense inventories with a social origin such as DBpedia. The method embraces the lightweight semantics of Social Media (*folksonomies*), does not rely on rigid manually maintained resources, such as WordNet, and does not expect the sense inventory to provide detailed relations between concepts. It supports the unsupervised and automatic disambiguation of tags by constructing a network of concepts. The use of graph-clustering techniques is strongly related to other approaches to disambiguation (Anaya-Sánchez et al., 2007; Specia and Motta, 2007; Tomuro and Shepitsen, 2009). However, an important difference is that the graphs are generated using a large sense inventory similar to Garca-Silva et al. (2009); Han and Zhao (2010).

The proposed approach is a type of graph-based disambiguation, i.e. it models meaning as a network (graph) of connected terms and concepts. The proposed algorithm identifies clusters of concepts instead of terms or term vectors as in Tomuro and Shepitsen (2009); Specia and Motta (2007); Mendes et al. (2011). Degree, the number edges to or from a certain node, is used as the measure for selecting concepts within individual clusters, since it proved effective in Navigli and Lapata (2007) and is a transparent and elegant measure. The resulting approach has similarities with that of Anaya-Sánchez et al. (2007) in the sense that they share a sense clustering and cluster selection/filtering stage, but my approach is much simpler, not WordNet specific and does not require recursive clustering of candidate clusters.

Han and Zhao (2010) show excellent coverage using Wikipedia as sense inventory for disambiguating named entities. My approach is not limited to named entities but works with any type of term coming from Social Media. My disambiguation methodology also has some overlap with Garca-Silva et al. (2009) with regard to using DBpedia as the sense inventory. However, my algorithm exploits the network-like structure of concept associations in DBpedia rather than the term-overlap in a 'flat' thesaurus of different concepts as similar to LESK (Lesk, 1986). The advantage of my approach is that it is largely insensitive to the actual

terms employed in the definitions of concepts, something which is not achieved by using an approach based on term-overlap such as Garca-Silva et al. (2009). The approached presented in this chapter is also much simpler than that of Mendes et al. (2011) through its use of standard DBpedia datasets as opposed to manual extraction of surface forms while achieving better performance.

Shen et al. (2012) use a combination of Wikipedia, Yago (Suchanek et al., 2007) and WordNet-based data to perform named entity disambiguation by constructing a "semantic network" from entities detected using the Wikipedia-Miner software. Additionally, they enrich the semantic network using a combination of 'global coherence', 'semantic associativity' and 'semantic similarity' measures. Shen et al. (2012) evaluated using mostly the same news corpus as used by Cucerzan (2007) and achieved a combined accuracy of 0.95. For the TAC-KBP2009 data set the disambiguation approach proposed by Shen et al. (2012) achieves an accuracy of 0.85.

My disambiguation algorithm takes the aforementioned considerations into account and therefore exclusively uses simple associative data to perform its disambiguation. This makes it relatively easy to disambiguate new terms, because it only requires a DBpedia reference concept with a few relevant links and categories to operate. The proposed disambiguation algorithm also deals with incompatible sets of concepts, such as resources annotated with terms from multiple domains or mutually ambiguous terms. It is also not limited to the DBpedia reference repository nor is it dependent on a large text corpus, because it only presupposes a large interconnected sense inventory with associative relations. In theory, it could also operate on Freebase (Bollacker et al., 2008) which has a similar set of characteristics when compared to DBpedia. However, Freebase does not include Wikipedia pagelinks which is the primary source of information for my disambiguation algorithm. The evaluation (section 5) shows reasonable performance on tag disambiguation compared to well established methods based on lexical overlap such as LESK and 'most frequent sense'.

3 Disambiguation

The previous discussion of related work regarding disambiguation has assumed a working definition of disambiguation. However, it is important to make it more precise to fully understand the disambiguation task that is to be solved. More specifically, I am interested in a form of disambiguation that associates a *term* with a *word sense* from a *sense inventory*:

> "The goal of a WSD algorithm is to associate a word w_i occurring in a document d with its appropriate meaning or senses, by exploiting the context C in which w_i is found, commonly defined as a set of words that precede and follow w_i. The sense s_j is selected from a predefined set of possibilities, usually known as a sense inventory." (Semeraro et al., 2007, p.2)

In the specific case of disambiguation in the context of a folksonomy the task changes slightly. Instead of disambiguating a word w_i in document d, the goal is to disambiguate a tag t_i associated with a resource r. It is not possible to rely on additional information present in

resource r, because r could be a photograph or video which does not contain any text. This change in the task also affects the notion of context C. Context C is not "the words that precede and follow w_i", but instead the other tags of the same *Tagging* activity (Kim et al., 2008b), i.e. tags associated with the same resource and, optionally, user. The tags associated with a Tagging activity are viewed as the input terms $t_1 \ldots t_n$. A mapping from each term to its most likely concept/sense s_j is the desired output. This task has to rely on a very small amount of information (context C) in contrast to other work (Tesconi et al., 2008).

In the previous definition, the very general term 'word sense' is used. This refers to some kind of identifier that represents one specific meaning of a word. In a set of synonymous words, each word will refer to the same word sense. A number of word senses can be stored in a sense inventory, i.e. a collection of word senses and their relevant metadata. A sense inventory can be specifically created for this purpose, but it is also possible to treat other resources, such as *reference repositories* (introduced in chapter 2, section 2.3), as sense inventories. A *reference concept* constitutes a word sense when a reference repository is treated as a sense inventory.

The difficulty of properly disambiguating a term varies. Given a large sense inventory, some terms can be trivially disambiguated, because there is only one word sense available for that term. In practice, it is frequently not this trivial, because there are multiple word senses available for the same term. For example, the term 'ajax' has at least three senses (`dbpedia:Ajax_(mythology)`, `dbpedia:Ajax_(programming)` and `dbpedia:AFC_Ajax`). The challenge for a disambiguation algorithm is to automatically select the right concept for a term in as many cases as possible regardless of the ambiguity of the term and its context. Note, that it is not assumed that all elements of context C which occur 'near' t_i or w_i are themselves unambiguous. In practice, multiple ambiguous terms provide context for each other and thus result in a single solution.

The disambiguation algorithm is considered to be successful when the algorithm selects *one* correct concept for each term. A broad overview of the important aspects of an unsupervised graph-based disambiguation algorithm in general involves four aspects:

1. How are candidate senses selected from the sense inventory?
2. What information from the context and what metadata from the sense inventory are used to construct the graph?
3. How does the algorithm assign values to nodes and edges in the graph or modify parts of the graph?
4. How is the enriched graph analyzed in order to select a word sense for a term?

Each of these four steps forms a crucial aspect of any graph-based disambiguation algorithm that exploits a sense inventory.

As presented in section 2, there is a plethora of algorithms for performing disambiguation using a sense inventory. Some of these are tailored to a specific domain or type of data such as natural language or tags. I developed an unsupervised graph-based disambiguation algorithm in order to be able to disambiguate tags in the context of ontology enrichment

(chapter 3) and semantic search (chapter 5). There are several independent reasons for doing this:

First, at the time this research started (2009), no unsupervised graph-based disambiguation algorithms existed for DBpedia-like sense inventories. DBpedia is a good fit as a sense inventory, because it can accommodate the shifting vocabulary of online communities and automatically integrates new concepts from communities in Social Media when they are added to Wikipedia.

Second, the data that must be disambiguated originate in Social Media. Data from Social Media is often conceptualized as a graph-based structure of people and content. Using a graph-based disambiguation methodology can lead to insightful symmetries between language and knowledge on the one hand and social structure and resources on the other. These are not further explored in the current chapter, but a discussion of this subject is contained in Section 6.

Third, given the lack of appropriate datasets for supervised disambiguation algorithms in the domain of tag-based Social Media an unsupervised method is required that does not rely on labeled examples of properly disambiguated terms from Social Media.

The next section will describe the main components of the disambiguation algorithm that is presented in this chapter.

4 Graph based disambiguation using DBpedia

The previous sections have introduced and reviewed the existing work on disambiguation algorithms. More specifically, a specific type of disambiguation: associating a term with a word sense from a sense repository. This section presents a new approach to disambiguation that uses a reference repository as a sense inventory in combination with graph analysis techniques. This section describes on a broad level what operations are involved and how the algorithm works. The subsequent sections explore each step in more detail. Finally, in section 4.5, an example is presented that illustrates the entire disambiguation process.

The proposed algorithm takes a set of tags as input. No explicit distinction is made by the algorithm between the tag that is the target of disambiguation and the context in which it appears, i.e. all tags in the context are disambiguated and treated equally. In the case of tag disambiguation, the context consists of other tags attached to the same resource or user.

First, for each tag all its possible reference concepts are retrieved from the sense inventory, i.e. a tag is treated as a lexicalization of one or more reference concepts. The DBpedia reference repository serves as the sense inventory. Each of these reference concepts is referred to as a *candidate concept*, i.e. a concept that may be associated with the tag. This results in a large collection of concepts from different domains that provide different interpretations for each tag. For example, for the tag ‘ajax’ all possible senses (concepts) will be retrieved from the sense inventory.

Second, information is acquired from the reference repository to create a *graph* of concepts. A graph is constructed using the lexical and conceptual information from the DBpedia reference repository. The reason for creating a graph is that the algorithm makes use of the

structure of the graph to decide on what the right concept for a term is. The concepts by themselves are just nodes in a graph and do not provide any structure. For this reason additional relations from the reference repository are required: they specify how one concept relates to another one. As a result, the candidate concepts are linked to each other using various edges present in the reference repository, i.e. they become interconnected.

Third, the community structure of the graph needs to be determined. This is accomplished by *clustering* the candidate concepts in the graph. For example, the concepts `dbpedia:Ajax_(mythology)` and `dbpedia:Amsterdam`, retrieved from the DBpedia reference repository in the first step, are not tightly connected, i.e. there are few if any (in)direct edges between them, and will thus not end up in the same cluster. In contrast, the concepts `dbpedia:AFC_Ajax` and `dbpedia:Amsterdam` are tightly connected and become part of the same cluster. After each concept is assigned to a cluster it becomes necessary to remove some of the candidate concepts in each cluster. This is due to the fact that there might be multiple competing candidate concepts for the same term in the same cluster.

Finally, the clusters are sorted according to the number of edges that they contain. From each cluster, only one candidate concept is selected, which in effect disambiguates a term.

The following enumeration provides an overview of the disambiguation algorithm and its four steps. Note that these match the generic properties of a graph-based disambiguation system outlined previously in section 3.

1. *Retrieve reference concepts* (section 4.1)
2. *Construct a graph* (section 4.2)
3. *Cluster and filter the graph* (section 4.3)
4. *Select concepts* (section 4.4)

In summary, a graph is created for a set of terms using the lexical and conceptual information from the DBpedia reference repository. The concepts are interconnected using information from the reference repository. Relevant properties of the resulting graph are determined using community-based clustering (Girvan and Newman, 2002). The outcome of this analysis determines the order of the clusters and which concepts are identified as representing the meaning for a term.

Each of the four steps will be described in detail in the following sections. Each step is abstractly depicted in fig. 4.1. Finally, section 4.5 presents a concrete example of the overall disambiguation algorithm.

4.1 Determine word senses

The start of the disambiguation process is the transition from terms to concepts, i.e. from the lexical level to the semantic level. In order to make this transition, a term needs to be associated with all of its possible interpretations (concepts). A reference repository is used as a sense inventory to retrieve the possible interpretations are for a term.

In my approach, DBpedia (Bizer et al., 2009b) is used as the primary *sense inventory* also referred to as a *reference repository* (see chapter 2 section 2.3 for details). This sense inventory specifies for each term t_i (when present in DBpedia) a set of possible senses S (DBpedia resources). DBpedia is used as a sense inventory, therefore its resources are treated as instances and/or concepts.

DBpedia contains disambiguation links which express the relation between ambiguous lexicalizations and all possible concepts (interpretations). The disambiguation relations in DBpedia are used to produce the concepts related to any input term t_i. These disambiguation links can either be absent when the term is not ambiguous or result in a huge amount of concepts when the term is highly ambiguous. It is not uncommon for these disambiguation links to result in many concepts. For example, the disambiguation links for the term 'ruby' result in no less than 70 candidate concepts. Within the computing domain, roughly 95% of the tags can be resolved to one or more DBpedia reference concepts (which are derived from Wikipedia articles) (Posea and Trausan-Matu, 2009). Following this approach, a set of DBpedia concepts can be retrieved for any input term t_i.

Redirects in DBpedia also affect the available concepts for a term. A redirect from one term to another means that the term is a synonym for the preferred term to which it refers. In the case of a redirect only the DBpedia resource for the preferred term is retained, because it is the only one that is linked to resource that contains the required metadata. Redirects are often used to redirect spelling errors or synonyms to the preferred term and concept. All available word senses S for a term t_i are thus retrieved via the redirects and disambiguation links in DBpedia. The algorithm generates a list of concepts for each term based on the senses available in the sense inventory.

4.2 Graph construction

The second step is to acquire more information from the sense inventory to create a *graph* of concepts. The reason for creating a graph is that the algorithm makes use of the structure of the graph to decide which is the appropriate concept for a term. The concepts by themselves are just nodes in a graph and do not provide any structure. It is important to add relations that connect these concepts, because otherwise the graph would just be a collection of 'dots'. Adding relations (edges) creates a network-like structure of interconnected concepts. These relations are provided by the sense inventory, i.e. the DBpedia reference repository.

For example, the terms in the set $\{ajax, soccer, amsterdam\}$ can each mean very different things, but the combination of these terms yields only a single concept for each term. A *graph-based disambiguation algorithm* determines the correct concept using the *structure of the graph* that results from interconnecting the concepts, i.e. the structure of how these concepts relate.

The previous step generated a set of concepts for each term, but did not give any information as to which concept should be selected. My disambiguation algorithm makes use of unsupervised *graph-based disambiguation* for mapping each term to its most likely concept/sense. It fits the framework described by Tsatsaronis et al. (2010):

> "Such methods create a graph comprising the words to be disambiguated and

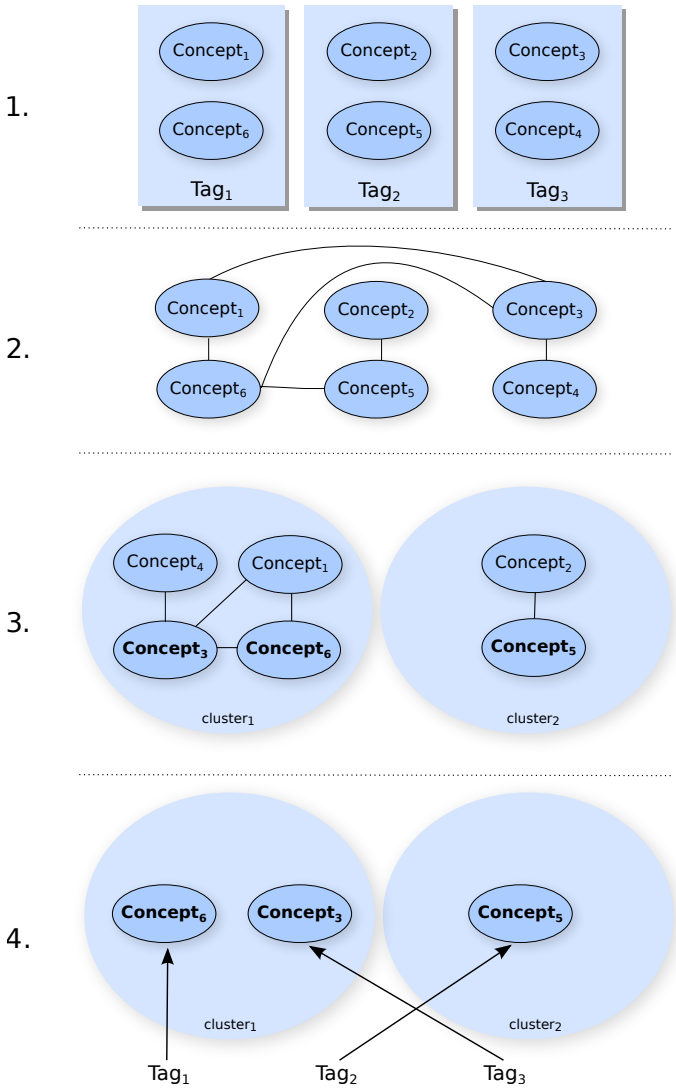


Figure 4.1: Schematic overview of the disambiguation process.

their corresponding candidate senses. The graph is expanded by adding semantic edges and nodes from a thesaurus. The selection of the most appropriate sense per word occurrence is then made through the use of graph processing algorithms that offer a degree of importance among the graph vertices." (Tsatsaronis et al., 2010, p.184)

My approach relies primarily on the structural information that is available in a graph to

select the best concept for a term. The disambiguation algorithm makes use of two sources of data from DBpedia to add 'semantic edges' between concepts. More specifically, it uses the *wikilinks* and *category* datasets.

Wikilinks A large part of DBpedia consists of links from one Wikipedia article to another, i.e. *wikilinks*. Wikilinks are untyped (associative), i.e. the type of relation between the two reference concepts that a link signifies is unknown. This lack of information is also characteristic of the lightweight semantics in folksonomies. Folksonomies also lack a formal conceptual structure and specific relations, e.g. tags in Social Media have neither explicit links between concepts and terms nor describe formal relations between terms. Any system built for addressing lightweight semantics for Social Media resources should therefore not rely on (static) highly structured external resources, because they usually lag behind a community's conceptualization (see chapter 3 for details). Instead, systems that target Social Media should embrace the flexibility and lightweight modeling as captured and described through tagging and other collaboratively created resources. It is for this reason that the proposed approach to disambiguation is intentionally limited to 'knowledge poor' data from DBpedia.

The wikilinks between all possible concept pairs (Cartesian product of all candidate concepts) are retrieved. Each candidate concept is added as a node and all wikilinks that exist between any pair of candidate concepts are added as *unweighted edges* to the graph.

One could ask why we do not also include indirect associative links and more information about categories. The reason is as follows: the algorithm mostly depends on untyped auto-associative links between DBpedia concepts. This gives a strong preference for strongly coherent sets of concepts, i.e. clusters of concepts with a high number of interconnecting edges. The alternative would be to sort the clusters by the number of concepts that they contain, but exploratory experiments suggest that this leads to poorer results due to the fact that the graph of wikilinks in DBpedia, the sense inventory, itself is dense, i.e. the number of edges divided by the number of nodes is high (for details see chapter 2 section 2.3.1). This in effect means that the graph clustering algorithm potentially creates clusters of concepts that are only remotely related, because of the graph density of DBpedia. It is for this reason that the clusters are sorted by the average degree of their concepts instead of the number of individual concepts in a cluster. Alternatively one could say that the clusters are sorted by their density. A higher cluster density means more edges in relation to the number of concepts and thus a greater certainty that the concepts in the cluster are actually related. This observation is compatible with the report that symmetric edges between concepts are good indicators for strong relatedness (Fogarolli, 2009).

Categories The wikilink-structure is not always sufficient to end up with a richly interconnected graph. A graph with a reasonable amount of edges is required for the clustering and concept filtering phase of the algorithm (section 4.3). It is for this reason that category assignments in DBpedia are also used to enrich the graph of concepts with additional edges.

DBpedia contains category assignments for virtually all DBpedia concepts. Additionally, a category can be part of one or more other categories via a subsumption relation. The subsumption relations together constitute a semi-hierarchal tree of categories. In order to enrich

the disambiguation graph, my algorithm determines whether any combination of candidate concepts shares a category directly or indirectly.

The algorithm first determines all the shared categories up to some distance n in the hierarchy of DBpedia categories, i.e. two concepts 'share' a category if both concepts s_i and s_j can be reached via a path starting from that category of length n or less. The shared category is then added to the graph as a new node (concept) and is directly connected to the two concepts that it contains.

4.3 Clustering and concept filtering

The goal of the third step is to identify the patterns in the graph, i.e. to quantify the structure of the graph such that the algorithm can act on it. At this point, the graph contains all of the information required for the remaining steps of the disambiguation algorithm, i.e. the sense inventory is no longer required. Two operations are performed in this step: *clustering* followed by *filtering*. The *clustering* step identifies groups of nodes (clusters) that are related in some way. The *filtering* operation removes the superfluous concepts from each cluster. Finally, this section also includes a theoretical discussion on *graph density* and how this affects the clustering and filtering operations presented in this section.

Clustering As previously mentioned, the goal of the clustering step is to identify groups of related concepts in the graph.

We consider two concepts in a sense inventory as more related if there are more edges between them. As a consequence, nodes for highly related concepts in the graph have many paths over edges between them. Clustering aims to identify groups of concepts that are closely related and isolate them from (groups of) concepts that are not or much less related to them. For example, the concepts `dbpedia:Ajax_(mythology)` and `dbpedia:Amsterdam` are not related and thus should not end up in the same cluster. This can be achieved by taken into account that there are few, if any, connections between these two concepts. In contrast, the concepts `dbpedia:AFC_Ajax` and `dbpedia:Amsterdam` are closely related and should become part of the same cluster, which can be achieved by taking into account the many connections between them. The cluster assignment is only based on the amount of paths between nodes in the graph.

The graph of interconnected concepts is processed by a graph clustering algorithm (Girvan and Newman, 2002) (Figure 4.1, step 3) such that clusters of concepts are established. The graph clustering algorithm exploits the global 'community structure' of the graph as illustrated in figure 4.2. Figure 4.2 contains a prototypical example of community structure; three clusters of densely connected nodes (circles with solid lines), with a much lower number of edges (gray lines) between the clusters. The nodes within a cluster are thus more densely connected to other nodes within the same cluster than to other nodes in the graph.

Central to the graph clustering algorithm from Girvan and Newman (2002) is the notion of *edge betweenness centrality* (Cuzzocrea et al., 2011, p.5). Edge betweenness centrality assigns a value to an edge that describes how many shortest paths in the graph use that edge. More formally; let $G =< S, E >$ be a connected undirected graph G with concepts S, edges

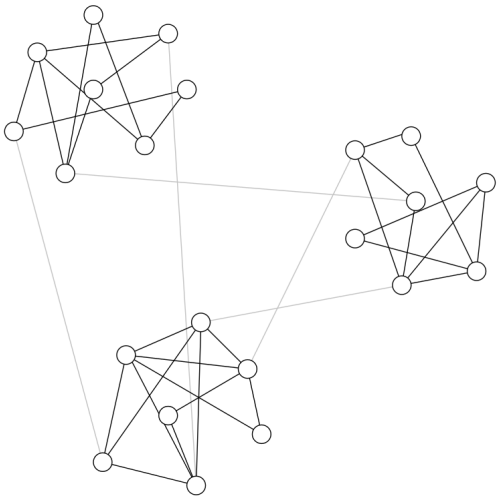


Figure 4.2: A prototypical graph with community structure. Figure from (Girvan and Newman, 2002, p.1).

1. Calculate the betweenness for all edges in the graph (equation 4.1).
2. Remove the edge with the highest betweenness value.
3. Recalculate betweenness values for every edge affected by the removal
4. Repeat from step 2 until the specified number of edges is removed[3].

Table 4.1: High level overview of the Girvan–Newman algorithm for clustering graphs with supposed community structure.

E and $s_i \in S$ and $s_j \in S$ two concepts in G, respectively. Let P denote the set of pairs (s_i, s_j) where one or more paths exist connecting s_i with s_j. Let $\delta_{s_i s_j}$ denote the number of shortest paths between concepts s_i and s_j. Let $\delta_{s_i s_j}(e)$ denote the number of shortest paths between s_i and s_j which go through edge $e \in E$. Edge betweenness centrality of an edge e can then be defined as in eq. (4.1).

$$EdgeBetweenness(e) = \sum_{(s_i, s_j) \in P} \frac{\delta_{s_i s_j}(e)}{\delta_{s_i s_j}} \tag{4.1}$$

The algorithm by Girvan and Newman (2002) assigns nodes that are part of a community structure to the same cluster and is described in table 4.1. Informally the concept of community structure can be understood by imagining that the edges that connect clusters resemble

[3]The number of edges that is removed affects performance of the disambiguation algorithm. Various values have been explored and will be discussed in section 5.

'highways' that link tightly bundled groups of city-roads. Each city is internally well connected, but virtually all shortest paths between cities have to include the highways. These highways thus have a high *edge betweenness value* and their removal results in clusters of nodes with edges (for each city). The resulting clusters were originally referred to as 'communities' and the graph is thus said to have a community structure (Girvan and Newman, 2002). More formally, the gradual removal of edges with high betweenness values results in clusters of concepts based on their 'community structure', i.e. groups of concepts with a large number of connections to nodes within their own cluster and a limited number of connections to concepts outside of their own cluster. Nodes are well connected within their own cluster, but there are few choices regarding the selection of edges that are connected to nodes outside of each cluster. This means that the edges that connect clusters are part of more shortest paths and thus have a higher betweenness value. In the extreme case of no community structure at all, i.e. singleton clusters, the disambiguation algorithm still functions correctly. However, in this case it failed to identify any coherence between concepts in the graph in order to improve performance. This situation can be simulated by removing all of the edges in the graph, i.e. by setting the "specified number of edges" in table 4.1 to the total number of edges in the graph. In this case the node with the highest number of edges to any candidate concept in any singleton cluster, is selected first for disambiguation. Although this does negatively impact the disambiguation accuracy the effect is relatively small. See section 5 for more information.

Each cluster of concepts (community), in the context of a disambiguation task, is assumed to model a coherent set of concepts, i.e. a set of interrelated concepts. This clustering step needs to be performed, because only considering isolated groups of concepts will not work in all cases. Consider for example three large densely connected clusters which are connected through a small amount of edges as shown in figure 4.2. Determining the clusters via the identification of isolated groups of concepts will not work, because every node in the graph is reachable from any other node. Many graphs thus require some method of determining which edges are indicative of the clusters themselves and which correspond to connections between clusters.

The use of the graph clustering algorithm of Girvan and Newman (2002) is an improvement over the previous work reported in Monachesi and Markus (2010b). Both the algorithm in Monachesi and Markus (2010b) and the one presented in this section, cluster the concepts in the graph. In Monachesi and Markus (2010b) this clustering is performed in two steps. The first step is the application of a graph-layout algorithm (Fruchterman and Reingold, 1991). Such an algorithm assigns specific coordinates to nodes which leads to a better visualization. The layout algorithm made nodes which were densely connected appear close together and others further apart. The second step in Monachesi and Markus (2010b) was to cluster using the improved visualization of the graph, e.g. groups of concepts that were close together would appear in the same cluster. The improved clustering method in this section reduces these two steps into one. The basic idea of the simplified clustering is not to cluster using a layout specific representation of the graph using 2D-coordinates, but to use the graph-structure (the nodes and edges) directly (Girvan and Newman, 2002). In the improved graph clustering approach, presented in this section, clusters are established based on the structural properties of the graph instead of an intermediate graph transformation that could introduce additional errors.

Filtering After assigning each concept to one cluster it becomes necessary to remove concepts from each cluster. This must be done because there might be multiple concepts for the same tag in the same cluster, because they are related to other concepts. However, recall that the disambiguation algorithm can only assign a single concept to each tag.

In order to identify which concepts must be removed from the cluster, the degree of each concept is calculated (fig. 4.1, step 3, nodes with bold text are retained). Degree is defined as the number of edges connecting a concept to other concepts. It can be interpreted as a measure of how 'important' or 'central' the concept is.

The degree values are used in step 3 of my disambiguation algorithm to reduce the number of concepts in a cluster. Some concepts need to be removed from their cluster, because they are 'competing' to get assigned to a tag. When there are multiple concepts for the same tag in the same cluster then only the concept with the highest degree is eligible. If there are multiple competing concepts with the exact same degree one of the competing concepts is randomly selected. All other concepts for that tag are removed from that cluster. The intuition behind this process is that each cluster now contains the maximally coherent meanings for the largest number of terms.

4.4 Concept selection

The final step associates the terms with the proper concepts by considering the clusters in a specific order (see fig. 4.1, step 4). This is accomplished by analyzing how many relations are available per concept and the clusters that have been established in the previous step. The most important aspect of this step is the order in which the clusters are processed.

Determining the optimal order in which the clusters are to be considered is more complicated than it may appear. It becomes even more difficult when these clusters (partially) overlap. For example, what happens when there are two clusters that both provide concepts for the same tag? Which cluster should be preferred? And why? I will motivate the order in which the clusters are processed using the structure of the graph. This entails that the order in which clusters are considered is relevant, because it influences which concepts will be preferred in the case of multiple candidate concepts for an ambiguous tag.

First, the clusters are sorted by the original average degree or, when equal, the total number of concepts. Second, the cluster that has the highest average degree, i.e. the highest number of edges in proportion to the number of nodes, is selected. Once a cluster is selected, its concepts can be used to disambiguate one or more tags. Once a tag is disambiguated all concepts belonging to that tag and edges related to that concept are removed from the graph, i.e. from all clusters. This means that some of the remaining clusters become smaller or even completely empty.

The remaining clusters with concepts belonging to tags that have not been disambiguated thus far are subsequently sorted by the number of *remaining* concepts. The rationale for not simply selecting the next-largest cluster without removing the concepts for the tags already disambiguated, is that this cluster probably contains conflicting concepts. Disregarding concepts belonging to already disambiguated tags retains truly complementary clusters instead

of conflicting ones. If any terms are still not disambiguated at this point the most frequent sense in the reference repository is selected as a reasonable default.

This concludes the description of the operation of the disambiguation algorithm. At this stage each tag has been associated with a single reference concept, i.e. the tag has been disambiguated. The following section will provide a detailed description of how the algorithm operates on a real example with actual data from the DBpedia reference repository.

4.5 Example

This section presents a detailed example of the disambiguation algorithm in action. These steps are graphically illustrated in fig. 4.3

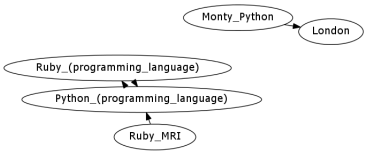
Consider a resource that has been annotated (tagged) with two tags; 'python' and 'ruby'. Both ruby and python are ambiguous tags, where 'ruby' could refer to things such as 'an expensive jewel' or 'a programming language' and python could mean 'a specific species of snake' or 'a programming language'. Actually 'ruby' can refer to 70 different concepts according to DBpedia version 3.7 and 'python' can refer to 18 different concepts, but I will keep the example simple. All the possible concepts for all the two tags are retrieved from the reference repository and serve as nodes in a graph each denoted by its unique URI. Each wikilink is added as an edge between the concept nodes in the graph. The result of these operations is shown in fig. 4.3a. Note that fig. 4.3 only contains nodes with at least one incoming or outgoing edge.

For layout reasons, each of the concept URIs in fig. 4.3a is not prefixed by `dbpedia`, but the reader should interpret them as if they do have that prefix. An additional search for shared categories between all possible concepts for the tags 'ruby' and 'python' yields more information which gets added to the graph as additional nodes and edges. The results of this second step is illustrated in fig. 4.3b.

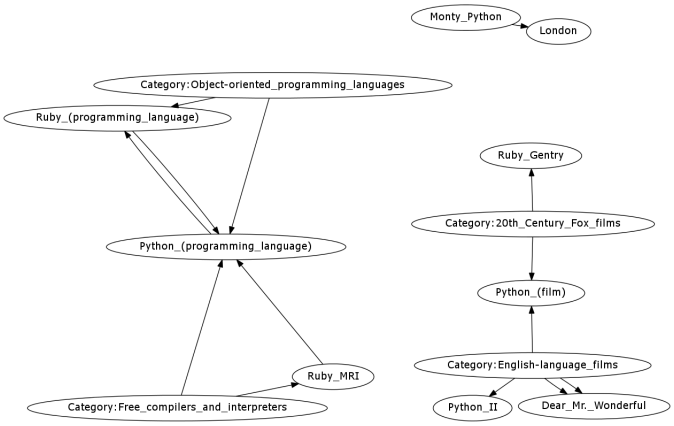
Categories are added as new nodes with edges to their respective concepts. However, category-nodes are not associated with any term and are therefore not eligible as a disambiguation result. The resulting graph is then clustered into clusters of concepts by exploiting its community structure. This gives rise to a large number of clusters. For simplicity, only those clusters that have more than one concept are listed. The clustering results are displayed in fig. 4.3c.

For each concept in a cluster the degree is calculated by counting the edges. The clusters are then pruned by removing all the concepts in a cluster linked to a term except the one with the highest degree. Categories are automatically removed, because they are not explicitly linked to any term. This results in three sets of concepts as shown in fig. 4.4.

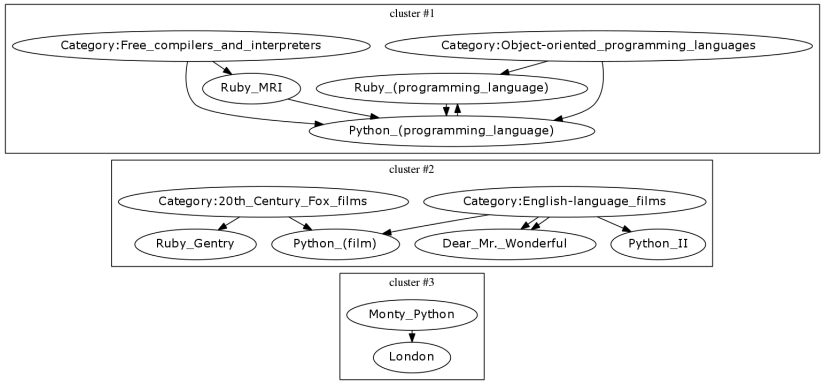
These sets of concepts are sorted by each clusters' average degree before pruning, i.e. nodes and edges that have been removed still count towards a cluster's overall degree. Recall that the overall degree of a cluster is used as a measure of coherence. Finally, the cluster with the highest average degree, i.e. cluster (1) in fig. 4.4, is selected and its concepts are linked to the input terms, thereby disambiguating the terms 'ruby' and 'python'. The algorithm then terminates, because no other terms are available that require disambiguation.



(a) The initial graph with concepts that have at least one edge.



(b) Graph extended with shared categories.



(c) Clusters identified in the graph.

Figure 4.3: Graphical depiction of the first three steps of the disambiguation algorithm.

1. {Python_(programming_language), Ruby_(programming_language)}
2. {Ruby_Gentry, Python_(film), Dear_Mr._Wonderful}
3. {London, Monty_Python}

Figure 4.4: Sets of concepts that remain after pruning candidate concepts that compete for the same term in their respective cluster.

tag	# concepts	selected tag sense
python	18	dbpedia:Python_(programming_language)
scripting	1	dbpedia:Scripting_language
javascript	1	dbpedia:JavaScript
programming	1	dbpedia:Computer_programming
opensource	21	dbpedia:Open_source
apache	44	dbpedia:Apache_HTTP_Server
java	65	dbpedia:JavaScript
framework	6	dbpedia:Software_framework

Table 4.2: Tagging action disambiguation example. The numbers after each tag refer to the initial number of candidate concepts

Table 4.2 presents an example of the successful disambiguation of a tagged resource annotated with a number of tags.

The previous example has implicitly assumed that all the concepts for the input terms are in a single cluster. It can also happen that this assumption does not hold. This happens, for example, when two clusters represent disjoint domains, i.e. multiple topics, multi-disciplinarity. Appendix B contains a considerably more complicated example that highlights other aspects of the disambiguation algorithm. It is an extension of the example that has been presented in this section. Appendix B illustrates what happens in a multi-disciplinary situation using four terms: [ruby, python, mona, leonardo], i.e. the domains of computing and art.

In summary, the disambiguation algorithm presented in this chapter is able to assign senses from a sense inventory to terms from Social Media. It accomplishes this by harnessing wikilinks and category assignment extracted from DBpedia to construct a graph and by using graph-based disambiguation. Each term is finally disambiguated by associating it with a concept from a reference repository that serves as a sense inventory.

5 Evaluation

The evaluation of the disambiguation methodology consists of two related, but distinct tasks.

First, in section 5.1, the objective is to assess the performance of the disambiguation algorithm on a number of tags. In order to determine this, a number of ambiguous tags and frequently co-occurring tags has been gathered. The accuracy of the disambiguation algorithm has been compared to the well-known LESK algorithm (Lesk, 1986), which is often used as a baseline in the evaluation of disambiguation algorithms, and other state of the art approaches.

Second, the impact of disambiguation on ontology enrichment is determined in section 5.2. Recall that chapter 3 presented Social Ontology Enrichment. The tags considered for enrichment in chapter 3 were forced to be unambiguous in order to achieve reliable results. However, this means that ambiguous terms that might be relevant in the context of the domain ontology are discarded. An extended evaluation that does use the disambiguation algorithm as part of the ontology enrichment process is therefore presented in section 5.2.

5.1 Tag disambiguation

This section presents an evaluation of the disambiguation algorithm on a manually constructed gold standard. First, I will present the evaluation methodology as well as the baselines and algorithms that are used in section 5.1.1. Second, in section 5.1.2 the experimental results are contrasted to those of the state of the art and related systems.

5.1.1 Setup

Although some related publications on tag disambiguation are available (Andrews et al., 2010; Tesconi et al., 2008; Garca-Silva et al., 2009) surprisingly none of them mention an extensive evaluation set. Most refer to a few limited examples which are illustrated in the paper, but more extensive standardized evaluation sets are to be preferred. Having such a dataset adds confidence in the performance of the disambiguation algorithm on examples other than the anecdotal examples in the paper. Additionally, it enables one to compare different disambiguation algorithms on standardized test sets as is common with the SemEval series of tasks (Manandhar et al., 2010). Not having such a test set makes direct comparisons of disambiguation algorithms difficult, if not impossible.

SemEval/Senseval data sets constitute the standards to be used for evaluation of disambiguation performance on a variety of tasks. However, most of these tasks are sentence-oriented. Disambiguation approaches that only rely on content words, such as nouns, for their disambiguation context reduce sentences to a bag of content words. These bags of content words are however different from tags in Social Media with respect to the number of nouns, abbreviations and term usage, which makes them unsuitable for validating tag disambiguation performance. Additionally, SemEval/Senseval datasets are strongly oriented towards WordNet (Miller, 1995) and therefore only include references to WordNet specific senses. However, in my disambiguation approach the senses come from DBpedia as opposed to WordNet with different coverage characteristics, e.g. many domain specific senses are available in DBpedia, but not in WordNet. There are tasks within SemEval-2010, though, that focus on domain specific documents annotated with WordNet senses (Agirre et al., 2010). A survey

of the literature did not reveal any publicly availably pre-existing evaluation set regarding tag disambiguation in the context of Social Media.

Due to the aforementioned reasons, I have decided to construct an evaluation set for tag disambiguation myself. This set consists of 633 tag assignments as part of 224 Tagging actions (sets of tags). The tag assignments have been gathered by selecting a number of ambiguous terms and finding a set of tags for a resource on the social bookmarking site `delicious.com`, the video sharing sites `youtube.com` and `vimeo.com` that matches one of the possible senses. The tags have been gathered by executing keyword search requests mostly for the terms described in chapter 5, table 5.1. Each tag assignment contains one or more ambiguous tags and a preferred sense for one or more tags. Each of the tag sets in the test set constitutes a set of content words; a bag-of-words partially enriched with a preferred DBpedia concept for one or more ambiguous target tags. Only ambiguous tags have been assigned a sense, i.e. to prevent an artificial increase of the disambiguation accuracy through the inclusion of unambiguous terms that have only one sense.

In order to show the effectiveness of my disambiguation algorithm, I have compared it to several well-known disambiguation algorithms on the same testset. In the SenseEval/SemEval set of tasks the baseline algorithm is either the 'most frequent sense'-algorithm or the simplified LESK algorithm. The 'most frequent sense'-algorithm selects the most frequent sense based on the WordNet's word sense frequency information. The simplified LESK algorithm is also often used as a baseline in the literature (Kilgarriff and Rosenzweig, 2000). It selects senses by maximizing lexical overlap between the sense and a term's context and defaults to the most frequent sense if there is no such overlap.

A minor modification of the simplified LESK algorithm (Lesk, 1986; Kilgarriff and Rosenzweig, 2000) has been used as a baseline for assessing the disambiguation performance. The simplified LESK-algorithm described by Vasilescu et al. (2004), shown in listing 4.1, selects the best sense via lexical overlap of the gloss or definition of the candidate sense with the context of the target term. The context of a tag consists of other tags that belong to the same Tagging-action. The sense whose gloss or definition has the largest overlap with the context of the target term will be chosen as the proper sense. Function words are either removed or assigned near-zero weights to reduce their impact.

By default, the most frequent word sense will be selected by the simplified LESK algorithm (Kilgarriff and Rosenzweig, 2000) as a smart default in order to deal with the situation where the overlap between all possible glosses with the context is zero or exactly the same. This is a better strategy than just choosing a word sense at random, i.e. the original behavior of LESK, because the most frequent word sense has a higher probability of being correct than a random sense. However, this requires the ability to quantify the frequency of a sense in the sense repository.

A different method was required to select the most frequent word sense for DBpedia, because the original algorithm uses a corpus and language statistics to determine it. The most frequent sense for DBpedia is determined by selecting the DBpedia concept with the highest degree of page links (wikilinks). The original simplified LESK algorithm assigns a weight to each term based on the TF-IDF score for that term given a corpus. My disambiguation approach does not use any corpus. I have therefore replaced the term weights with an English stop words list used by the popular MySQL database software version 5.1. Function words and common

determiners receive a very low weight using TF-IDF. The list of stop words achieves a similar effect by removing them from the BOW altogether. This addresses some of the drawbacks of not having TF-IDF scores for terms.

All these changes constitute a considerable modification of the original LESK and simplified LESK algorithms in order to apply them to DBpedia. It is for this reason that I will refer to the DBpedia-specific version as the 'Simplified LESK for DBpedia algorithm'.

```
1  def simplified_lesk(word, context)
2      best_sense = most_frequent_sense(word)
3      max_overlap = 0
4
5      for sense in getAllSenses(word) do
6          signature = get_definition_and_or_gloss(sense)
7          signature = remove_stop_words(signature)
8          overlap = |signature ∩ context|
9
10         if overlap > max_overlap
11             max_overlap = overlap
12             best_sense = sense
13         end
14     end
15     return best_sense
16 end
```

Listing 4.1: The Simplified LESK for DBpedia algorithm

5.1.2 Evaluation

As mentioned in the previous section the algorithm presented in this chapter is compared to several well-known baselines. Table 4.3 lists the abbreviation used to refer to each of the evaluated disambiguation algorithms. Table 4.5 lists the disambiguation accuracy for the different algorithms listed in table 4.3 and, in addition, some related systems and evaluation sets. The evaluation focuses on the *accuracy* of the proposed disambiguation algorithm, because every word sense is eligible in my algorithm and there is no confidence value or a similar parameter that can be used to influence recall. This makes a comparison to other approaches, such as that reported by Fader et al. (2009) difficult, because they only include recall-precision-plots.

The performance of each of these four algorithms has been determined on the manually created tag set. Additionally, the performance is contrasted with various other approaches which are listed in table 4.4. Note that the accuracy for the disambiguation algorithm in this chapter is only for ambiguous terms.

On the test set, the most frequent sense baseline (MF) performs poorly, i.e. it achieves an accuracy of 0.47. This is likely due to several factors. First, a different method is used for determining the sense frequency, incoming and outgoing pagelinks, instead of WordNet's sense frequency. Second, my evaluation set includes a relatively large amount of domain

[4]The average over 1000 runs of the Random sense selection algorithm is reported.

Abbreviation	Description
Random	Select a random sense for a term from the reference repository[4]
MF	Always select word sense with highest degree
MF'	Use the most frequent sense accuracy rate reported in a study
SL	The Simplified LESK for DBpedia algorithm (listing 4.1)
GD	Graph-based disambiguation without shared categories.
GDS	Graph-based disambiguation with shared categories
SM	Spotlight Mixed as described in (Mendes et al., 2011)
CZ	The algorithm described in (Cucerzan, 2007)

Table 4.3: The eight different types of disambiguation algorithms that are used in the evaluation and their abbreviations.

Current		(Mendes et al., 2011)		(Cucerzan, 2007)		(Cucerzan, 2007)-tagsets		
Algorithm	Acc.	Algorithm	Acc.	Algorithm	Acc.	Algorithm	Acc.	↑ Acc.
Random	0.13	Random	0.18			Random	0.08	0.55
MF	0.47	MF'	0.55	MF'	0.51	MF	0.72	0.84
SL	0.42					SL	0.72	0.84
GD	**0.84**	SM	0.81	CZ	0.91	GD	0.78	0.87

Table 4.4: Comparison of the accuracy achieved by state of the art disambiguation algorithms based on DBpedia and/or Wikipedia. The abbreviation *Acc* refers to the accuracy of ambiguous terms. ↑ *Acc* refers to the accuracy based on both ambiguous and unambiguous terms.

specific terms and senses that are overall less frequent in the general discourse. For instance, the term 'apache' in WordNet 3.1 does not include the Computer Science sense of the term (web server software), but only senses related to Native Americans and Parisian gangsters.

The results in table 4.4 show that the disambiguation algorithm presented in this chapter achieves an accuracy of 0.84 on the test set which is an excellent result. A comparison with two approaches previously discussed in section 2 seems to suggest that this is competitive, if not better, than some of the state of the art. It should be noted that both Mendes et al. (2011) and Cucerzan (2007) manually extract all surface forms from Wikipedia, whereas my algorithm only uses the standard data available in DBpedia, i.e. disambiguation links and redirects, which is significantly more limited in coverage. I have included the performance of a random-sense-algorithm to quantify the amount of polysemy in each dataset when possible, i.e. it becomes increasingly unlikely that a random sense is correct as the number of senses for a term increases. The accuracy in table 4.4 shows that the amount of polysemy is comparable to that of other datasets in the literature.

The Cucerzan (2007)-tagsets in table 4.4 refers to the news dataset used by Cucerzan (2007) that has been converted to 20 tagsets that accommodates the changes in DBpedia 3.8, i.e. resource identifiers have been updated when required. Terms that could not in any way be used to identify the right reference concept through DBpedia via either disambiguation links or redirects were removed. Additionally, the accuracy reported by Cucerzan (2007) for this corpus considered both ambiguous and unambiguous terms[5]. This unintentionally inflates accuracy due to the fact that the disambiguation of a term with only one sense is guaranteed to be correct. It is also unclear whether Shen et al. (2012) reports the accuracy for only ambiguous terms or if the accuracy includes unambiguous ones as well. I have therefore included additional results in the table 4.4 (column $\uparrow$ *Acc*), that include unambiguous terms in the overall accuracy.

The results in table 4.4 show excellent results on the new (Cucerzan, 2007)-tagsets dataset. More specifically, it achieves an accuracy of 0.87 when as compared to the the reported accuracy in Cucerzan (2007) of 0.91. My results are slightly poorer, but recall that the GD algorithm only makes use of DBpedia as opposed to the use of the full Wikipedia in Cucerzan (2007). Additionally, the test set from Cucerzan (2007) is very limited, with only 20 examples. The good results on the (Cucerzan, 2007)-tagsets dataset shows that the proposed approach is not necessarily specific to tag disambiguation, but can also be used to disambiguate to a set of entities extracted from news stories. No direct comparison to the data used by Mendes et al. (2011) was possible due to underspecification of the dataset in both the paper and the supplementary material provided on their website. Note that both Cucerzan (2007) and Mendes et al. (2011) use the full text of news articles in their evaluation, whereas *GD* is limited to just sets of content words.

Table 4.5 and fig. 4.5 describes variations on the parameter 'edges' for the GD-algorithm. Removing edges with a high betweenness value uncovers clusters of nodes with community structure (Girvan and Newman, 2002). In the context of the disambiguation algorithm, the

[5]The actual paper is not clear about this, but private correspondence with the author has confirmed this to be the case.

[6]Relative amount of edges with the highest edge betweenness removed by the community-based clustering algorithm (Girvan and Newman, 2002)

Algorithm	Category distance	Edges[6]	Accuracy
GD	0	0%	0.84
GD	0	5%	0.84
GD	0	10%	0.84
GD	0	20%	0.84
GD	0	40%	0.82
GD	0	90%	0.75
GDS	1	0%	0.79
GDS	1	5%	0.79
GDS	1	10%	0.78
GDS	1	20%	0.76
GDS	1	40%	0.74
GDS	1	90%	0.76
GDS	2	0%	0.67
GDS	2	5%	0.67
GDS	2	10%	0.69
GDS	2	20%	0.67
GDS	2	40%	0.63
GDS	2	90%	0.67

Table 4.5: Accuracy results for various parameters of the disambiguation algorithm.

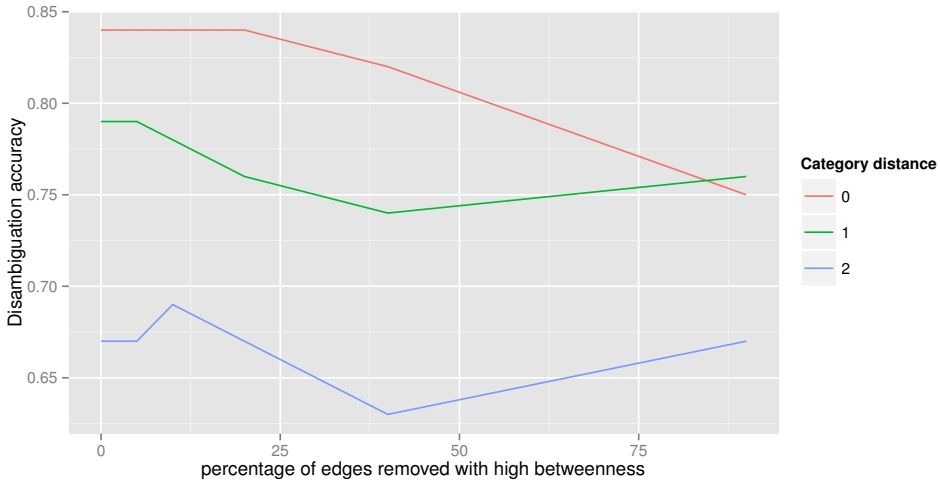


Figure 4.5: Accuracy results for various parameters of the disambiguation algorithm.

community structure is an indication of the coherence of a set of related senses. Large clusters, i.e. a large coherent set of senses, will get preferential treatment over small clusters. The rationale is that these large clusters are more likely to be correct than smaller ones, because they represent a larger amount of coherence between the terms. One would expect that this feature affects accuracy considerably. However, if we remove up to 90% of the edges, and as a result end up with mostly singleton clusters, the performance is still competitive at 0.75 compared to 0.84 when only 0-20% of the edges is removed. This suggests that the coherence of a set of senses only contributes to an increase in accuracy of 0.09 absolute. Although still considerable, the contribution of sense coherence, as measured through community-based clustering, is not as large as one might expect.

The disambiguation accuracy seems to be quite robust with respect to the number of edges that are removed by the community-based clustering algorithm. More specifically, the accuracy does not change when removing either 0% up to 20% of the edges with high betweenness when no shared categories are included. This suggests that the graph is already well connected and the community is supported by few edges with high betweenness among individual clusters.

Another striking result in table 4.5 is the fact that the inclusion of shared categories in the graph has, overall, a negative effect on the accuracy. One would expect that the additional information provided by the shared category should allow the community-based clustering to more reliably identify clusters of coherent senses. However, the impact of coherence is quite limited, as previously discussed in section 4.3, page 108. It seems to be the case that the shared categories add too many edges in the graph that do not really signify a semantic association, i.e. a false positive. This trend continues with setting the maximum allowed distance in the category hierarchy from 1 to 2. One would expect that a larger amount of edges with high betweenness values would need to be removed from the graph, but the results in table 4.5 show no clear performance improvement for increased removal of edges. I therefore have to conclude that the poor signal-to-noise ratio of the shared categories in the disambiguation algorithm does not lead to an increase in disambiguation accuracy.

Navigli (2009) discusses performance results of disambiguation algorithms of SenseEval-2, SenseEval-3 and SenseEval 2007. Performance of the best unsupervised approaches is consistently 10 to 15% lower than the supervised approaches. The best performing unsupervised disambiguation algorithm on the “Coarse-grained All Words”-task of SenseEval-2007 achieved an accuracy of 70.2%. In this task the baseline, i.e. selecting the most frequent sense, achieved an accuracy of 78.9%. The task ”All-words word sense disambiguation on a specific domain” of SemEval-2010 outperforms most disambiguation algorithms for the English language with a modest reported baseline precision of 0.505 (Agirre et al., 2010).

It is difficult to compare the results of disambiguation algorithms evaluated on WordNet due to the considerable difference in the average polysemy between WordNet (2.79 for WordNet 3.0[7]) and DBpedia (7.69 for the test set used in this chapter and 13.23 for (Cucerzan, 2007)-tagsets). This affects recall, precision and accuracy of disambiguation algorithms. For example, the most frequent sense will perform reasonably well on WordNet, but will be much worse on DBpedia due to the increase in polysemy. Reported disambiguation performance should therefore be adjusted for the polysemy of the data that is taken into consideration. In

[7] `http://wordnet.princeton.edu/wordnet/man/wnstats.7WN.html` retrieved 23-05-2013

this evaluation, this is accomplished by including a random sense, i.e. a random sense will perform well with low polysemy and poorly with high polysemy in the data set.

It is clear that my disambiguation approach, based on unstructured associative links between DBpedia reference concepts, achieves satisfactory performance on tag disambiguation. The clustering approach used by my disambiguation algorithm clearly outperforms the LESK algorithm which is based on lexical overlap and the most frequent sense algorithm. The performance of my unsupervised algorithm is similar to the performance of other state of the art algorithms for unsupervised disambiguation. However, it is conceptually transparent, elegant and does not rely on the extraction of surface forms from Wikipedia, but can use standard DBpedia datasets directly.

5.2 Ontology enrichment

Ontology enrichment was the main focus of chapter 3. However, the ontology enrichment pipeline evaluation (chapter 3, section 6) presented in that chapter did not contain an evaluation of ontology enrichment using disambiguation. Only ontology enrichment without the use of a disambiguation algorithm was presented previously in that chapter. This section presents the results from the ontology enrichment process while taking disambiguation into account. The results of the conceptual enrichment are presented in table 4.6, alongside the original results previously presented in chapter 3 in order to make the impact of disambiguation clear.

SOE added 289 (category distance 3) and 251 (category distance 2) new concepts without any disambiguation, but with filtering. In contrast, 1123 concepts were added for category distance 3 with disambiguation enabled and 1048 for category distance 2. This more than triples the number of concepts available for enrichment due to disambiguation. Disambiguation allows for the enrichment of ambiguous tags as represented by reference concepts, such as `dbpedia:Android_software`, `dbpedia:Apple_Inc.`, `dbpedia:Python_(programming_language)`, `dbpedia:Ajax_(programming)` and `dbpedia:Ruby_on_Rails`. These concepts were not eligible for enrichment without disambiguation.

Recall, that ontology enrichment without disambiguation cannot retrieve a concept from a reference repository if the lexicalization from the domain ontology is ambiguous. For this reason, the ontology enrichment pipeline without disambiguation has to omit 870 terms because of their ambiguity. This includes, tags such as `linux`, `gmail`, `uml`, `ide`, `rest` and `ubuntu`. The results in this section clearly show the advantage of using disambiguation in ontology enrichment with respect to domain coverage. The negative impact on the quality of the enrichment appears relatively small when one compares the relative amount of correct statements added to the ontology with and without disambiguation.

A rough estimate of the expected enrichment quality that includes ambiguous terms can be obtained by multiplying the enrichment quality for unambiguous terms (0.94) with the average disambiguation accuracy reported in table 4.4 (0.84). This results in an estimated enrichment quality of 0.79. This is close to the actual enrichment quality for a category distance of two presented in table 4.6 of 0.81 for filtered results and 0.78 for unfiltered results. The increase in available lexicalizations follows a similar trend as the conceptual enrichment as previously reported in chapter 3, section 6.2.

	No disambiguation				With disambiguation			
	Category distance 2		Category distance 3		Category distance 2		Category distance 3	
	Associative	Ontological	Associative	Ontological	Associative	Ontological	Associative	Ontological
Acceptable	470 (59%)	230 (94%)	444 (62%)	342(75%)	622 (43%)	696 (81%)	434 (40%)	1073 (66%)
Unacceptable	133 (41%)	4 (6%)	267 (38%)	112 (25%)	812 (57%)	168 (19%)	653 (60%)	545 (34%)
Total	603	234	711	454	1434	864	1087	1620

(a) With filtering

	No disambiguation				With disambiguation			
	Category distance 2		Category distance 3		Category distance 2		Category distance 3	
	Associative	Ontological	Associative	Ontological	Associative	Ontological	Associative	Ontological
Acceptable	503 (59%)	234 (91%)	446 (59%)	344 (62%)	628 (43%)	706 (78%)	437 (39%)	1090 (55%)
Unacceptable	348 (41%)	23 (9%)	309 (41%)	215 (38%)	840 (57%)	203 (22%)	673 (61%)	903 (45%)
Total	851	257	755	559	1468	909	1110	1993

(b) Without filtering

Table 4.6: Enrichment results from the LT4eL-ontology for new concepts and relations. *Associative* refers to the use of the `ltfll:related` relation. *Ontological* refers to the use of either a shared category or a DBpedia specific property. These results have been obtained with DBpedia version 3.8 and only the first 10 cooccuring tags for a concept lexicalization using a variable category distance.

6 Conclusion

I have presented an effective disambiguation algorithm which builds on established tools and techniques for graph-based disambiguation. It makes use of socially derived sense inventories to disambiguate terms coming from Social Media. It can operate with loosely structured data and inconsistencies and is domain independent. The disambiguation approach takes advantage of simple graph-based measures such as degree and betweenness centrality thus exploiting the dense associative structure of DBpedia.

The disambiguation is competitive with the state of the art and in some cases surpasses it. It proposes a relatively simple and semantically transparent approach to graph-based disambiguation that can deal both with noisy reference repositories and complex disambiguation tasks such as tag disambiguation and ontology mapping. My approach is also much simpler than that of Mendes et al. (2011) through its use of standard DBpedia datasets as opposed to manual extraction of surface forms, while achieving better performance. This suggests that the use of redirects and disambiguation pages in DBpedia makes the use of surface forms in the original text redundant. It achieves an accuracy of 0.84 on a large set of tags and an accuracy of 0.78 (ambiguous terms only) or 0.87 (unambiguous and ambiguous terms) on a dataset of news stories derived from Cucerzan (2007).

Although, the performance is slightly worse than that reported in Cucerzan (2007) one has to take into account that the comparison with Cucerzan (2007) is only based on a very small sample of 20 examples. Additionally, there are several possibilities for improvement that are discussed in chapter 7. However, the main objective was to introduce a disambiguation method that makes use of noisy unstructured data similar to that of folksonomies. Even though it is intentionally limited to the knowledge-poor data from DBpedia its competitive disambiguation accuracy of 0.84 clearly shows that it has succeeded in this regard.

Finally, the disambiguation algorithm is also able to considerably increase the coverage of ontology enrichment, because it allows it to resolve ambiguous tags that it encounters during the enrichment process to reference concepts. The integration of disambiguation in the overall ontology enrichment methodology presented in chapter 3 is an excellent addition. As a result, the number of new reference concepts integrated with the domain ontology more than doubles, i.e. the coverage of the domain ontology increases.

Chapter 5

Semantic Search

1 Introduction

There is an ever increasing amount of information available on the Internet. The source of much of this information is increasingly shifting towards Social Media which have a low barrier of adoption for sharing and creating content. It is impossible to consume this information in its entirety. Users need search engines to cope with large amounts of online information. Common search engines take a list of terms, or a boolean expression over terms, as input and yield an ordered list of references to documents and other resources. Terms are called *keywords* in this context, so this type of search is called *keyword-based search*. Keyword-based search is currently the standard approach to search, but it has problems with ambiguous keywords (leading to many irrelevant search results) and poor keyword choice (leading to non-retrieval of relevant documents). Two important reasons for poor keyword choice are; (1) the user does not know what the correct terms are, or (2) the user does know the correct terms, but the community uses different terms in the Social Media where the search takes place. More sophisticated, approaches to search are needed to overcome these problems.

One of these alternatives is *semantic search*. Semantic search uses concepts instead of keywords. Semantic search has the potential to solve issues with ambiguous terms and poor keywords, but requires a domain ontology or other type of knowledge model. Semantic search holds the promise of performing much better than keyword-based search, because it automatically avoids issues with ambiguity. In addition, it makes it possible to automatically revise a query in a way that is fully automatic and more sophisticated than using keyword-based search systems. However, semantic search usually requires resources to be annotated with concepts instead of terms in order to work. In addition, semantic search frequently relies on a document's textual content for semantic analysis which rules out retrieving videos or images. In the application domain envisaged in this thesis, Social Media resources are not annotated with concepts, and we aim to retrieve video or image resources as well so there is still a gap to be bridged before semantic search can be applied in this context. In this chapter, I describe how this gap, the application of ontology-support semantic search in Social Media, is bridged.

I describe the outcomes of investigations into an approach to semantic search, called *SOSEM*, which is specifically designed for tag-based Social Media. SOSEM retrieves relevant resources from tag-based Social Media using a domain ontology and automatic semantic filtering of search results. SOSEM includes as one of its components an arbitrary keyword-based search engine. It retains the original order of the search results of this search engine, but improves them via post-processing using semantic knowledge. SOSEM is able to automatically identify whether search results are relevant in the context of a domain ontology without requiring resources to be semantically annotated with concepts. The semantic content of resources is identified even if ambiguous or different keywords are used. If a query is too poor, improved queries are automatically generated using a domain ontology's lexicon, concepts and ontological relations. In this way, the SOSEM approach addresses the problems for keyword-based search, i.e. ambiguity and poor keywords. The output of the ontology enrichment methodology, presented in chapter 3, can be used with SOSEM to automatically resolve vocabulary mismatches between users and communities.

SOSEM only relies on the tags that have been associated to resources in Social Media for its identification of semantic content. Both textual and non-textual resources, such as images or videos, can be retrieved using this strategy. It operates by automatically linking the tag annotations of a resource to concepts using a reference repository and the disambiguation algorithm, presented in chapter 4. SOSEM also exploits the advantages of ontology enrichment presented in chapter 3 for automatically improving the precision of search results.

SOSEM is innovative in several respects: Only tags are used for identifying the semantic content of resources as opposed to textual content. As argued in chapter 2, section 3 this takes advantage of the community vocabulary used to annotate the resources, which may be different from the language used in the resource itself. The approach is not tied to a specific domain or type of tag-based search engine. SOSEM has been evaluated in an increasingly important context in which domain specific resources need to be retrieved from Social Media instead of from a curated domain specific repository. SOSEM uses an innovative combination of ontology mapping and reference repositories using a disambiguation algorithm. These characteristics enable SOSEM to support new data collections automatically, as later sections will show. SOSEM as a whole, incorporates keywords, tags, domain ontologies and large reference repositories. It constitutes a novel approach to ontology-supported semantic search in Social Media.

I will start by giving an overview of the loosely defined research area of semantic search in section 2, followed by a detailed explanation of some of the difficulties encountered during search by users in section 3. Section 4 will give an overview of SOSEM and its components. Section 4 will provide implementation details and background motivations of its various components. The evaluation (section 5) proves that the precision of the search results, using SOSEM, increases significantly compared to the original keyword-based search results, with only a limited impact on recall.

2 State of the Art

Semantic search can be broadly defined as a "document retrieval process that exploits domain knowledge" (Mangold, 2007, p.24). The required domain knowledge is often encoded as explicit semantic features such as specific identifiers for people, organizations and concepts (see chapter 2 section 2 for details). These features are more abstract than the original content of the document and improve retrieving the resource in response to a search query. First, it enables one to find documents that contain different but synonymous terms. For example, 'NL', 'The Netherlands' and 'Holland' all refer to the same semantic entity `dbpedia:Netherlands`. A search request using the semantic entity `dbpedia:Netherlands` can retrieve resources containing any of its lexicalisations. Second, since concepts are not ambiguous, semantic search is not affected by problems related to ambiguity as opposed to keyword-based search.

Semantic search can be performed using a broad class of techniques and it is for this reason that some authors have identified up to eleven different types of semantic search[1]. Semantic search is also frequently conflated with Question Answering (Hirschman and Gaizauskas, 2001) and Knowledge Retrieval (Yao et al., 2007), which adds to the confusion. There is a significant number of survey papers about semantic search specifically (Mangold, 2007; Hildebrand et al., 2007; Wei et al., 2008; Davies et al., 2009) and a conference series (Tran et al., 2011) illustrating the activity in this research area. Many of the approaches to semantic search are domain or corpus-specific such as e-learning (Stojanovic et al., 2001; Monachesi et al., 2008) or airplane maintenance (Bhagdev et al., 2008).

The required domain knowledge for semantic search can be formalized by means of an ontology (Gruber, 1993), but there are also approaches to semantic search that do not rely on ontologies. Egozi et al. (2011) for example includes an excellent historical overview of such approaches to semantic search from a slightly different Information Retrieval (IR) perspective. Not much of the IR-perspective to semantic search will be included, because they usually lack explicit semantic features (concepts). In addition, the lack of concepts that are part of a domain ontology does not allow for more sophisticated representations of domain knowledge, which are preferable in learning scenarios (Westerhout et al., 2011).

The approaches to semantic search that are relevant to discuss in the context of this chapter are the ones that require an ontology and/or data from the Semantic Web. I thus restrict semantic search to those systems that improve document retrieval by means of concept analysis within a(n) (domain) ontology. I will refer to such systems as *ontology-supported semantic search*. I will only discuss ontology-supported semantic search specifically in the context of resource retrieval. Knowledge Retrieval and Question-Answering are outside of the scope of this dissertation. Ontology-supported semantic search aims to maximize precision and recall of search results by means of explicit semantic features from an ontology. Virtually all of the semantic search systems surveyed by Mangold (2007) exploit an ontology in some manner. This also holds true for the semantic search systems discussed in the more limited survey by Davies et al. (2009).

There are two important aspects of ontology-supported semantic search that constitute the core functionality of semantic search: query disambiguation (section 2.1) and query rewriting

[1] http://www.informationweek.com/news/software/bi/222400100 retrieved 2012-08-06

(section 2.2). A proper understanding of these two aspects is vital for grasping how semantic search operates. Section 2.3 will survey other semantic search methods related to the one presented in this chapter.

2.1 Query disambiguation

The first aspect of ontology-supported semantic search is query disambiguation. Its primary function is the increase of precision. This section only surveys disambiguation in the context of document retrieval. For an extensive literature overview of disambiguation algorithms in general see chapter 4, section 2. It is not a trivial task to do query disambiguation for document retrieval in such a way that it actually does not harm performance:

> "To tackle polysemy, the main proposed method was to apply automatic word sense disambiguation algorithms to documents and query. Disambiguation methods use resources such as the Wordnet thesaurus or cooccurrence data (Schütze and Pedersen, 1995) to find the possible senses of a word and map word occurrences to the correct sense. These disambiguated senses are then used in indexing and in query processing, so that only documents that match the correct sense are retrieved. The inaccuracy of automatic disambiguation is the main obstacle in achieving significant improvement using these methods, as incorrect disambiguation is likely to harm performance rather than merely not improve it." from (Egozi et al., 2011)

Schütze and Pedersen (1995) incorporate the use of multiple active senses for a single term as opposed to the common assumption that a term realizes only a single word sense in a context. Word senses in this approach are automatically learned from a corpus using Singular Value Decomposition (SVD). Schütze and Pedersen (1995) argue that word sense disambiguation always needs to rely on a domain corpus or some other source of detailed domain knowledge, because different domains impose different sense distinctions and language conventions. Schütze and Pedersen (1995) illustrate that document retrieval performance assisted by word sense disambiguation can improve retrieval precision by 14% when **combined** with keyword-based search. However, their disambiguation algorithm actually achieved poorer performance in about half of the cases when used in isolation.

Egozi et al. (2011); Gabrilovich and Markovitch (2006) use Wikipedi a to generate explicit semantic features (concept vectors) for documents. These concept vectors are then used for resource retrieval under the assumption that they provide a good abstraction over a document's content. An advantage of this approach is that the model does not need to be retrained for each domain because it is based on Wikipedia as opposed to corpus-specific methods such as in Schütze and Pedersen (1995). However, considerable post-processing is required on the initial Wikipedia-derived concept features in order to achieve satisfactory retrieval performance. Egozi et al. (2011) achieved a performance increase of 18% at best over the bag of words (BOW) method used as a baseline on TREC-8 datasets (Voorhees and Harman, 1999).

Disambiguation is a vital part of semantic search and needs to be well integrated into the overall approach. Egozi et al. (2011) show that state of the art performance can be achieved

with Wikipedia-derived features. I will take a related approach with SOSEM in order to achieve domain-independent semantic search without requiring a domain-specific training corpus.

2.2 Query rewriting

Query rewriting is an important aspect of Semantic Search. Query rewriting involves the automatic or interactive modification of a search request. An example of query rewriting is the functionality offered by many search engines to suggest alternatives to the original query very similar to the one the user entered. These alternatives frequently consist of either spelling corrections, synonyms and common additional terms that make the original query more specific. Not all semantic search systems require query rewriting as it depends on the level of coupling between the semantic representation and the document collection. Query rewriting strategies are always part of a larger semantic search system and can therefore not be evaluated in isolation.

Mangold (2007) distinguishes between tightly coupled and loosely coupled systems. Tightly coupled systems have direct links between the concepts from an ontology and the documents in the corpus whereas loosely coupled systems do not. Loosely coupled systems access documents via a term-based query, constructed using an ontology. Semantic search performed as post processing of an external keyword-based search engine is loosely coupled and reusable.

Mangold (2007) notes that improving the recall of a query using an ontology can be achieved by replacing a term with a hypernym. Three broad classes of query-modification techniques are described; manual query modification, query rewriting and graph based query modification. Manual query modification requires the user to generate a new query in response to output generated by the system. Query rewriting involves the automatic substitution, addition or removal of keywords from a query using an ontology. Graph based query modification requires tight coupling and treats documents and queries as graphs in order to perform graph-based matching of documents with queries.

Navigli and Velardi (2003) evaluate the impact of ontology based query expansion. They show that expanding a query with synonyms or hyperonyms has only a negligible effect. Other types of information from an ontology are much more effective at improving search results. These include "words in the same semantic domain (and same level of generality) of the query words appear as the best candidates for expansion" (Navigli and Velardi, 2003, p.47). The use of nouns extracted from WordNet (Miller, 1995) glosses (definitions and/or examples sentences) outperforms query expansion based on synsets and hypernyms using WordNet. Navigli and Velardi (2003) evaluated a set of 24 queries from the 'TREC2001 Web Track' and determined the accuracy of the first 10 search results from the Google search engine. The use of gloss words for query expansion achieves a 26% improvement for unambiguous words and 23% improvement for disambiguated words compared to the original keyword-based baseline query. Accuracy improvements of query expansion using WordNet's synonyms or hypernyms varied between 1% and 3%.

Query rewriting is only required for loosely coupled systems. It is important because it allows for the retrieval of resources that are not found on the basis of the initial keyword

for a concept. SOSEM can therefore also employ query rewriting as part of its strategy, to automatically retrieve relevant resources when initial search results are poor.

In some cases, query rewriting might be relevant. For example, a search query using 'js' might not yield suitable resources, whereas a search query with 'javascript' will. Although both 'js' and 'javascript' are acceptable lexicalizations for the `lt4el:JavaScript` concept, sometimes only one will yield good results. An ontology that has both lexicalizations for the `lt4el:JavaScript` concept is capable of modifying the query when poor results are retrieved in order to improve the quality of the search results. However, it also needs to contain the poorly performing term as an alternate lexicalization in order to recognize which concept the user intended. In spite of this, the critical review of Navigli and Velardi (2003) suggests that the impact will be negligible. Query rewriting is therefore not evaluated in the context of SOSEM.

2.3 Related systems

This section will survey a number of approaches to ontology-supported semantic search that are related to SOSEM. The systems and approaches are presented in chronological order. This order illustrates a trend that moves from reasoning with high quality RDF data (section 2.3.1) towards exploiting larger noisy data collections using vector space models (see section 2.3.2). Section 2.3.3 describes approaches that exploit large data collections that increasingly originate from Social Media and online collaborative resources such as Wikipedia.

2.3.1 RDF exploitation

The first system covered in the survey from Mangold (2007) with functional overlap with SOSEM is SHOE (Hein and Hendler, 2000). SHOE requires web pages to contain semantic metadata as part of their content. This was a controversial approach in 2000, but it is an accepted practice in 2012, e.g. it is currently part of the evolving HTML5 standard[2] [3]. SHOE at the time did not exploit RDF yet, but RDF was explicitly referenced by the authors in their paper as a compatible technology. Hein and Hendler (2000) experimented with a search interface where users constructed queries by constructing graphs. This interface was quickly abandoned, because they determined that such an interface was only usable by experts and not by regular users. An updated version of the user interface allowed users to construct a graph by selecting one or more terms from a hierarchy. However, this approach was reported to be restricted to a single domain at a time. A major reported hurdle for the SHOE system was not having a method of convincing users of the added value of annotation, i.e. manually adding semantic metadata to online resources using concepts from an ontology. The discussion of Social Media, carried out in chapter 2, has illustrated why tagging has largely solved this resource annotation problem.

A more recent approach to semantic search uses Semantic Web data retrieved from the Web. The system from Guha et al. (2003) exploits RDF data to find concepts or instances. Guha

[2] http://www.w3.org/TR/2012/WD-microdata-20120329/
[3] http://www.w3.org/TR/2012/WD-rdfa-in-html-20120329/

et al. (2003) propose a set of APIs to access semantic content. An ontology is dynamically constructed in response to a user's search query using RDF data from the Internet that is crawled on-the-fly. HTML scrapers were developed to extract semantic information from websites that did not provide RDF data themselves. Terms in a search query are linked to one or more concepts in the ontology graph. Query disambiguation is performed by minimizing the overall distance between the search query's concepts in the ontology graph. User context in the form of a user's previous queries and concepts is also employed to decide on the most likely word sense given multiple candidates. The additional data from the Semantic Web is converted into human-readable form using templates and presented alongside with traditional keyword-based search results.

Tran et al. (2007) propose "an approach for translating keyword queries to description logic conjunctive queries using background knowledge available in ontologies". The ontology is used to answer the question directly or provide concepts that reflect the information need of the user. This is accomplished by generating a subgraph from the ontology that links the various keywords of a query. The evaluation, based on 42 queries, shows a precision of 0.69 and recall of 0.42. The authors also state that "queries which were obviously out of the scope of the knowledge base were removed" (Tran et al., 2007, p.534). "One major problem our approach suffers from is the fact that it does not consider that keywords can be ambiguous with respect to labels in the ontology" (Tran et al., 2007, p.536).

Wang et al. (2008) propose translating keyword-based search queries to ontological concepts by looking up the appropriate literals in an ontology. The lexicalizations of the concepts are enriched with additional terms extracted from Wikipedia to increase the likelihood of a user's query matching a concept. Their approach is also capable of generating a fragment of an ontology that summarizes the information need originally expressed as a keyword-based query. This visualization allows users to explore neighboring concepts and illustrates how the concepts from the keyword-based query interrelate. Wang et al. (2008) explore three ranking methods: path-based, relevance-based and importance-based. The importance-based ranking method assigns an optimized cost to edges and nodes in the ontology and performs best. The average Target Query Position (difference between gold-standard ranking and actual ranking with 10=max and 0=min) obtained with this method is 9.125.

2.3.2 Vector space models

Bonino et al. (2004) present an approach based on vector space models. Vector space models originate from Information Retrieval research. Vector space models consist of multi-dimensional spaces generated from a corpus. Terms and documents are represented as vectors in a multi dimensional model. The cosine angle between terms or documents acts as a measure of similarity. Bonino et al. (2004) propose to use concepts instead of terms to create a vector space model using a corpus annotated with concepts. A manually created ontology is used to automatically annotate a collection of documents. Keyword-based queries are matched to concepts from the ontology and documents are retrieved that are highly similar to the concept vectors. The ontology is exploited to automatically improve a query by adding relevant concepts. Query refinement is applied by including broader and narrower concepts in the query. The authors argue that this increases performance, because documents can match conceptually with a larger amount of concepts. Bonino et al. (2004) evaluated their approach

using two queries. The contribution of query revision is reported to increase the precision from 0.50 to 0.67 for the first query and only marginally for the second query[4]. An important reason for the poor performance is that recall is at about 20% or less for the two example queries with precision between about 60% and 70%. The authors note that results "... are not yet competitive from the point of view of precision and recall since the Basic Search module is not optimized, however they show that an ontology based query expansion process is able to provide improvements" (Bonino et al., 2004).

Castells et al. (2007) explore a similar route as Bonino et al. (2004), i.e. they propose a hybrid of RDF and vector space models in order to realize "an adaptation of the vector-space model for ontology-based information retrieval" (Castells et al., 2007, p.261). However, in contrast to Bonino et al. (2004) they presuppose that queries are expressed in a formal RDF query language as opposed to keywords. They thus skip the step that transforms a keyword-based query to a semantic query using concepts. Traditionally, ontology reasoning is used to improve resource retrieval, but this works poorly if the ontology is of sufficient coverage. The aim of Castells et al. (2007) is to improve resource retrieval using other means than ontology reasoning and to support resource retrieval with incomplete ontologies. Their evaluation using 20 queries on a CNN-news corpus shows a ~0.1 improvement in precision when using ontology-based search compared to keyword-based search. The authors also evaluated a combined ranking score. This combined score takes the average of term- and ontology-based scores and handles some special corner cases. This combined score results in a significant improvement of 0.2 to the precision compared to term-based search. The combined method was overall more effective, because term-based similarity compensates for situations where a query is not sufficiently covered by concepts from an ontology.

2.3.3 Searching in Social data

Kato et al. (2008) show that exploiting the coocurrence patterns in folksonomies can significantly improve recall and precision of search results. Their approach aims to improve the precision of image search results. Kato et al. (2008) observe that precision and recall using keyword-based search for images increases significantly when concrete keywords are used as opposed to abstract ones. For example, the keyword 'spring' returns poor results whereas 'crocus', 'tulip garden', or 'blossom' yield more relevant results. WordNet, a large dataset of FlickR tags and clustering are used to transition from an abstract term to a set of concrete terms. Precision at a recall of 0.5 increases from near zero using an abstract term to about 0.4 using the automatically generated set of concrete terms.

Santamaría et al. (2010) investigate whether the diversity of simple web search results can be improved using semantic search exploiting Wikipedia and Wordnet. Their approach aims to support users performing exploratory searches in Social Media using semantic search and ontologies. They report that Wikipedia has a much better coverage of word senses in search results in comparison to Wordnet (56% vs 32%). Disambiguation is performed using TiMBL (Daelemans et al., 2007) on both the Wikipedia article text for the word senses and text surrounding the wikilinks from other articles. The distribution of senses was estimated using the

[4]The paper contained recall vs precision plots. The second plot seemed to lack information regarding performance of the system without query refinement beyond a recall of 0.08

wikilinks and the relative number of visits received by each sense (article) in Wikipedia. Using this information to diversify the search results for simple queries allows them to achieve a coverage of 77% of the predominant word senses compared to an original baseline of 49% when applied to the first 10 results from Google Search.

dos Reis et al. (2011) combine both ontologies and social networks in order to improve search results. Software agents embody a type of information need of a user and determine the semantic interpretation of search queries. For example, an agent about tourism will disambiguate the term 'java' differently than an agent about computer technology. They thus establish a link between the language used by users and how it relates to different Communities of Practice on the Web in a social environment. Very few details of the technical implementation are provided.

The SOSEM semantic search approach has in common with dos Reis et al. (2011) that both aim to improve resource retrieval in Social Media using ontology-supported semantic search. However, in my approach, I rely on the presence of tags instead of the document content. The added value of exploiting tags is that non-textual resources such as images or videos can also be retrieved using semantic search. The approaches from Egozi et al. (2011); Bonino et al. (2004); Castells et al. (2007); Tran et al. (2007) also operate only on the textual content of documents and do not presuppose a Social Media setting with tags. Both my approach and that of Egozi et al. (2011) share the assumption that Wikipedia-derived concepts are effective for conceptual analysis. Similarly, Wang et al. (2008) recognize the value of the lexical information of Wikipedia to bridge the gap from keywords to ontological concepts. My approach to semantic search supported by a domain ontology achieves a goal which is the exact opposite of Santamaría et al. (2010) who also used Wikipedia: more specific and unambiguous search results instead of more diverse ones. Finally, ambiguity is explicitly supported in contrast to Tran et al. (2007).

3 Semantic Search

Keyword-based search has problems related to users submitting poor queries, vocabulary mismatches between users and communities, and ambiguity. These issues are especially relevant in the context of e-learning as previously discussed in chapter 2, section 4.2. This section provides details about keyword-based search in the context of Social Media and how semantic search aims to address them.

Issues with keyword-based search Searching for information is very different from consuming activity streams and viewing recommendations, because this process is triggered by the user via a search query. Keyword based search assumes that the user is able to effectively express his or her information need (Taylor, 1962) to an information system, but this assumption does not always hold. Consider, for example, a user looking for information about a specific species of dinosaur without exactly knowing what it is called or a specific piece of scientific literature. Activity streams and recommendations are based on a user's prior actions, whereas keyword based search assumes that a user is able to think of good keywords for an entirely new area of interest.

For a search query to be successful the sought-after resources need to match the search query in some way. For keyword-based search engines this means having lexical overlap between the search query and the resources. Appropriate resources are considerably more difficult to retrieve when the user does not know the correct terms for expressing his or her information need. They can also not be retrieved when the terms are technically correct, but differ from those used in documents. For example, Danescu-Niculescu-Mizil et al. (2013) showed that proper adoption of the community vocabulary is a reliable indicator of user participation in a community. Danescu-Niculescu-Mizil et al. (2013) show that it is more reliable than activity-based measures of participation. In the context of the application domain used in this dissertation, users may not yet be aware of the proper terminology to compose good search queries. To complicate matters even further; the search query may be sensitive to spelling errors, ambiguity and community-specific preferences.

For example, a search for 'websites' (plural) will, counterintuitively, yield different results than for 'website' (singular) using some keyword-based search engines (for an extended example see chapter 2, section 4.3). Another example would be searching for videos concerning JavaScript using the keyword 'javascript'. Videos which have only been tagged with the shorter, but also common abbreviation 'js' will be left out. A search query consisting of a formally correct keyword that nobody uses is ineffective. This situation is similar to the early stages of composing a question for an information system as described in Taylor (1962). It is thus important to align the vocabulary of a search request, composed by novice users, with the actual vocabulary used by the community.

A second problem for keyword-based search is ambiguity. Ambiguity has a significant impact on the quality of keyword-based search results if the query involves an ambiguous term such as 'python', 'ajax' or 'java' (for details see chapter 4, section 2). Alternatively, we could say that the search results reflect the community's predominant understanding of a term. Viewed from this perspective, multiple communities share a single platform for sharing and or creating resources. Some terms are unintentionally shared between different communities. A search query using a shared term will result in resources from different communities. The size and activity of each community will determine the popularity of its particular interpretation of a term and thereby the search results (Santamaría et al., 2010). Terms within a community have a significantly lower amount of ambiguity, because communities enforce conversational boundaries.

For example, imagine an expert on snakes searching for 'python' on `delicious.com`. This user will largely encounter resources about a computer programming language, because `delicious.com` is heavily biased toward computer-related subjects. Similarly, a computer scientist searching for 'ajax' on the video sharing site `youtube.com` will largely get soccer-related videos. These are both examples of one community's interpretation of a term being more dominant than another one's. The search results in these examples are not incorrect in any objective sense, but may be considered irrelevant for users that have another domain of interest.

Users might not yet be experienced enough to formulate more complicated queries for informational search (for details see chapter 2, section 4.3) due to a lack of domain knowledge. An analysis of the search logs of an internal intranet of British Telecom from January until May 2004 revealed that queries contained 1.8 terms on average (Davies et al., 2009). Similarly, the

average query length of the search results from the AOL search engine March-May 2006 was 2.34 (Shi, 2007). Both sources show that short search queries are actually quite common and that one can not blindly rely on users' ability to compose long sophisticated search queries for locating information. Not all users can rely on additional lexical competence (Marconi, 1995) to improve their search queries.

This concludes the discussion of problems related to keyword-based search and I will now turn to potential solutions to these problems.

Addressing some of the problems related to keyword-based search It is possible to improve keyword-based search with stemming in order to address issues where the plural and singular forms of a term result in different types of resources. It is less clear how to automatically identify whether 'js' and 'javascript' should yield similar types of resources. Personalization can be applied to keyword-base search to give priority to resources with considerable lexical overlap with previously accessed resources. However, this technique cannot be applied when a user searches for information regarding a new domain of interest. Similarly, keyword-based search cannot address issues with ambiguous terms when no other contextual information is available. Various types of techniques are known to improve the performance of generic keyword-based search, but there are limits as to what problems of keyword-based search they can address. This chapter will exclusively focus on semantic search as one of the alternatives to generic keyword-based search.

Semantic search has the potential to provide significant advantages over keyword-based search, but requires that all resources are annotated with concepts instead of terms. This is not the case in Social Media, because resources are annotated with tags, i.e. short terms used for annotation. The integrated whole of users, resources and tags in Social Media constitutes a *folksonomy*; a lightweight conceptualization of a domain as a counterpart to the formal ontology (see chapter 2 section 3). Semantic search, however, relies on a formal ontology as opposed to a folksonomy. The transition from an arbitrary Social Media folksonomy to a formal ontology enables semantic search to exploit the structure and quality of an ontology with the amount of data created and shared in Social Media. SOSEM is able to make the transition from the ontology to the folksonomy and vice versa using a combination of ontology enrichment from chapter 3 and the disambiguation approach from chapter 4.

In order to apply semantic search to resources coming from Social Media we need to address issues with keyword-based search in Social Media tags such as the ambiguity of tags. Due to the fact that tags have a relatively high degree of ambiguity (for details see chapter 4, section 2) search results with low precision in response to short ambiguous queries can be frequent.

The SOSEM approach to semantic search presented in this chapter takes as one of its components an independently existing keyword-based search engine that works on Social Media using the tags associated to resources (a tag-based Social Media search engine). It generates keyword-based queries for this tag-based search engine and it operates on its output. It can thus be regarded as a meta-search approach. SOSEM will treat the source (search engine) of the search results as a black box, as we cannot say anything in general about the algorithm that generates those results. However, the fact that tag-based social search has certain issues does not mean that its results can be disregarded entirely. The original order of the

search results reflects important features about them such as relevance, popularity or quality. The SOSEM approach therefore retains the original order of the results and improves them through post-processing using semantic knowledge. This process will remove search results which are not relevant for a specific search query as determined by a domain ontology and thereby increase precision.

In section 4, I will present the SOSEM approach for improving the quality of search results from Social Media, such as YouTube. It is based on previous work from the LTfLL-project (Monachesi and Markus, 2010b,a; Markus and Westerhout, 2011). It makes use of ontologies that are automatically enriched with tags extracted from Social Media resources (see chapter 3). An (enriched) domain ontology is used to apply semantic search to Social Media results that have not been previously annotated with concepts. The tags from these search results will be automatically linked to the proper concepts using my disambiguation method presented in chapter 4. The SOSEM approach aims to improve the precision of search results. The evaluation, described in section 5, shows that it allows for a significant improvement of search in tag-based Social Media even when one uses ambiguous and/or poor keywords.

4 SOSEM design

Keyword based search in Social Media suffers from problems with misaligned vocabularies between communities and users, but also ambiguity resulting in poor precision. These problems can be addressed using semantic search. In order to apply semantic search to Social Media, tags need to be linked to concepts and vocabulary mismatches between users and communities need to be resolved. The SOSEM approach to semantic search will employ enriched domain ontologies (for details see chapter 3) and domain-independent disambiguation (for details see chapter 4) in order to realize semantic search. Both regular domain ontologies and enriched ontologies can be used, but the process of ontology enrichment (see chapter 3) is not part of SOSEM itself. SOSEM makes use of existing keyword-based search engines for Social Media and improves the precision of these search engines using semantic analysis of the associated tags.

The central idea of ontology-supported semantic search in general is that concepts help differentiate between relevant and non-relevant results in a way that is not possible with terms alone. This advantage of semantic search also applies to semantic search systems in a loosely coupled environment (Mangold, 2007) such as Social Media. A users' information need in SOSEM is expressed using one or more ontological concepts, search results are analyzed and only those results that include the right concept are retained. The SOSEM approach to semantic search thus discards search results that do not match the *concepts* from the search query. The precision of the search results using this method for ambiguous or poor keyword-based queries improves considerably. SOSEM assumes that the resources have been annotated with a number of tags, and uses only these tags and not any textual content of the resource.

The algorithm executes six steps in order to achieve its goal:

1. Determine what domain ontology concepts are contained in the user's search query. (section 4.1)
2. Perform keyword-based search requests on tag-based Social Media using the concepts' lexicalisations extracted from the ontology in the generated search query. (section 4.2)
3. Disambiguate the terms/tags from each search result. (section 4.3)
4. Perform ontology mapping from the selected domain ontology to a reference repository. (section 4.4)
5. Filter the search results using the concepts obtained via disambiguation of the search results and original search request. (section 4.5)
6. Perform query rewriting if the results are poor. (section 4.6)

The overall process is also displayed as a diagram shown in figure 5.1. Each of the six steps of SOSEM is explicitly referred to using each step's number.

The subsequent sections will describe the various steps in more detail and elaborate on the various trade offs and design considerations. I will conclude with a simplified example of the complete SOSEM approach in section 4.7.

4.1 Search query concepts

The input for the SOSEM semantic search process is a search query. The output of the first step is a set of domain ontology concepts. The SOSEM approach can only function if a domain ontology is available and can be used by the system. The search query is either expressed as a list of keywords provided by a user or a concept from a domain ontology selected by a user. If the input query is a list of keywords then the goal of this initial step is to transform the input search query into a set of concepts from the domain ontology.

An implementation of the SOSEM semantic search approach was integrated into an e-learning support system called iFLSS and was developed as part of the LTfLL-project (Berlanga et al., 2009). The web-interface for the iFLSS is shown in fig. 5.2. Users employ the iFLSS to perform semantic search on resources coming from various Social Media, including YouTube. Users have two means of expressing their information need; (1) graphically by clicking on the concept in the domain ontology that is visualized as a graph, (2) by entering a number of keywords in a search box. Selecting a concept by clicking on it immediately provides information on its lexicalization, definition and relevant search results retrieved using the SOSEM approach. The visualized ontology makes it possible for a user of the iFLSS to select the concept directly without having to resort to a keyword-based search query at all.

If the query takes the form of a list of keywords, the role of the domain ontology is to constrain all possible interpretations of the keywords. For example, a domain ontology about Greek

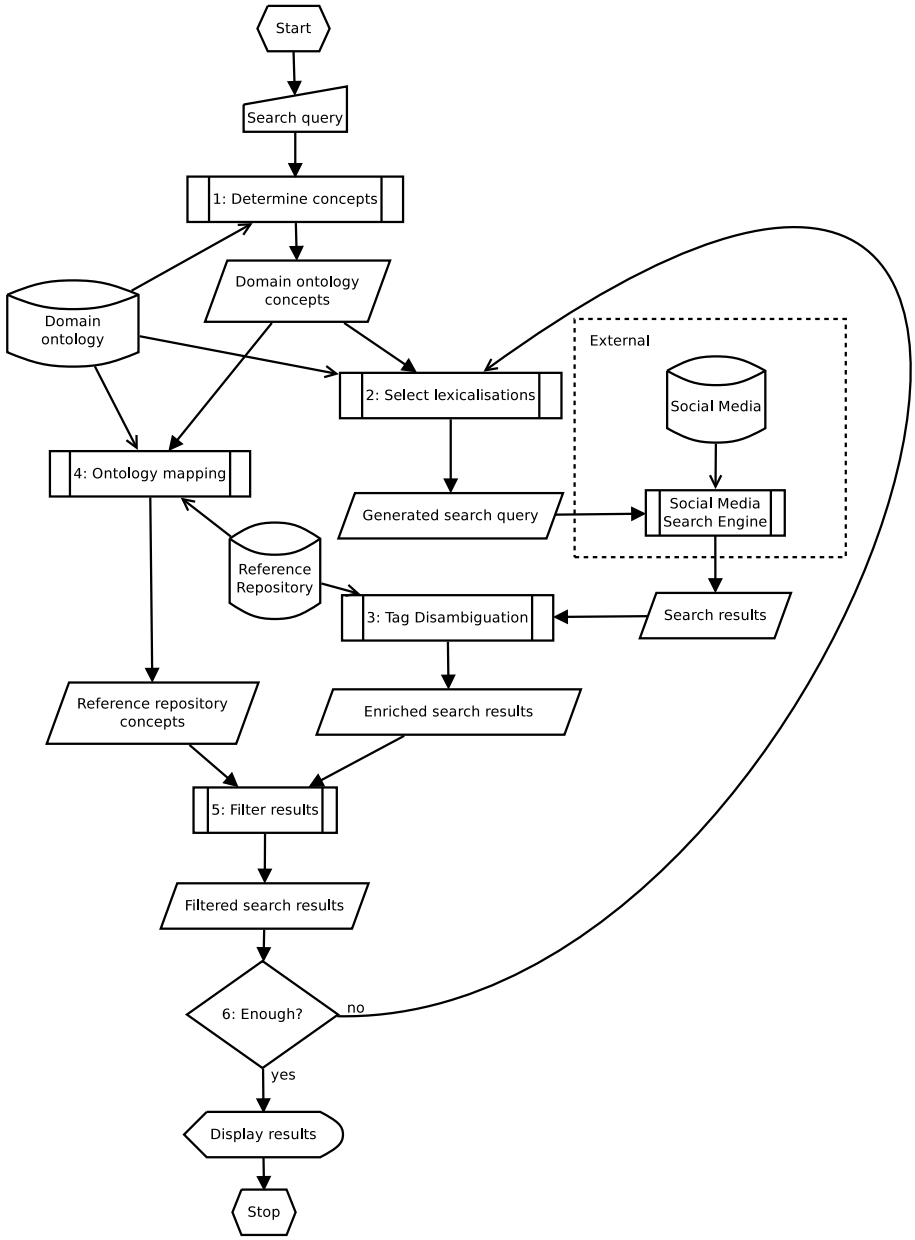


Figure 5.1: Schematic diagram of the SOSEM semantic search process for Social Media.

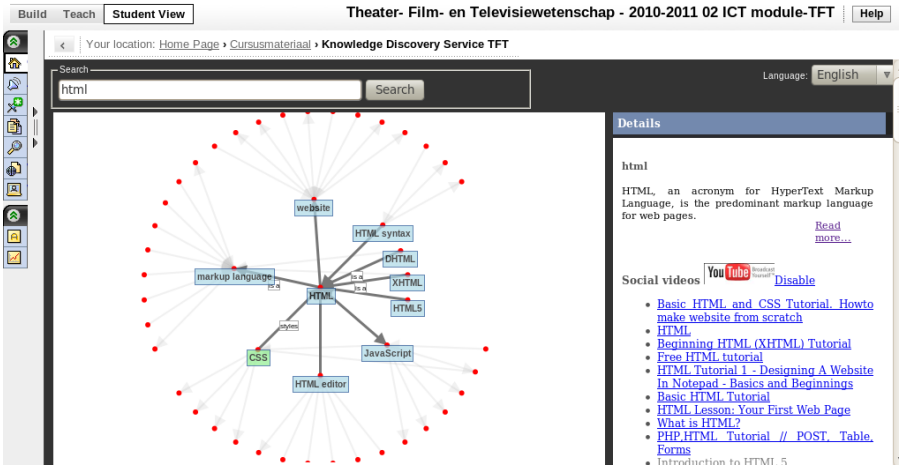


Figure 5.2: The Knowledge Discovery component of the informal Learning Support System: (1) "Search box, (2) "Ontology browsing, graph visualisation of a domain model, (3) "Definition of the centralized concept, (4) "Learning Materials, dynamically retrieved from social networks. Image and adapted caption from (Westerhout et al., 2011)

mythology will not provide concepts like `dbpedia:Football_team` for the term 'ajax'. We will say that a domain ontology provides a highly *biased interpretation* of a term towards a domain concept. The goal of this step is thus to establish the set of domain ontology concepts that the search query represents. Search queries are treated as conjunctive queries without support for disjunction, negation or brackets. The appropriate domain ontology concept for each of the (multi-word) terms of the search query is identified using the ontology's lexicon.

An important effect of limiting the query interpretation to the domain ontology's concepts is that ambiguity is significantly reduced, because it is restricted to a specific domain. In contrast, subsequent steps in the SOSEM approach will show that disambiguation across multiple domains is essential for semantic search. Additionally, an ambiguous search request consisting of a singular term can only be disambiguated using a domain ontology since no other sources of information are available. It is frequently the case that a term in the context of a domain ontology can only refer to a single concept. However, if a term is ambiguous in the domain ontology's lexicon then multiple domain ontology concepts are selected instead of one for an unambiguous term.

In summary, the search query is given a *biased domain ontology interpretation* in terms of domain ontology concepts. The analysis of a keyword-based query may include identifying multi-word terms, stemming and optionally additional NLP-analysis for morphologically rich languages. The concepts from the search query constitute an unambiguous semantic search query to be used in subsequent steps.

4.2 Search request generation

The input of the second step is a set of domain ontology concepts. The output is a keyword-based query. The domain ontology is used to generate a keyword-based query using the domain ontology concepts. Keywords are selected using the lexical information that is available for each concept in the (enriched) domain ontology. The goal is to acquire relevant resources from a tag-based Social Media search engine using keyword-based search.

The 'biased domain ontology interpretation' obtained in the previous step is used to compose a search query on tag-based Social Media, because external tag-based Social Media search engines can only be queried using terms. This step involves interaction between the conceptual level and the term-level (lexicon) of concepts. Recall that a concept can have multiple terms (lexicalisations) attached to it (see chapter 2, section 2.1.1 for details). SOSEM uses the terms associated with a concept for building search queries for tag-based Social Media and is thus loosely coupled (Mangold, 2007).

The preferred term of each concept is used to create a keyword-based search query. The order of the terms is irrelevant. Only one term is selected for every concept identified in the original keyword-based query. Note that this use of terms re-introduces issues with ambiguity and divergent vocabularies for the same concepts. These issues will be addressed in subsequent steps of this approach. The terms used to compose the search query are simple terms and provide no support for regular expressions or boolean operators. The term-based search query, generated using the domain ontology's concepts, is then sent to the tag-based social search engine which is assumed to be a 'black box'. This black box yields an ordered number of search results that each have a title, one or more tags, authorship information and links to one or more resources. These search results could all be relevant or irrelevant for a specific domain, but all we know at this stage is that they match the generated term-based search query. More specifically, they contain the same terms (tags) as specified by the generated term-based search query.

Query rewriting is applied when search results are largely not relevant or when not enough resources are returned. This involves rewriting the query (see section 4.6 for details) using other preferred and alternative terms available in the domain ontology.

4.3 Tag disambiguation

The input for the third step consists of the search results obtained in the previous step. The output consists of the same search results enriched with concepts. The goal is to link the (ambiguous) tags from each search result to the appropriate concepts using a disambiguation algorithm (presented in chapter 4).

Each search result has been 'tagged', i.e. each result has a list of terms associated with it that come from the community vocabulary. The external Social Media search engine that generated the search results often includes only the most frequent tags for each resource. Although all the search results contain the term, e.g. 'ajax', this does not mean that the conceptual content matches as well. Tag disambiguation is essential for identifying the appropriate concept for each tag. These concepts will be used in step 5 to filter results that do not match on the conceptual level. Only search results that match on the conceptual level are retained.

In contrast to the term disambiguation in the context of the domain ontology, each of the search results should be given a *neutral* term-concept assignment, i.e. the terms in the search results are linked to concepts without any obvious domain bias. This is different from the search query itself which intentionally gets a domain bias via the domain ontology. A reference repository is used to disambiguate, because it provides a neutral view of the potential meaning of a term, i.e. it treats all domains as equally likely.

The neutral tag-concept assignments are obtained by disambiguation using one or more reference repositories. More specifically: for every resource, the tags included for each resource are considered. Such a set of tags is called a *Tagging-instance* in the SCOT vocabulary (Kim et al., 2008a). The disambiguation algorithm associates the proper concept (sense) for a tag given the other tags as context, as previously described in chapter 4. Each tag of a search result is linked to a concept from a reference repository.

4.4 Ontology mapping

The input for the fourth step is the set of domain ontology concepts obtained in step 1. The output is a set of corresponding reference ontology concepts (for details see chapter 2, section 2.3). The goal is to convert the set of domain ontology concepts, obtained from the input keywords, into a set of equivalent reference concepts using the DBpedia reference repository and the (enriched) domain ontology using the disambiguation algorithm (for details see chapter 4).

At this stage, the semantic search algorithm has two sets of concepts that are available; (1) domain ontology concepts derived from the user search query and (2) reference concepts linked to the tags in the Social Media search results obtained in the previous step. Using ontology mapping the domain ontology concepts will be linked to corresponding reference concepts. This requires an ontology mapping operation, as previously presented in chapter 3, section 4.2, because the disambiguated search results only contain concepts from the reference repository and are thus different from the set of domain ontology concepts derived from the keyword-based search query. Ontology mapping is required in order to be able to compare reference repository concepts to domain ontology concepts.

After ontology mapping, two comparable sets of concepts are available from the same sense repository. Search results from Social Media can now be filtered by checking whether the concepts associated with the tags are equal (have the same URI) to those specified by the domain ontology's biased interpretation of the search query (see section 4.5).

4.5 Search result filtering

The input for the fifth step consists of the enriched search results from step 3. The output consists of a subset of these enriched search results. The goal is to filter the search results using the mapped search query concepts obtained in step 4. This is accomplished by comparing the mapped search query concepts with the concepts from enriched search results.

All the search results have in common that they have lexical overlap with the keyword-based search query. However, not all search results may be relevant if the terms of a search query are

ambiguous. The disambiguation algorithm in step 3 has identified the appropriate concepts for the tags of each search result. The effect is that each enriched search result has a number of reference concepts attached to it that disambiguate the tags. Ontology mapping in step 4 has identified the appropriate reference concept counterparts of the domain ontology concepts of the search query. Comparing the various concepts allows for removing search results that do not match on the conceptual level, thereby improve precision when compared to pure keyword-based retrieval.

For example, a resource on Greek mythology can be tagged with `ajax,greece,iliad` given a term-based search query for 'ajax'. The terms 'greece' and 'iliad' provide context which allows the disambiguation algorithm to properly disambiguate 'ajax'. The terms 'greece' and 'iliad' are however not used for deciding whether the search result is relevant or not. Only a subset of the tags of each search result is relevant to this end. It only considers the reference concepts associated to tags/terms shared by both the search result and the original search query.

More formally the previous ontology mapping step has resulted in a set of reference repository concepts for the search query Q and a set of reference concepts for each tagged search result, R. Each search result is removed if $Q \not\subseteq R$, i.e. the **concepts** of the search query all need to be contained in the set of concepts of a search result. The search results that remain thus cover the concepts present in the search request.

Terms that are not ambiguous at all, i.e. terms yielding only one reference concept, do not reduce recall. The ontology mapping step always maps the domain ontology concept to a single reference repository concept in this situation. The same applies to the tag disambiguation for search results which is performed in step 4. Search results which contain an unambiguous tag are guaranteed to be retained, because the linked concept always matches the reference concept obtained through ontology mapping. For example, assume that a user enters the keyword-based query 'html'. The LT4eL domain ontology on computing yields the concept `lt4el:HTML` which is linked, using ontology mapping, to the `dbpedia:HTML` reference repository concept. The tags of the search results, obtained using the domain ontology's preferred term 'html', are all disambiguated as `dbpedia:HTML`. All search results for 'html' are retained during filtering, because the two reference concepts match.

4.6 Search query rewriting

The input for the sixth step are a keyword-based search query generated by a domain ontology and the amount of results retained after filtering in step 5. The output is a modified search query expressing the same information need using alternative terms. The goal is to generate a modified search query using alternative terms from the domain ontology's lexicon in order to improve the query and thereby the precision of the search results. This step is only triggered if the number of retained search results is too small.

Having few relevant search results is an indication that the generated search query is poor relative to the relevant Social Media. In such cases an alternative lexicalization from the domain ontology's lexicon should be used to improve the generated search query. This alternative lexicalization may better represent the dominant community vocabulary and thus generate

more relevant search results. It was previously implicitly assumed that the preferred term for a concept would lead to sufficient search results, i.e. a list of search results that contains a sufficient number of resources with the correct concept. It occasionally happens that the preferred term for a concept in a domain ontology actually leads to poor or no search results at all. For such cases, the preferred term of the concept is not part of the community vocabulary for that domain. For example, if the term 'Asynchronous JavaScript and XML' is used instead of the more popular term 'ajax' very few results will be obtained. A term can also be identified as a poor lexicalization of a concept using the previously described semantic search process. For example, if a term does generate a lot of search results, but very few contain the correct concept, this is also a good indicator that the chosen term is poor and/or insufficient.

Query rewriting is triggered if the number of search results retained after filtering does not pass some threshold ratio, for example when more than half (>0.5) of the search results is determined to be irrelevant. The simplest method of improving the search query is by trying an alternative term for the same concept. These alternative lexicalizations (stored using `SKOS:prefLabel` or `SKOS:altLabel` properties in the domain ontology) are perhaps more acceptable in the online community and thus increase the relevance and/or number of search results. The search query still covers the exact same concepts, but uses different terms for these concepts. The modified search request is subsequently sent to the external Social Media search engine in order to acquire other resources. Results are again filtered as presented in previous sections.

When alternative terms for the concepts are either unavailable or ineffective, a query expansion step is triggered. In such cases, it is assumed that the search query is too generic and requires additional terms to narrow down appropriate resources, e.g. the concept `dbpedia:Web_Ontology_Language` only has one term; 'OWL'. This term alone may result in many resources but many of them will be not relevant to the intended query (low precision). In these situations, the system can extend the query by adding a term from a super concept, or when not available, a related concept. The introduction of additional concepts will add terms to the query which makes it more specific. This increases the likelihood of obtaining relevant results (high precision).

A subsumption relation is preferred for retrieving a related concept, because this prevents creating a query that is too specific. For example, consider a query for `dbpedia:Utrecht_(province)` that yields poor results. Adding the super concept `dbpedia:Provinces_of_the_Netherlands` instead of a related concept such as `dbpedia:Utrecht_Hill_Ridge` is preferred. The related concept `dbpedia:Utrecht_Hill_Ridge` will exclude resources about `dbpedia:Utrecht_(province)` in general, because it makes the query more specific than intended.

The domain ontology is used to decide which term to add or change, which is a simplified variant of the method suggested in Pan et al. (2009) and is part of the categorization scheme proposed by Mangold (2007). Pan et al. (2009) propose to select the ontology lexicalization with the highest TF-IDF score as the best keyword. However, in SOSEM no corpus is available for the calculation of TF-IDF scores. Additionally, Pan et al. (2009) add terms related to super-concepts to the search query in order to improve results. SOSEM can add additional keywords using the lexicalizations of super-concepts to make a query more specific.

Adapting the query based on the number of relevant results, as determined via the previously described filtering process allows the system to gather enough relevant resources without requiring additional user interaction.

This concludes the description of the SOSEM approach. The approach assists users with retrieving information from tag-based Social Media. It is able to identify whether search results are relevant in the context of the domain ontology even when ambiguous or different terms are used to describe it. The quality of search queries is automatically assessed via filtering and disambiguation, improved queries can be automatically generated using lexical information from an ontology and search queries can be automatically enhanced via ontological relations.

4.7 Example

I will illustrate the application of SOSEM using an example. The overall process is also depicted graphically in figure 5.1. SOSEM starts with the keyword-based search query 'xsl' and a domain ontology about computing technology. The domain ontology will first provide a biased interpretation of this term via the concept `lt4el:XSL`. The second step is to send a search query to a tag-based Social Media search engine using the ontology's preferred lexicalisation for the concept. The search engine returns the following resource titles and tags (terms):

1. "XSL tutorial HD"
   `xsl, soccer, css, football`
2. "XSL programming commands"
   `xml, xsl, xsd, metadata`
3. "XSL attack"
   `xsl, aes, crypto, mathematics`

In the third step the tags associated with each search result are disambiguated. As a consequence the different reference repository concepts are identified for the tag 'xsl' in each search result:

1. "XSL tutorial HD"
   `xsl` → `dbpedia:Xtreme_Soccer_League`
2. "XSL programming commands"
   `xsl` → `dbpedia:XSL`
3. "XSL attack"
   `xsl` → `dbpedia:XSL_attack`

The fourth step maps the domain ontology's concept `lt4el:XSL` to the most likely reference repository concept `dbpedia:XSL` using ontology mapping. This involves extracting the preferred term for each of the directly connected concepts in the domain ontology and using the

resulting set of terms as a disambiguation context. We thereby determine that the word sense for the user's search query 'xsl' is the reference concept `dbpedia:XSL`.

The fifth step compares the reference concept for the search query 'xsl' for each of the search results with the search query concept. Only result number (2) is retained, because it has the same URI as reference concept from the search query.

In the sixth step, additional query revision can be performed when very few results are retained after filtering. This either means trying alternative lexicalisations for the `lt4el:XSL` domain ontology concept or the addition of lexicalisations from more generic concepts. For example, the term 'xml' may be added to the search query, because `lt4el:XML` is a superconcept of `lt4el:XSL` in the domain ontology.

Figure 5.3 shows the filtered and unfiltered results for the keyword-based query 'XSL' for YouTube. The search results provided by YouTube have been filtered using a domain ontology on computer technology. Results that have been crossed out have not been accepted by SOSEM. Both striked and unstriked search results will be included in a regular keyword-based search, because these do match on the lexical level. However, only unstriked results will remain after filtering, because the SOSEM approach associates different concepts with each search results and only a subset matches the concept specified in the search query. The relative amount of relevant resources (precision) is thus higher than before filtering.

XSL Programming Commands part 1
~~Aldenhoven 2.05.2009 XSL 26,6PS~~
Transforming XML,XSL to HTML using saxon9
~~XSL Promotion **HD**~~
Creating a catalogue in InDesign with ONIX, XSL and XML
~~Big Bore 86ccm / Team 2 / SR 50 XSL~~
CSS vs. XSL
~~Screen-used Medical Tricorder TR990-XSL~~
~~XSL Tutorial **HD**.wmv~~
~~Halo 3: Unstoppable! MzzD~~
~~XSL Server Video Read Teh Description~~
~~Gears Of War2 CLAN MATCH MoB vs xSl 09x~~
~~XSL and SSL Promotional Video~~
~~XSL Championship Week 1~~
~~Drunkin' acoustics w/ XSL Adam Sandler - Mr Bakeo~~
XSL Programming Commands part 2
~~XSL Promo Video~~
~~KenX and Commenters: ASL vs. XSL or SL?~~
XSL-FO exempel
~~Drew Ducker's (Ducks) bicycle kick Detroit vs Milwaukee XSL 09~~
ONIX XML to ePub XMP for Adobe InDesign using XSL.
XSL Output Instructions
Visual Studio 2008: Render XSL File
XML & XSL.avi
XSL Include and Import

Figure 5.3: List of search results for the query 'xsl' from YouTube. Striked search results have not been accepted by the filter. The term 'xsl' in the domain of computing usually refers to a set of technologies for specifying XML document transformation and presentation. However, according to Wikipedia 'xsl' can also refer to the 'Xtreme Soccer League' or an encryption breaking method. All unstriked results are correctly identified to be about the domain concept.

5 Evaluation

The effectiveness of an implementation of SOSEM applied to tag-based Social Media search has been evaluated. First, its performance has been compared to regular keyword-based search. Evaluation results show a significant increase in precision of the search results compared to keyword-based search. Second, SOSEM performance using an enriched and unenriched domain ontology (for details, see chapter 3) has been compared. Results show that an enriched ontology significantly increases precision compared to an unenriched domain ontology. I will now turn to a discussion of the evaluation setup and its results.

Setup A concrete implementation of keyword-based search is required in order to determine the improvement of SOSEM's semantic search over keyword-based search. There are several choices with respect to tag-based Social Media search engines that one can make. However, social video sharing sites are of particular interest for SOSEM's evaluation due to several reasons. First, they are highly popular and thus representative of what regular users will encounter. Second, videos are annotated using tags and the search APIs include the tags for each search result, which is crucial for SOSEM to operate. Third, they offer a diverse range of subjects for an ambiguous search query and they are not clearly biased towards a particular domain. Fourth, the search engines are restricted to just the tags, description and title of each video, whereas other keyword-based search may use additional textual content of documents to improve results. This ensures that it is a direct comparison with keyword-based search as opposed to more elaborate techniques. I have selected two social video sharing sites in particular: YouTube and Vimeo for the evaluation. The aforementioned characteristics of YouTube and Vimeo make their initial search results a suitable baseline for comparison to semantic search.

A domain ontology has been enriched with the relevant terms and concepts extracted from Social Media (for details, see chapter 3). The resulting enriched ontology should be better at disambiguating and filtering Social Media search results due to the reduced gap, both conceptually and lexically, between the domain ontology and online Social Media community. A search query that is about a concept that was not present in the original ontology would not acquire a biased interpretation by the domain ontology (see section 4.1, for details). This would in turn degrade the performance of the ontology-supported filtering process. We thus expect higher precision and recall values for the enriched ontology when compared to the original domain ontology.

In order to create a test set of terms to be used in the experiment, all of the lexicalizations which are part of the enriched LT4eL domain ontology (Lemnitzer et al., 2008) using DBpedia (Bizer et al., 2009b) have been considered. Lexicalizations which yielded more than a single DBpedia resource have been determined to be ambiguous. 23.4% of the lexicalizations had more than one concept with a matching lexicalization according to DBpedia version 3.7. This is a surprising result for an ontology about computing, because we expected most terms to be highly specific and unlikely to be shared by other domains. The average number of word senses acquired for all ambiguous terms was close to 3.4.

Lexicalizations that generated at least 20 resources in YouTube or Vimeo were selected from this list to be part of the evaluation. Multi-word lexicalizations were discarded, because most

ajax	anchor	apple	cgi
chrome	client	css	delicious
dom	dos	dtd	java
lan	lisp	lom	loose
mouse	opera	owl	python
rdf	rest	safari	sax
scrum	soap	spam	uri
vista	wan	xls	xsl

Table 5.1: A selection of ambiguous salient terms extracted from the enriched LT4eL ontology on computing (Lemnitzer et al., 2008).

tag-based social media do not support whitespace characters and order among tags has no interpretation. As part of the evaluation set, 32 ambiguous tags were selected. Each of the tags constitutes a query and these queries have been executed with SOSEM. The search results have been analyzed twice: once using an enriched ontology and once using an unenriched ontology for each of the two social video sharing sites. The ambiguous tags that were used are listed in table 5.1. It may not be obvious, but each of the ambiguous terms has at least one sense related to computer technology. SOSEM has been validated using an enriched domain ontology about computer technology. This made it necessary to select computer technology-related terminology in order to be able to show the impact and usefulness of ontologies to support semantic search. It was manually validated whether each term yielded enough resources in either YouTube or Vimeo.

Five of the terms listed in table 5.1 have been sent individually to YouTube's search API[5]. The full set of 32 queries was validated with Vimeo. Each search result consists of a title, a number of tags and a URL to the video. The available data is thus very similar to that of the example in section 4.7. The first 20 search results of the single keyword queries have been independently determined to be either relevant or non-relevant by two domain experts for both YouTube and Vimeo. Both domain experts were informed that only computer related search results were appropriate. In some cases this entailed having to watch part of the video in order to determine the relevancy when both the title and the tags of the video were not sufficient. In total about 200 relevant/non-relevant judgments were gathered from each domain expert for YouTube and 3200 for Vimeo.

Results The average precision and recall scores that have been obtained for the queries listed previously are shown in tables 5.2 and 5.3. The recall of the original unfiltered results is assumed to be 1.0, because SOSEM only post-processes the output of an external keyword-based search engine. 'Initial relevance' in tables 5.2 and 5.3 refers to the ratio of relevant resources in the context of domain ontology of the original unfiltered search results. 'Recall' refers to the number of relevant results retained after filtering using SOSEM. 'Precision'

[5]YouTube discontinued the use of tags in August 2012 while running these experiments which made additional evaluation activities impossible. `http://apiblog.youtube.com/2012/08/video-tags-just-for-uploaders.html` retrieved 12-07-2013

	Original			Enriched		
	Expert 1	Expert 2	Average	Expert 1	Expert 2	Average
Initial relevance	0.21	0.22	0.22	0.21	0.22	0.22
Recall	0.92	0.92	0.92	0.83	0.72	0.77
Precision	0.47	0.31	0.39	0.89	0.60	0.75
F-score			0.54			0.75

Table 5.2: Precision and recall values for ontology-supported semantic search using 5 single term queries on YouTube.

refers to the ratio of correctly identified relevant or irrelevant resources by SOSEM. Ideally, the precision of semantic search increases considerably with only a minimal impact on recall, i.e. few false negatives. All results are presented for SOSEM using the original domain ontology and the enriched domain ontology presented in Monachesi and Markus (2010b)[6]. For more information about these two ontologies see chapter 3.

The results in table 5.2 show an increase in precision when applying the original domain ontology for selecting relevant resources from YouTube. This positive trend continues with the additional application of the enriched domain ontology. The average position of the relevant search results was 10.5, $\sigma = 5.6$ for the YouTube results. 1 signifies the top of the search result and 20 the bottom. This means that relevant or irrelevant results are not located at a particular position in the search results. It is thus relevant to apply semantic filtering, because irrelevant results can be among the most prominent search results.

The performance for Vimeo follows a similar trend. The results in table 5.3 also show a significant increase in recall and precision when the enriched ontology is used instead of the original domain ontology. The inter-annotator agreement between the two experts is quite high ($\kappa = 0.92$). About two-thirds of the queries, listed in table 5.1, consist of search results which are either all relevant or all irrelevant. This might skew the results. I have therefore selected a subset of 13 queries[7] that each have at least 5 relevant or non-relevant results among the first 20 results. The *overall* accuracy refers to the full set of 32 terms and the *mixed* accuracy refers to the 13 queries just mentioned.

There is an important reason for the difference in performance between the enriched and unenriched ontology for both YouTube and Vimeo: A query that contains a lexicalization of a concept that is not part of the original unenriched ontology will not receive a biased interpretation from the domain ontology (section 4.1), whereas it will receive it when using the enriched ontology. Additionally, the precision of SOSEM increases when the domain ontology contains more lexicalizations of a concept. One of the intended results of ontology enrichment (chapter 3) is the addition of the community vocabulary to existing concepts by

[6]Note, that I have used the enriched ontology reported in Monachesi and Markus (2010b), which is based on DBpedia 3.7, because this ontology has been used for the YouTube evaluation. YouTube's discontinuation of tags as part of its API and user interface did not allow me to repeat the experiments with the latest ontology enrichment based on DBpedia 3.8 as presented in chapter 3. The use of the exact same ontology allows us to directly compare the results for Vimeo and YouTube without introducing different ontologies as another variable.

[7]The terms associated with these 13 queries are: ajax, chrome, client, java, lan, lisp, mouse, rdf, rest, spam, vista, wan

	Original						Enriched					
	Expert 1		Expert 2		Average		Expert 1		Expert 2		Average	
Type	Overall	Mixed	Overall	Mixed	Overall	Mixed	Overall	Mixed	Overall	Mixed	Overall	Mixed
Initial relevance	0.32	0.48	0.32	0.48	0.32	0.48	0.32	0.48	0.32	0.48	0.32	0.48
Recall	0.22	0.32	0.23	0.32	0.23	0.32	0.86	0.84	0.82	0.83	0.84	0.84
Precision	0.73	0.71	0.73	0.70	0.73	0.71	0.83	0.73	0.81	0.72	0.82	0.73
F-score					0.34	0.44					**0.83**	**0.78**

Table 5.3: Precision and recall values for the ontology-supported semantic search based on 32 single term queries on Vimeo for overall results and 13 queries for the mixed precision and recall.

means of additional lexical entries.

The false negatives that appear during concept filtering are in many cases due to poor resource tagging. In these situations the disambiguation algorithm is prevented from associating the correct word sense. In cases where the first search results from original search results do not contain enough relevant resources, query rewriting can be performed.

The combination of ontology-based semantic search with social video sharing websites, makes it difficult to compare the SOSEM-approach to other approaches discussed in the literature. This particular evaluation has focused on one domain ontology and two Social Media sites. However, it is important to emphasize that SOSEM can be used with new domain ontologies and sites if required. Even though a direct comparison is difficult, it is still worthwhile to relate its performance to the state of the art. Schütze and Pedersen (1995) note a 14% increase in retrieval performance when employing semantic search, Egozi et al. (2011) achieve an 18% increase in relevance, Castells et al. (2007) achieved an absolute increase of 0.2 in precision compared to term-based search and Tran et al. (2007) report a precision of 0.69 at a recall of 0.42. The evaluation of SOSEM seems to suggest that it outperforms the state of the art both in terms of precision and recall. An important aspect is the ontology enrichment process which boosts the F-score from 0.36 to 0.84 for the 32-query Vimeo-based evaluation which is a significant improvement. As a result, the relevance of the search results using SOSEM, as evaluated on the Vimeo data, increases from an initial 0.34 to 0.69. These results highlight the effectiveness of the incorporation of a community's vocabulary and concepts in an existing domain ontology.

The evaluation shows that SOSEM increases the quality of search results. The precision of the search results is considerably higher compared to keyword-based search. The ambiguities of the single keyword queries disappear with more specific queries when query revision is applied and, as hypothesized, the amount of filtering required with SOSEM drops considerably. The recall figures suggest that disambiguation performs reasonably well and few relevant results are omitted. The results also show a big increase in performance for the enriched ontology over the original unenriched ontology. This supports the hypothesis of chapter 3 about the enriched ontology reducing the gap between the expert approved domain ontology and the community's point of view as embodied by data coming from Social Media.

6 Conclusion

The SOSEM approach has been presented, this is an ontology-supported semantic search method that improves Social Media search results. It takes advantage of existing search engines and improves precision of their results via semantic analysis through disambiguation and query rewriting using a domain ontology. I have applied SOSEM in combination with the YouTube and Vimeo tag-based search engines and improved on their results for various computer-related search requests using the enriched and unenriched LT4eL ontology. A domain ontology was employed in order to determine the proper concepts in response to an ambiguous term.

The semantic search process is performed as post-processing of an external tag-based search engine and is thus loosely coupled and reusable. The previously developed techniques of

Social Ontology Enrichment (chapter 3) and tag disambiguation (chapter 4) have been reused as the principal components of the SOSEM semantic search approach. Unique about SOSEM is its combined use of arbitrary domain ontologies and reference repositories, support for arbitrary tag-based Social Media search engines and its exclusive use of tags as opposed to document content. Although I have exclusively focused on the use of tags with SOSEM, the methodology is compatible with, and in some cases complementary to, other approaches to search.

The evaluation of SOSEM shows that ontology enrichment through Social Media analysis does not just lead to new conceptual structure, but significantly improves the quality of semantic search applied to Social Media itself. Enrichment results obtained from one Social Media site (delicious.com) are transferable to other sites, i.e. YouTube and Vimeo, and improve results considerably. The use of correct but unpopular terms for concepts can be automatically compensated for using the alternative lexicalizations of identified concepts. Ontology-supported semantic search for Social Media is thus able to mediate between preferred formal terms provided by a domain ontology and the vocabulary of online communities.

Chapter 6

Learning with Topic Models

1 Introduction

In lifelong learning (LLL) and self directed learning (SDL), it is important to have the ability to reflect on one's learning process and answer questions such as "Can I switch to a more advanced topic?" or "How much more effort will it take to fully understand this piece of information?". Reflective processes, which are an important aspect of effective learning, can be supported by computer-based tests and tools. However, these aids are not always available, especially outside of a learning institute or university. Would it be possible to automatically construct such aids from the learning objects themselves? One way to construct these aids is through analysis of the language contained in the learning objects, because there is a strong relationship between having the right sort of conceptual knowledge and using a particular vocabulary. For instance, the ability to effectively search is constrained by lexical knowledge (see chapter 5 for details). One can imagine being able to distinguish between an expert and a novice by means of their choice of words, i.e. their vocabulary.

Previous chapters have dealt with the technical problem of mediating between the vocabulary and conceptualization of a learner and that of a community or group of experts. In the context of those chapters, an expert's conceptualization, i.e. a domain ontology, is integrated with that of a community, i.e. a folksonomy, by means of the metadata (tags) associated with the relevant resources from Social Media. As a result, ontologies become accessible to novices via the partial integration with the folksonomy (chapter 3) and enriched ontologies provide improved access to information from Social Media (chapter 5). The goal of this chapter is not necessarily to mediate between different types of conceptualizations and vocabularies, but to understand what lexical and conceptual differences reveal about a learner's level of domain knowledge and how this information can be elicited from learners by means of the content of documents.

In this chapter, I present a computational approach, TOMOFF, for personalized knowledge assessment for arbitrary domains. TOMOFF allows for the automatic identification of a learner's level of conceptual knowledge. The approach is supported by a *learning corpus* of

learning objects, i.e. text documents relevant to the domain written by experts. The learning corpus represents the gold standard with respect to the conceptual associations one is supposed to make and the proper vocabulary used to express them. Computational *topic models* play a central role with the respect to the analysis of the learning corpus in TOMOFF. Topic models, and more generally vector space models, are constructed by computer algorithms and can be used to automatically generate high level summaries of document collections.

Topic models in TOMOFF are employed to automatically construct a personalized assessment task that learners can use to quickly evaluate their domain understanding using only 30 to 40 words of input. As a result, the simplicity of TOMOFF allows learners to quickly and easily assess their level of understanding using a combination of a learning corpus and an innovative use of topic modeling algorithms. Evaluations performed in the context of academic courses over a period of two years show good correlations with actual learning outcomes of up to r=0.59**[1].

This chapter starts with an introduction to the use of language technology for e-learning in section 2, in order to provide a theoretical background for this approach. In section 3, an introduction to topic models is provided. This is followed by a review of the state of the art in topic modeling in section 4. The TOMOFF methodology, a combination of rating elicitation, topic models and a topic labeling task, is presented in section 5. The overall setup of the evaluation of TOMOFF is presented in section 6 and the results of that evaluation and analysis in section 7. Finally, section 8 summarizes the overall results and the major conclusions.

2 Learning feedback through language analysis

The educational practice of building learning support systems is shifting from pedagogically orientated approaches that focus on acquiring a fixed curriculum to just-in-time (JIT) and lifelong learning (LLL) approaches (Collis and Moonen, 2002). JIT and LLL both rely on a large body of accessible learning objects that target a specific area of interest or skill. The learning objects can be accessed using social networks and social bookmarking services (Marlow et al., 2006) or regular search engines. Social networks and collaborative bookmarking systems are a natural fit for self directed informal learning since they allow an almost unprecedented amount of personalization. Current approaches to self directed learning aim to suggest relevant documents tailored to a specific task or a person's interests.

However, from a learning perspective, the personalization should also take a learner's background knowledge and learning goals into account (Ley et al., 2010). Additionally, the convergence to online self directed learning requires increased assistance in the form of learning analytics in order to assist learners in evaluating their performance and progress (Butcher and Sumner, 2011). Most of the issues related to the support of learning, e.g. staff overload, also apply to formal learning settings in academia, which is the primary evaluation context of the proposed system. Software technology for learning should not just focus on enhanced methods for information retrieval. "We must also provide scaffolds that enhance critical thinking

[1] The following significance levels are used in this chapter: for $p \leq 0.05$ a single '*' is used and for $p \leq 0.01$ two stars ('**') are used and for $p \leq 0.001$ three stars ('***') are used.

and reflection about that information" (Lin et al., 1999, p.44). However, it is not so clear how conceptual knowledge that has been acquired through JIT, LLL and traditional academic environments should be automatically assessed and how learners should be supported in this dynamic environment.

An important aspect of "critical thinking" is the ability to reflect[2] on the learning process (Seale and Cann, 2000; Chi et al., 1994), e.g. the ability of learners to identify gaps, strengths and weaknesses in their personal learning process. Meta-cognition in the context of learning refers to a learner's "cognitive judgments about their own cognitive states and abilities" (Paris et al., 1990, p.16). Meta-cognition from an applied pedagogical perspective is not concerned with the psycholinguistic representation of concepts, but intentionally limits itself to those aspects concerned with learner's actions and choices as part of the learning process. For example, a student that has spent three hours reading two pages of text, but independently identifies that he still does not properly understand the text is said to exhibit meta-cognition. Similarly, a teacher can assist a student with critical thinking and self-reflection without having a working theory on how the mind operates and how concepts are represented and manipulated.

Generally, the concept of meta-cognition (Flavell, 1979; Lai, 2011) is divided into two components:

- Knowledge of cognition
- Regulation of cognition

The *regulation of cognition* involves the actions taken by the learner in response to awareness of, for example, poor comprehension of a learning object. This can lead to the adoption or revision of the reading strategy, planning, and so forth. It involves the modification of the behavior and the selection of alternative strategies for coping with the observation made. The regulation of cognition involves "planning, implementing, monitoring, and evaluating strategy use" (Schraw and Sperling Dennison, 1994, p.471). The complementary other aspect of meta-cognition is *knowledge of cognition*. This involves knowledge of "ones strengths and weaknesses, knowledge about strategies and when to use those strategies" (Schraw and Sperling Dennison, 1994, p.471).

Schraw and Sperling Dennison (1994) studied further decompositions of meta-cognition. Although more fine-grained distinctions between different meta-cognitive processes are theoretically attractive, their results suggest that more elaborate classifications of meta-cognition cannot be clearly differentiated in an experimental setting. Schraw (1998), on page 114, distinguishes among three different kinds of meta-cognitive awareness with respect to knowl-

[2]There is a plethora of definitions of 'reflection' with respect to learning with different nuances: "Reflecting on action is the thinking that takes place after an event or experience. It is thinking back on what we have done in order to discover how the knowledge we put into action may have contributed to an unexpected outcome. (Seale and Cann, 2000, p.310) or alternatively; "the vagueness about reflection ... exists because the term reflection describes both a cognitive process and a structured learning activity. We define reflection as the intentional consideration of an experience in light of particular learning objectives." (Hatcher and Bringle, 1997, p.153).

edge of cognition:

- "Declarative knowledge refers to knowing "about" things."
- "Procedural knowledge refers to knowing "how" to do things"
- "Conditional knowledge refers to knowing the "why" and "when" aspects of cognition."

Conceptual development can be viewed as being part of a learner's declarative knowledge. This is considerably different from skill acquisition, e.g. woodworking or typing. Therefore, the approach presented in this chapter, TOMOFF, focuses on the 'declarative' aspects of meta-cognitive awareness. Although all three kinds of meta-cognitive awareness interact, only declarative aspects will be explicitly targeted as part of TOMOFF.

Reflection and critical thinking preferably occur automatically, but they can also be triggered via the use of a formative assessment[3] and explicit prompts. Reflection strategies supported by formative feedback and prompts have a positive effect on the development of meta-cognitive skills (Van den Boom et al., 2004) and possibly learning performance (Dunn and Mulvenon, 2009). Winters et al. (2008) concludes that "adaptive scaffolding for conceptual understanding ... increased planning, monitoring, and effective strategy use in concert with improved learning outcomes." (Winters et al., 2008, p.438).

There are many approaches that seek to support meta-cognitive process by means of externalizing, i.e. by means of visualization, a learner's understanding of a domain in order to identify strengths and weaknesses. One of these approaches is the *concept map*. Concept maps usually consist of concepts expressed by terms and (associative) relations or a (simple) taxonomic structure. Concept maps allow for arbitrary levels of abstraction and detail. They originate from educational research and share considerable overlap with ontologies, but it is important to stress the differences. Ontologies are designed for generally accepted conceptualizations of domain structure, whereas concepts maps are, intentionally, uniquely generated, i.e. drawn on paper or with special software, by individual learners and serve as a succinct expression of their *personal conceptualization* of a piece of domain knowledge. Concept maps depend on the ability to graphically elicit the conceptualization of knowledge by individual learners and have been used successfully to gauge conceptual understanding (Novak and Cañas, 2008; Kinchin and Cabot, 2010). The structure of the individual concept map is argued to be indicative of the quality or depth of the domain knowledge acquired by a learner (Kinchin and Cabot, 2010; Gog et al., 2009). Others even advocate that the process of constructing a concept map (collaboratively) improves understanding and/or retention of domain knowledge (Kwon and Cifuentes, 2009).

In the context of the TOMOFF methodology, the ultimate goal is the elicitation of knowledge in order to trigger strategies and actions that regulate the behavior of learners such that learning outcomes improve. More specifically, TOMOFF can establish what learners are likely

[3] *Formative assessment* is an activity during learning that aims to identify relevant content and strategies that improve individual learning. This contrasts with *summative assessment* which seeks to determine the objective outcome or result of learning activities (Huhta, 2007). Formative feedback is thus the 'output' normally obtained via formative assessment.

to know well and what they do not know well. Of course, the most significant type of information for learners is the identification of knowledge that they believe to have mastered sufficiently, while this is actually not the case, i.e. to detect when a learner has overestimated the level of knowledge acquisition. TOMOFF does not actively attempt to influence the current strategy adopted by the learner. Instead it only aims to provide additional information that can be used by a learner to moderate his or her use of strategies as part of the overall learning process.

The primary source of information used by TOMOFF to assist learners consists of language artifacts. The analysis of the language artifacts that learners use as part of their learning process can provide insight in a learner's development (Wild et al., 2010; Lemaire and Dessus, 2001). The output of such language analysis approaches is frequently on the level of 'concepts'[4] or 'topics' in order to generalize over multiple instances of similar information. This abstraction level is comparable to that of concept maps. However, in TOMOFF these abstractions are automatically extracted from a corpus of learning objects using *topic modeling* as opposed to having to be manually constructed by individual learners. Doing part of the work automatically by means of computer algorithms allows one to stimulate reflection and critical thinking in a more efficient manner.

The following section will give a general introduction of topic modeling, the central component in TOMOFF, as a computational approach for the construction of suitable feedback for learners based on collections of learning objects.

3 Short introduction to Topic Modeling

Text analysis methods frequently need to abstract from specific tokens in order to become robust at the task of identifying the semantic content of text. For example, it is better to treat the tokens 'dino', 'dinosaur' and 'dinosaurs' as referring to a single concept in spite of their lexical variation. Part of these lexical differences can be resolved via stemming and/or lemmatization. Concepts can be grouped together in order to form a *topic*, i.e. a set of semantically related concepts. Although the problem of lexical variation can be addressed using conventional means, identifying that 'allosaurus' and 'triceratops' belong to the same topic is, however, less trivial. This can be addressed with the help of topic models.

An effective method of solving the issue of establishing a relation between terms and topics is through *topic modeling*. Topic modeling or topic inferencing are computational techniques for deriving a fixed number of *topics* from a text corpus. With topic modeling it can be automatically learned which terms are good indicators of the presence of a particular topic in a text and which are not. Topic modeling techniques are domain independent, language independent, unsupervised and are not necessarily limited to text analysis problems alone (Wang and Grimson, 2007). Documents are treated as bag-of-words and are used to infer a distribution over topics, i.e. for each topic a probability is calculated that represents the chance that it is expressed in the document. Note that the order of the terms in a document

[4] Be aware though that these 'concepts' are not always concepts in the Semantic Web sense nor are they necessarily embedded in an ontology

becomes irrelevant when a document is represented as a bag-of-words. However, it is possible to refer to a specific word using an index, since bag-of-words are represented here as vectors.

Topic modeling algorithms extract topics and the corresponding terms automatically by considering the distribution and co-occurrence of terms in documents as part of corpus. Many different topic modeling algorithms are available, each with different capabilities and performance characteristics. In this section we will consider one topic modeling framework in particular, i.e. LDA (Blei et al., 2003), because most other algorithms are extensions of it. A topic is a distribution over a fixed vocabulary, e.g. each term in the fixed vocabulary has a probability of being generated by the topic. Semantically related terms need not have morphological overlap, but are grouped together as a function of their respective distribution in a corpus, i.e. it is assumed that semantically related terms tend to appear in documents together. This knowledge-poor and probabilistic methodology with respect to the latent semantics of text is a generic one, any corpus is assumed to share this fundamental characteristic[5]. Topic models establish semantically meaningful sets of terms (topics) that can be understood by people.

In contrast, although techniques such as TF-IDF[6] and keyword extraction also identify salient terms in a corpus of documents, they (1) do not explicitly establish semantically meaningful clusters of terms and (2) do not provide a transparent probabilistic framework of the relationship between the vocabulary and the topics. For example, keyword extraction can be used to distill semantically salient terms from documents. However, keyword extraction does not establish how the salient terms from one document relate to those in another one, which topic models do, using a solid probabilistic framework. Topic models and more generally vector space models (Turney et al., 2010) are thus a more robust abstraction of a document than that provided by TF-IDF (Dredze et al., 2008) and/or concept-based query expansion, because they are less sensitive to specific terminology and require no data or training sets other than the corpus itself (Yi and Allan, 2009).

Any resource or corpus can be described by a distribution over topics. Each topic in turn can be represented as a distribution over terms. Any resource is thus assumed to exhibit topics in some distribution. The primary advantage of decomposing documents into topic distributions is that two documents may overlap in their topic compositions even if they do not have any words in common. Topic modeling has been primarily used to make large document collections available via their content on the abstraction of topics in addition to keyword based search or social recommendations. It can also be employed to quantitatively establish the topics contained in individual documents.

An alternative way of understanding topic modeling approaches[7] is to view them from a generative perspective. That is to say; what parameter-values need to be learned in order to maximize the likelihood that the topic model *generated* the corpus? Every corpus has a *vocabulary*, i.e. the set of terms in the corpus that is used to express the information. The

[5] In a similar fashion Zipf's law (Zipf, 1932), the logarithmic distribution of word frequencies in natural language corpora, is also considered as a fundamental characteristic and of many other phenomena as well (Adamic and Huberman, 2002).

[6] TF-IDF is a common method used in information retrieval. It assigns a score to terms calculated by multiplying the **t**erm **f**requency of a term t within a document or resource r with the **i**nverse **d**ocument **f**requency of a corpus R, i.e. $|\{r \in R : t \in r\}| \times log \frac{|R|}{1+|\{r \in R : t \in r\}|}$

[7] More specifically, "Latent Dirichlet Allocation"-based approaches (Blei et al., 2003)

documents in the corpus are represented as bags-of-words, i.e. sets of pairs (word type, frequency) in which word order is not retained. These bag-of-words are usually represented by vectors in the implementation. There are roughly three important distributions that need to be learned (1) a distribution over all terms in the vocabulary for every topic and (2) a distribution over all topics for every document and (3) an assignment of each term in a document to a topic. All these different distributions are unknown at first and need to be inferred just from the known (observed) documents in the corpus. The goal is thus to find the parameters for all three unknown distributions that maximize the likelihood of having generated the documents in the corpus.

For example, a topic about dinosaurs should have a high probability of generating terms like 'dinosaur', 't-rex' and low probabilities for terms like 'household' or 'web-technology'. The 'correct'[8] topic also needs to be associated with each term in the document. Finally, the topic about dinosaurs itself should have a high probability for a document about dinosaurs, such that (from the generative perspective) it can generate the correct terms and thus the input document. The other topics which make up the topic model should, at the same time, be assigned low probabilities given a document about dinosaurs. Similarly, a document on two different topics should result in two topics with high probabilities and the rest with low probabilities.

After having trained the topic model it is possible to decompose arbitrary documents into topic distributions, i.e. for each topic in the topic model a probability is assigned that the terms in the document have been generated by that topic. This topic distribution can be represented as a vector $\vec{\theta}_d$ where each topic is consistently assigned to a fixed position in the vector. Each document may express any number of topics supported by the topic model. However, only a few of these topics will 'stand out'. E.g. of the 1000 topics detected in a document (most with very low probabilities) perhaps only 3 have a probability significantly higher than 0. Put otherwise, every topic supported by the topic model will be detected in any document, but the overwhelming majority of the topics will have a probability very close to 0. For example, the Wikipedia article about the 'Allosaurus' will be decomposed into a list of topic probabilities partially displayed in table 6.1.

Each line in table 6.1 describes information about one topic present in the document. The second column in listing 6.1 is the probability that the topic is present in the document. Observe that the first 16 topics do not sum to 1.0, but to approximately 0.72 which means that there will be a large tail of topics with very low probability values. The third column lists terms and values that express the probability that the term contributes to the presence of the topic. Topics have no inherent title such as a library category or ontology might have, but the content of a topic is frequently expressed, for humans, using a topic's most salient words.

More formally let the topics be $\beta_{1:K}$, where each β_k is a distribution over the vocabulary for a topic k. The topic proportions for the dth document are θ_d, where $\theta_{d,k}$ is the topic proportion for topic k in document d. The topic assignments for the dth document are z_d, where $z_{d,n}$ is the topic assignment for the nth term in document d. Finally, the observed terms for document d are w_d, where $w_{d,n}$ is the nth term in document d [9]. The inference algorithm, frequently the

[8] Because LDA is an unsupervised machine learning algorithm there is no objective function that labels topic attributions as either correct or incorrect.

[9] This notation is based on the notation used in (Blei, 2011)

topic	probability	most salient words and their probability
1	0.376743887593	0.028*formation + 0.028*fossil + 0.021*specimen + 0.020*dinosaur + 0.019*extinct
2	0.0636178492227	0.161*ship + 0.060*submarine + 0.025*patrol + 0.024*navy + 0.022*vessel
3	0.0621609952826	0.057*body + 0.036*bone + 0.024*pain + 0.022*injury + 0.022*muscle
4	0.0266637192253	0.041*theory + 0.010*effect + 0.007*difference + 0.007*concept + 0.006*process
5	0.0234474913037	0.142*specie + 0.054*genus + 0.035*linnaeus + 0.028*taxonomy + 0.022*zeller
6	0.0207676189202	0.011*experience + 0.009*relationship + 0.007*positive + 0.007*feel + 0.006*sense
7	0.0175311737275	0.015*weight + 0.010*size + 0.010*bgcolor + 0.008*type + 0.007*length
8	0.0171104976223	0.260*seattle + 0.155*wa + 0.120*yale + 0.055*dawson + 0.042*alexandria
9	0.0164161062568	0.278*signature + 0.133*blake + 0.104*var + 0.036*carbuncle + 0.033*troll
10	0.0155655054702	0.033*fly + 0.031*larva + 0.030*egg + 0.026*feed + 0.020*monkey
11	0.0151986787278	0.161*science + 0.128*research + 0.070*institute + 0.068*technology + 0.050*journal
12	0.0140922239102	0.185*isbn + 0.121*ed + 0.089*press + 0.076*pp + 0.062*vol
13	0.0131666961783	0.198*ru + 0.158*fr + 0.130*e + 0.060*pl + 0.056*tr
14	0.01240855282	0.116*william + 0.063*robert + 0.025*thomas + 0.024*james + 0.020*henry
15	0.0107717303808	0.181*col + 0.149*lt + 0.094*morrison + 0.094*bg + 0.041*flora
16	0.010576362712	0.351*pa + 0.077*macedonian + 0.077*macedonia + 0.077*je + 0.064*loop

Table 6.1: The first 16 topics with the highest probabilities generated by an LDA 1000 topic model trained on the English Wikipedia for the article 'Allosaurus' with the time stamp 5-12-2011 20:13.

EM-algorithm (Dempster et al., 1977), then needs to approximate the hidden variables z_d, θ_d and $\beta_{1:K}$ given w_d. Equation 6.1 displays the important statistical interrelations between the various distributions that define LDA that I have just discussed. It succinctly states that the topic assignment $z_{d,n}$ depends on the per-document topic distribution θ_d and that the observed terms in the documents $w_{d,n}$ depend on the topic assignment $z_{d,n}$ and all the topics $\beta_{1:K}$. Note that N represents the number of unique terms in the entire corpus.

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D}) = \prod_{i=1}^{K} p(\beta_i) \prod_{d=1}^{D} p(\theta_d)(\prod_{n=1}^{N} p(z_{d,n}|\theta_d)p(w_{d,n}|\beta_{1:K}, z_{d,n})) \quad (6.1)$$

This concludes the generic introduction to topic modeling and more specifically, LDA-based topic modeling. The following section reviews the state of the art of topic modeling and the various extensions of LDA relevant to this chapter.

4 State of the Art

This section reviews the state of the art with respect to topic modeling in general and how it relates to e-learning. First, I will introduce topic modeling within the broader area of information extraction from text. Second, unsupervised topic modeling techniques and their applications are discussed. Third, supervised topic modeling algorithms are presented and some of their general applications are highlighted. Fourth, related work on topic labeling is covered. Finally, the contributions of the methodology and algorithms presented in this chapter are linked to and contrasted with existing work within the context of e-learning.

Information extraction from text is an active area of research represented by a wide array of approaches. An important aspect of information extraction is the extraction of semantic information from text. These methods can be roughly divided into those that are knowledge rich, e.g. ontologies (see chapter 3, section 2 for an overview of these methods), and knowledge poor. I will focus on the knowledge poor methods in this section.

There are various methods for characterizing, summarizing or otherwise representing the relevant semantic information of documents. One example of such methods is TF-IDF vectors and concept-based query expansion (Voorhees, 1994; Navigli and Velardi, 2003). Documents can also be summarized using automatically extracted keywords (Lemnitzer and Monachesi, 2008). This section focuses on a particular class of knowledge poor methods called vector-space models (Van de Cruys, 2010) and more specifically topic modeling.

Popular topic modeling techniques are (Probabilistic) Latent Semantic Indexing/Analysis (LSA, pLSI) (Deerwester et al., 1990; Landauer et al., 1998; Hofmann, 1999) and the family of algorithms based on the model provided by Latent Dirichlet Allocation (LDA) (Blei et al., 2003; Blei, 2012). The topics derived from topic modeling can be employed for navigation in large document collections (Griffiths and Steyvers, 2004), tag recommendation (Krestel et al., 2009), faceted search (Mimno and McCallum, 2007), query expansion (Yi and Allan, 2009) and automatic essay assessment (Landauer, 2003) amongst others.

Topic models are an interesting alternative to knowledge rich approaches, because they are domain and language-independent and unsupervised. Additionally, the type of lexico-semantic abstractions generated by topic models closely resemble those made by humans (Wolfe et al., 1998). The evidence that topic models are able to learn proper abstractions from mere text corpora is diverse. One example is a comparison of subject headers generated by a topic model and the official ones from the Library of Congress (Noh et al., 2011). Noh et al. (2011) show that the subject headers from a topic model are better at identifying whether a group of books shares a common theme than manually assigned subject headers. Ramirez et al. (2011) validated that the semantic coherence of topic models corresponds well to that of manually labeled data such as news corpora with category information. Ramirez et al. (2011) recasted the topic model approach as a special type of soft clustering and showed that the multi-labeled Reuters news corpus is properly described by the topics of LDA models when compared with manually assigned category labels. Liu et al. (2009) developed a system for the visualization of the presence of topics over time to assist users trying to understand complex corpora. Figure 6.1 shows an example of such a visualization of topics over time.

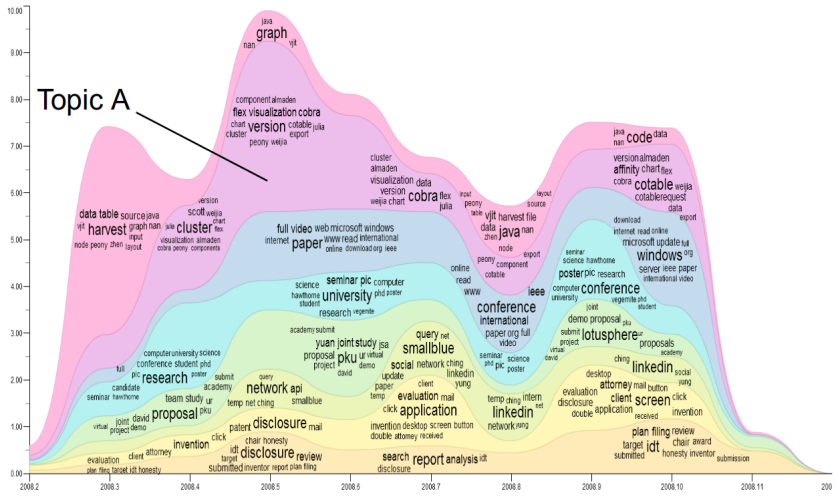


Figure 6.1: Visualisation of the presence of various topics over time in an large e-mail corpus using the TIARA-system (Liu et al., 2009)

Topic models are frequently employed to navigate in large amounts of information, e.g. millions of documents, through their topical structure (Liu et al., 2009), which requires the use of efficient approximation methods. Extending topic modeling algorithms, such as LDA, to process larger data collections requires scalable optimization and sampling algorithms such as distributed computation (Newman et al., 2007), Gibbs-sampling (Porteous et al., 2008) or online LDA (Hoffman et al., 2010). The number of studies that look at very small corpora, i.e. those with 500 documents or less, is much smaller (Wolfe et al., 1998) and at this point it is unclear whether small corpora can also result in topics that are as clear as the topics in larger corpora for use in a learning task although they can be used for documentation classification (Villalon and Calvo, 2009).

Andrzejewski and Buttler (2011) investigated whether presenting the output of a topic model alongside search results lead to improved discoverability of information in domain specific corpora. "Quantitative results on benchmark TREC datasets show that this technique can result in major improvements for a non-trivial proportion of queries" and "the presentation of enriched and related topics alongside search results can help to deliver insights about corpus themes, which may be beneficial for knowledge discovery as well" (Andrzejewski and Buttler, 2011, p.607).

Other examples that illustrate the possible applications of topic modeling techniques include the automatic identification of spoilers, i.e. comments that give away too much of the plot of a movie (Guo and Ramakrishnan, 2010), variance of salient topics over time in Science (Blei and Lafferty, 2006), identification of personas in film characters (Smith, 2013) and response prediction to political blog posts (Yano et al., 2009). In each case, topic models solve problems that can also be solved with other, more opaque, state of the art methods and techniques. However, the main advantage of topic models is that their internal structure is much easier to interpret by non-experts, whereas that of more opaque methods, such as Support Vector Machines or Neural Networks, is not.

Supervised topic models The previous paragraph has surveyed unsupervised topic modeling techniques and applications, but for some applications, supervised topic modeling offers compelling advantages. Blei and McAuliffe (2010) present an adaption of LDA called sLDA that allows for training a topic model in a supervised manner. Each document has a class label or number attached to it which the trained model's latent topics should predict. Blei and McAuliffe (2010) show that this method outperforms simple linear regression on the regular, unsupervised, topic distribution of the same corpus. Zhu et al. (2009) further improved on the work from Blei and McAuliffe (2010) with MedLDA, which employs an additional algorithm based on the max-margin principle from the existing theory on Support Vector Machines (SVM). MedLDA (Zhu et al., 2009) optimizes the topic model structure in such a way that the topic representations of different target values have an equal maximum distance separating them in a hyperplane. Associating different output values with the same documents will lead to different topic models due to the joint estimate of both the hidden variables of the LDA model and adherence to the maximum-margin constraint of the resulting feature vectors. Similarly, Rubin et al. (2011) illustrates better performance, especially on limited datasets, on multi-label document classification using supervised topic models when compared to binary SVMs.

Topic labels An important feature of topic models is that the resulting topics can be automatically visualized using their most salient words. It is not possible to visualize a topic using all of its words, because a topic is a distribution over a vocabulary of, potentially, thousands of terms. It is therefore common to use a number of terms from the vocabulary that have a high probability for a particular topic. The collection of salient terms that is used to visually represent a topic for humans is referred to as a *topic label*. Topic labels based on a topic's most salient words can be effectively interpreted by humans (Mei et al., 2007; Chang et al., 2009; Lau et al., 2011). In addition, techniques exist to further refine these topic labels automatically in order to improve their interpretability (Mei et al., 2007; Lau et al., 2011;

Hulpus et al., 2013). For example, by generating a single multi-word expression as a topic label as opposed to a collection of ten salient terms. However, these studies on the quality of topic labels and how to improve on them do not address the fact that appropriate domain knowledge related to the domain corpus is crucial for the assessment of the quality of a label. In other words, the ability to interpret a topic label depends on the level of familiarity with the content of the corpus.

Related work The previous section has reviewed state of the art approaches to topic modeling in the context of the framework of Latent Dirichlet Allocation (Blei et al., 2003). This section links the overall approach proposed in this chapter, that is based on LDA, to related work on vector-space models and techniques as applied to e-learning and assessment.

Lin et al. (1999) emphasize the importance of providing "scaffolds that enhance critical thinking and reflection". These can be automatically established for arbitrary learning objects through language technology. More specifically, Latent Semantic Indexing (LSI) and graph layout algorithms have been applied to generate 'concept map'-like visualizations of the thematic differences between user provided text and reference corpora (Wild et al., 2010). Concept maps will be discussed in more detail in section 5. Wild et al. (2010) claim that these types of visualizations support the learner during reflection, although no information is provided with regard to the proper interpretation of the visualization. Another example is the use of essay grading techniques supported by language analysis (Valenti et al., 2003). Essay grading, supported by LSA-based techniques, has been extended with detailed feedback on parts of the essay in order to automatically provide better feedback (Lemaire and Dessus, 2001; Trausan-Matu et al., 2008; Loiseau et al., 2011). More recently, ReaderBench was introduced, which is a continuation of the work from Loiseau et al. (2011); Rebedea et al. (2010a) with respect to using LDA, LSA and NLP techniques for measuring text complexity and reading strategies (Dascalu et al., 2013).

Wolfe et al. (1998) used LSA-based models to match learners with learning objects that are compatible with their existing background knowledge and level. Wolfe et al. (1998) showed that the LSA-based recommendation of learning objects performed just as well as pre-existing methods. "Results show a non-monotonic relationship in which learning was greatest for texts that were neither too easy nor too difficult. LSA proved as effective at predicting learning from these texts as traditional knowledge assessment measures." (Wolfe et al., 1998, p.2). (Wolfe et al., 1998) further note that "the representation or grouping of information in the LSA space is at a relatively basic or introductory level.". I argue that this effect is due to the use of a generic corpus instead of a specialized learning corpus and I explicitly address this as part of the evaluation of TOMOFF. An overview of the applications of LSA-based techniques with respect to e-learning is available in Dessus (2009).

This chapter introduces an approach that provides "scaffolds that enhance critical thinking and reflection" automatically on an abstraction level close to concept maps, using robust topic modeling (Blei et al., 2003; Zhu et al., 2009). Formative feedback is generated that can support critical thinking and reflection through analysis of the language employed in learning objects in the same spirit as Kinchin and Cabot (2010); Gog et al. (2009), but with the important difference that concept maps-like representations are not manually crafted by learners, but instead are automatically derived from their personal learning objects. It thus

makes the latent patterns in their learning process explicit and available for reflection (Markus and Westerhout, 2011). TOMOFF allows for the generation of a highly structured task that both reduces input requirements and enforces appropriate domain coverage. This is compatible with the observation that "semantic associations are not per se the optimum in terms of word knowledge development ... the free association task [is] less suitable as a test of word knowledge development and emphasizes the need for more structured tasks that specifically target (the recognition of) semantic meaning aspects of words" (Cremer, 2013, p.190). An important difference between TOMOFF and related work on essay-grading is the fact that the trained topic model is presented to learners as opposed to only using it as a machine learning tool in the background (Wolfe et al., 1998; Lemaire and Dessus, 2001; Landauer, 2003; Hasan, 2012; Jorge-Botana et al., 2010).

TOMOFF is designed to be complementary to other techniques based on essay assessment (Wolfe et al., 1998; Lemaire and Dessus, 2001; Landauer, 2003; Hasan, 2012; Jorge-Botana et al., 2010), short answer assessment (Ziai et al., 2012) and the assessment of personal blog entries (Wild et al., 2010), because it only requires 30-40 words of input from a learner. TOMOFF's objective is not to maximize agreement with human graders using a supervised learning approach, but is instead intentionally limited to the use of only a learning corpus alone, i.e. without any explicit teacher or expert feedback. This allows TOMOFF to be used in today's time constrained and just-in-time (collaborative) learning environments that lack institutional support. TOMOFF has been evaluated within two university courses over a period of two years and its performance has been compared to the actual learning outcomes as measured through traditional exams. The strength of the correlations with learning outcomes is slightly less than some of the essay-based evaluations, which have $r \approx 0.6$ vs $r \approx 0.7$[10], (Wolfe et al., 1998; Lemaire and Dessus, 2001; Landauer, 2003) (see section 7), but this is to be expected given the severely reduced input, lack of any expert feedback to be used for supervised learning and the evaluation context of TOMOFF. The intentional limitations of TOMOFF with regard to the data that it is allowed to use also represent its major strengths and increase the likelihood of user adoption.

5 TOMOFF design

This section presents the TOMOFF approach to formative assessment using topic models. Section 5.1 will give a broad overview of the TOMOFF approach and its three main components: rating elicitation (section 5.2), topic model construction (section 5.3) and the manual topic labeling task (section 5.4).

5.1 Overview

Concept maps have been effectively used in the past in order to gauge the domain knowledge of individuals (Markham et al., 1994; Novak and Cañas, 2006, 2008; Kinchin and Cabot,

[10] Actually, there is a huge amount of variability in the literature with regard to the reported correlation with manual human grading of essays. At present, a proper meta-analysis of the literature on LSA-based essay grading is not available. The reported average of 0.7 is a reasonable average of the quoted studies that made use of relatively small corpora and those that appeared to be methodologically sound.

2010; Gog et al., 2009). However, concepts maps are considerably different from computational topic models, because they are the result of a person instead of a computer algorithm. However, it is known that rudimentary concept maps can be automatically constructed with the help of language technology (Wild et al., 2010) from a learner's language artifacts, e.g. blog posts. The advantage of these methods is that they no longer require the learner to engage in an additional activity, e.g. a formative assessment task. The language artifacts associated with, or created by, the learner alone are presumed to contain all the information required to skip the formative assessment task itself and to immediately provide formative feedback in the form of a concept map. The method of triggering high level reflective processes in learners via concept maps provides a way of structuring the feedback. Techniques conceptually related to concept maps can be leveraged and improved by the automatic construction of such representations and can be used to present them to the learner directly. The internal dialogue and 'sensemaking' by the learner then triggers a reflective process that leads to beneficial adjustments of the learning process (Butcher and Sumner, 2011). Additionally, "students prefer feedback from a computer because it enables students to make mistakes in private and is impersonal and non-judgemental" (Jordan, 2011). Computer-supported assessment and feedback is more common in the formal sciences, but language technology, such as topic modeling, holds the promise of transferring automatic computer feedback to arbitrary domains. One of the reasons that the aforementioned work operates on an abstract level close to concept maps has to do with the generation of feedback. With the current state of technology it is still not feasible to generate personalized natural language expressions containing formative feedback. Providing feedback on a more abstract level akin to concept maps is however possible, with the advantage that it is very compact, does not require a sophisticated tailored grammar and is relatively straightforward to generate.

However, an important difference that exists between concept maps and traditional topic models is that topic models are automatically inferred from common language artifacts, whereas a concept map is manually constructed by the learner. Stated differently, a concept map is an inherently personal artifact whereas a traditional topic model is a generic representation of the lexico-semantic patterns in a text corpus. In order to leverage the advantages of concept maps, a type of personalization needs to be integrated with the topic model itself. The fact that the language artifacts and metadata associated with an individual learner are used as opposed to a generic corpus will, I propose, lead to a variation in the structure of the topic model similar to the variation observed in manually constructed concept maps.

Traditional applications of topic modeling have so far concentrated on modeling the content of a generic corpus, as previously discussed in section 4. In the process, individual differences in how documents are interpreted, i.e. the relevant topics in the exact same corpus are likely to vary between learners, have been ignored. These individual differences might not be relevant in a generic information retrieval context, but they are relevant in a learning context.

I hypothesize that individual variation between a learner's topic models for the same documents is indicative of his or her conceptual development (Wolfe et al., 1998) and that an appropriate topic model can be used to support the meta-cognitive processes of a learner. Meta-cognitive processes are especially important in LLL and self directed learning environments. Additionally, the incorporation of individual characteristics in the topic models should improve the correspondence between the topic model and an individual learner's domain understanding. I submit that computational topic modeling can be used to automatically

create feedback of a form that is similar to that established through concept maps, i.e. on a similar level of abstraction. However, instead of relying on complex domain conceptualizations, a minimalistic approach is taken that combines conceptual understanding with lexical knowledge of the domain of interest.

Whereas some approaches to the application of language technology in assessment and feedback focus on the analysis of artifacts produced by learners (Wolfe et al., 1998; Lemaire and Dessus, 2001; Wild et al., 2010; Trausan-Matu et al., 2008), TOMOFF takes a different position and assumes that learners produce *few* linguistic resources themselves, but leave a trail of the *many* resources that they access and annotate (see chapter 2, section 3.3 for details). TOMOFF uses the annotations of resources by learners to infer important characteristics that can be used to generate formative feedback in later stages. Academic courses with limited amounts of students and learning objects is the main evaluation context of TOMOFF and it shares many commonalities with JIT and LLL situations. I will refer to such a small corpus, i.e. less than 500 documents, with learning resources as a *learning corpus*.

Domain understanding is measured in TOMOFF as a function of a learner's ability to interpret a visualization of a topic model. From an ontological point of view, a topic can be viewed as a collection of interrelated concepts[11]. A good interpretation of a topic would be the minimal set of super-concepts that subsume the salient concepts in that topic. In this sense, it is clearer how the knowledge-poor conceptualization of domain knowledge provided by a topic model compares to ontologies used in chapters 3 and 5. Ontologies are more precise and reliable, because the relations between concepts are known, which they are not to a topic model. The same elements of lexical and conceptual domain knowledge re-appear in TOMOFF in the context of topic modeling. The quality of a learner's understanding of a topic model is measured by having them create a short summary of the topics, i.e. to *label* the topics. The quality of this label is then automatically assessed using the topic model itself.

The TOMOFF methodology consists of the following three steps:

1. *Rating elicitation* (section 5.2)
   The goal of the first step, rating elicitation, is to acquire a personal estimate of how well the learner thinks he has understood a learning object and to elicit important information about how a specific learner views the learning corpus.

2. *Topic model construction* (section 5.3)
   The information from the rating elicitation step is used to construct a topic model that is optimized with respect to a specific learner or a group of learners.

3. *Topic labeling* (section 5.4) Topics are presented to the learner in the form of a 'topic labeling task'. The performance of a learner on this task is indicative of the level of knowledge acquisition.

The following sections introduce each of these three steps in greater detail.

[11] This is a broad generalization in order to highlight the interrelations between concepts from an ontology and the type of lexico-semantic representation constructed by a topic model.

Difficulty	Description
Trivial	A completely trivial useless resource
Easy	Easy to understand, but I knew more about the subject than the resource contained
OK	Easy to understand
Challenging	I learned something new or understanding it took some effort.
Hard	I could not really understand this / Incomprehensible.

Table 6.2: Ratings available to learners when providing feedback in TOMOFF about the difficulty of a learning object.

5.2 Rating elicitation

The rating elicitation step is about getting learners to provide feedback on the documents that they have read as part of a learning corpus. The ratings do not have to be integrated within the actual document itself, but can be represented in a method similar to stand-off annotation[12]. This allows for the use of documents which are not locally stored, e.g. remote webpages. The elicited ratings will be used in combination with supervised topic modeling algorithms to construct personalized topic models based on the learning corpus (section 5.3).

By default, topic modeling techniques aim to create topics that match the categorization that humans would perform in a card-sorting task Rugg and McGeorge (1997). However, this concerns only the generic conceptual organization of a corpus, whereas there are likely to be individual differences which can only be incorporated if there is some amount of learner feedback about the documents in the corpus. More specifically, every individual will vary as to what information is considered important or difficult. This information is not present in the corpus itself, but is the result of an individual's conceptualization (for details, see chapter 2, section 4.1). The ratings act as the simplified output of the meta-cognitive processes that the learners employs when deciding on the difficulty of the resource. The form in which learners express their feedback is intentionally simplistic and only requires users to rate the resource, i.e. express their declarative knowledge, on a 5-point Likert scale.

As a result, each document receives a rating that represents a subjective level of comprehension, i.e. an indicator provided by the learner on the amount of conceptual struggle required to comprehend the resource. The interpretation of the difficulty ratings as suggested by TOMOFF to learners is listed in table 6.2. These ratings mirror the staged model of gaining expertise Dreyfus (2004) as summarized by (Eraut, 1994, p.124) displayed in table 6.3.

The rating used by users actually represents a moving target relative to the learner's current stage of development instead of a static reference. For example, a learner at stage '1' may assign a rating of '5' to a resource, which 'objectively' targets an audience in stage '3'. Implicitly, the rating given by a user reflects the subjective assessment of what the user believes to be the required stage of knowledge acquisition. As a result, we expect a significant amount of noise in the ratings that learners assign to resources.

The quality of this feedback in the form of ratings depends on the assumption that learn-

[12] Stand-off annotation is a method of enriching a resource with additional metadata without altering the resource itself, i.e. the annotation of the resource resides in a location different from that of the actual resource.

(1) **Novice**. Rigid adherence to taught rules; little situational perception; no discretionary judgment.

(2) **Advanced beginner**. Guidelines for action based on attributes or aspects; situational perception still limited; all attributes and aspects are treated separately, with equal importance.

(3) **Competent**. Coping with crowdedness; actions seen at least partially in terms of long-term goals; conscious deliberate planning; standardized and routinised tasks.

(4) **Proficient**. Sees situations holistically, rather than in terms of aspects; sees what is most important in a situation; perceives deviations from normal patterns; decision making less laboured. Uses maxims for guidance, whose meaning varies according to the situation.

(5) **Expert**. No longer relies on rules or guidelines; intuitive grasp of situations based on a deep tacit understanding; analytic approaches used only in novel situations or where problems occur; vision of what is possible.

Table 6.3: A summary of the five important developmental stages of a model of stage-wise gaining expertise from (Eraut, 1994, p.124)

ers can perform adequate self-assessment of their current level of knowledge about a topic. Baker (1989) argues that readers are rather bad at assessing their comprehension of both texts within and outside of their domain of expertise. Surprisingly, domain experts were shown to overestimate their text comprehension on texts from their own domain when compared to novices, whose self-assessments were actually closer to their true level of comprehension (Baker, 1989). Assuming that learners' self-assessments are quite noisy and in some cases over-estimates, does this mean that these are useless? Good performance on self-assessments mostly correlates with better overall reading skills (Baker, 1989, p.33-34). It thus seems likely that improved meta-cognitive abilities lead to better manual selection of appropriate learning resources.

Although Baker (1989) argues that the self-assessments are skewed, I hypothesize that the assessments still exhibit some overall trend that can be salvaged. For example, consider a learner that rates resources that are easy as 1 and difficult as 5, although the appropriate ratings would be 2 and 4 respectively. It is the case that the assigned ratings are heavily skewed, but they still allow us to differentiate, in a somewhat degraded manner, between easy and difficult resources. This method can be used to identify trends in a learner's feedback even if the ratings themselves are skewed when compared to some objective measure. Recall that the objective is not for learners to 'correctly' rate resources. The goal is to make use of the ratings that are available, in order to increase the relevance of the feedback of TOMOFF.

The ratings are elicited and stored in a database along with the date and time at which each rating has been submitted. The ratings are used by the next step which involves the construction of a topic model that exploits both the documents and, optionally, their ratings.

5.3 Topic model construction

A topic model is constructed using a learning corpus and, optionally, the ratings gathered in the previous step. It is created in such a way that it can be used to measure the level of acquisition of domain knowledge by a learner or group of learners. Topic models that take the ratings into account, i.e. supervised topic models, are expected to correspond better to the domain understanding of learners than unsupervised ones. The resulting topic model is to be

used in the final step, the topic labeling task.

The central component in the TOMOFF methodology is the use of a topic model to construct an abstraction over a corpus. The topic model represents a summary of a learning corpus that can be interpreted by humans, i.e. it is not a black-box classification model. An important aspect of the topic model construction is that it is based on a manually gathered learning corpus. This corpus can include, for example, research papers read during class, relevant web pages, book chapters etc. The advantage of building a topic model specifically for the learning corpus is that each of the topics is relevant and highly specific to the content in the corpus. This sharply contrasts with a generic corpus, such as those extracted from Wikipedia, as I will show in the evaluation in section 7. In a learning context, it is also important that the topic modeling process can operate with a relatively small corpus. It is known that some related approaches to topic modeling can achieve satisfactory results with small corpora (Wolfe et al., 1998; Jorge-Botana et al., 2010). This is important, because a learning corpus is relatively small compared to larger, generic corpora such as Wikipedia or those of newspapers.

The topic modeling step can accommodate various types of topic models, regardless of whether they have been constructed using unsupervised or supervised methods. Various strategies are available for training topic models with different performance characteristics. Online LDA (Blei and McAuliffe, 2010), an example of an unsupervised topic modeling algorithm, can only be trained on the documents themselves, but scales very well to large document collections. In contrast, a supervised topic modeling algorithm like MedLDA (Zhu et al., 2009) can also take advantage of additional meta data for the documents, e.g. in our set-up, the rating of a document by at least one learner. In order to assess what type of topic model is most effective in TOMOFF, four different topic models are evaluated. More specifically, an LDA-based model trained on Wikipedia, an LDA-based model trained on a learning corpus and two types of MedLDA-models trained on a learning corpus and the associated ratings. More details about which topic models have been evaluated and used within TOMOFF will be provided in section 6.

Although a number of different topic models is evaluated, each topic model can be used in approximately the same way in the following step. This generality enables TOMOFF to accommodate different types of topic modeling algorithms without requiring architectural changes.

5.4 Topic labeling task

The final step in the TOMOFF methodology is the topic labeling task. It uses the topic model constructed in the previous step in order to generate a personalized formative feedback task. The task generated is sent to a learner who attempts to solve it to the best of his/her ability. The performance of the learner on this task is indicative of the level of acquisition of the information in the learning corpus. The answers generated by the learner on the task are then analyzed by TOMOFF.

A topic model consists of a number of probability distributions (see section 3 for details). However, it needs to be visualized in order to become accessible to learners. β_k, i.e. the distribution over the vocabulary N, is of particular interest, because it can be used to visualize

Objective / Subjective	Thinks he knows	Thinks he does not know
Does demonstrably know	Correct meta-cognitive estimation	Underestimation
Does demonstrably not know	Overestimation	Correct meta-cognitive estimation

Table 6.4: A summary of the four interactions between a meta-cognitive assessment and actual knowledge acquisition.

a topic using its most salient words, i.e. words with a high probability in β_k for a topic k. More formally, let k be a topic, extracting the most salient word for that topic is achieved via $\arg\max_{n \in N} f(k,n) = \beta_{k,n}$. Similarly, multiple salient words can be extracted that together represent the global maximum of β_k. A number of salient words together constitutes an automatically generated *topic label*.

In the TOMOFF approach, the internal structure of a topic model is used to automatically construct a topic labeling task. As part of this task, the learner is presented with a number of topics visualized using the ten most salient words. The learner is required to provide a new label for each of these visualized topics to the best of their ability. For example, given the salient words 'Triceratops', 'Brachiosaurus' and 'Sinosauropteryx' a good topic label, the super-concept that subsumes all the salient words, could be 'dinosaurs'. The ability to interpret a topic's most salient words depends on both knowledge of the words themselves, i.e. concept lexicalizations (see chapter 2, section 2.1 for details), and an understanding of the important relations between those words, i.e. what the words have in common. More formally: a topic label, from an ontological perspective, should consist of the smallest set of super-concepts that subsume all of a topic's salient words.

In the context of TOMOFF, the topic labeling task requires a learner to generate a label of at most three words that reflects the theme expressed by a topic's most salient words. A learner is only able to make this transition from the most salient words to a proper label if he or she can identify the theme that the words express. We assume that this is only possible if the learner is familiar with the content in the learning corpus. A learner can provide a topic label when he thinks he knows the answer or not provide a label if he is unable to. There are four situations with respect to the topic labeling task that relate to what the learner thinks he knows and what he actually knows. These four situations are summarized in table 6.4. TOMOFF covers the majority of these situations by means of automatic assessment of the quality of a topic label. Except for the fact that it cannot properly deal with underestimation of knowledge acquisition.

Recall that section 2 mentioned that the quality of a concept map is related to the quality of knowledge acquisition. Similarly, in the topic labeling task the 'quality' of a topic label corresponds to the 'quality' of knowledge acquisition, i.e. poor or absent labels reflect poor understanding whereas good labels reflect good understanding. A naive approach to an evaluation of the topic labeling task would consist of looking at the number of topics for which the learner managed to assign a label to a topic. An unlabeled topic would then reflect poor understanding, but can also mean that the learner was too tired or unmotivated to perform the task. However, the presence of an arbitrary topic label is no guarantee for a proper interpretation of the corresponding topic summary. The actual quality of the topic label itself is also an important indicator for how well the learner performed the task. A topic with an incorrect

or inappropriate topic label should be identified as such.

In order to distinguish between inappropriate and appropriate labels an analysis of the topic label is required. The intuition behind this methodology is that a topic label should be as specific as possible. For example, given a topic model inferred from a learning corpus about computational linguistics, it is, in principle, possible to assign the label 'computational linguistics' to each and every topic. Technically, this topic label is a proper super-concept for each of the salient words in each topic, but it is not very specific. We would actually expect a unique topic label for each and every topic, because each topic is assumed to cover a different part of the vocabulary and thereby concepts. The idea of enforcing unique topic labels is also insufficient, because the unique topic labels 'computational linguistics 1', 'computational linguistics 2', 'computational linguistics 3' are all equally vague and non-specific. A topic label should only be identified as a good topic label if it is specific to the topic to which it was assigned and no other topic. Alternatively, we could state that the topic label should act as a unique identifier for only one topic and no other. A good topic label, using this intuition, has a high score, whereas a topic label that likely refers to one or more other topics, i.e. it is vague, non-specific or inappropriate, has a low score. I will now go into the specifics as to how to determine the 'degree of specificity' of a topic label with respect to a range of other topics in an arbitrary topic model (either supervised or unsupervised).

Implementation In order to quantitatively assess the quality of topic labels, a method is required to objectively 'score' each individual topic label. This method is displayed in pseudocode in listing 6.1. Please note that the explanation makes use of the notation previously introduced in section 3. The method takes as inputs the topic label of a learner, i.e. a set of words $\{w_1, w_2, \ldots, w_z\}$ (line 1) and a topic t to which it is assigned (line 2). If the topic label is empty (line 4) a score of 0 is immediately returned. If the topic label is not empty, the algorithm continues. Next, for each topic k in a topic model, the product of all probability values for each word in the label is calculated (lines 7 and 8). These probability values are part of the topic-specific distribution over the vocabulary, i.e. β_k. Using this method a likelihood value can be calculated for each topic for the input topic label. The topics are sorted according to this likelihood value (line 9). The highest value will be at the top of the list and the lowest values at the bottom. Finally, the distance between the topic to which the label is assigned and the topic with the highest likelihood value for that label is calculated (line 10). The distance is inversely proportional to the score.

A large distance means that the label given to the topic is inappropriate or vague whereas a small distance represents an appropriate topic. The topic to which the label was added, t, should be the highest one in the list, i.e. it is the most likely topic to be associated with that label. Any deviation from this fact affects the position of the topic in the list of sorted topics S. More specifically, the smaller the likelihood of the label for that topic, the lower its position in the list. The difference between the intended position of the topic and the actual position determines the score obtained for the topic label. The perfect topic label will get a score of one, because the difference in position is zero, i.e. the most likely topic is the topic to which the label has been assigned. The worst possible topic label will get a score close to zero, because the difference in position between the intended topic and the most likely topic is large.

Each score obtained for a topic label is referred to as a '*topic label quality score*'. The total score obtained for all labels of a topic model, divided by the number of topics, is referred to as the '*average topic label quality score*'.

```
1  {w1,w2,...,wz} <- input   #topic label
2  t <- input                #topic to which the label is assigned
3
4  if ({w1,w2,...,wz} = ∅):
5     score = 0
6  else
7     for k = (1..K):
8        topics[k] = ∏_{j=1}^{z} β_{k,wj}
9     S = sort(topics,desc)
10    score = 1.0 / (index(t,S) + 1)
11 end
```

Listing 6.1: pseudo-code for determining the topic label quality score for a specific label.

This concludes the presentation of the topic labeling task and the included methodology to determine the quality of an arbitrary topic label in the context of a specific topic model. This task enables one to estimate the level of knowledge acquisition of a learner with respect to a learning corpus.

In summary, the TOMOFF methodology as a whole aims to support the meta-cognitive processes employed by learners. For example, it can provide helpful feedback in support of learner's meta-cognitive processes. These objectives are achieved by TOMOFF by means of a learning corpus, feedback provided about the corpus, topic modeling applied to the learning corpus and a topic labeling task. The primary advantage of this methodology is that it is a completely unsupervised, domain independent and transparent method. The performance of a learner on the topic labeling task correlates with the level of knowledge acquisition and constitutes a type of formative assessment.

The evaluation of TOMOFF presented in the following section is limited to determining whether this correlation between topic labeling performance and knowledge acquisition exists, how topic models differ with respect to this correlation and how reliable the methodology is. I will not focus on how a learner makes use of the outcome of the topic labeling task analysis, e.g. to change his or her learning strategy. It is beyond the scope of this chapter, but an interesting direction for future research.

6 Evaluation setup

The primary goals of the evaluation of TOMOFF are to determine whether (1) it is indeed the case that a learner's ability to correctly interpret a topic model correlates with the level at which the learner has mastered the material objectively, (2) whether the use of a topic model that takes individual feedback into account improves this correlation, and (3) whether the topic modeling techniques used in TOMOFF also perform satisfactorily with small corpora.

Course	Type	Students
The Social Semantic Web	graduate course	≈20
Introduction to computational linguistics[13]	undergraduate course	≈30

Table 6.5: University courses in which TOMOFF has been evaluated in the 2012 and 2013.

The evaluation has taken place within two university courses taught at Utrecht University over a period of 2 years. This evaluation context is helpful for two reasons. First, the official mandatory exams that the students have to take as part of the course represent the 'objective level of knowledge acquisition' with which the results of TOMOFF can be compared with, i.e. the exam performance allows us to correlate features of the system with actual learning outcomes. Second, the students provide feedback on the papers that they read during the course which allows one to train a supervised topic model on the learning corpus and feedback.

Section 6.1 reviews the different corpora that have been used to construct the various topic models. Section 6.2 provides an overview of which topic models have been evaluated and how they've been built. Subsequently, in section 6.3 the overall statistics of the feedback that the students of the courses have provided is presented. Finally, section 6.4 describes the overall setup how the topic labeling task was administered.

6.1 Courses & Learning corpora

As part of the evaluation, we wish to determine the impact of incorporating learner feedback into the topic modeling process. This requires the involvement of actual learners. For this reason we have performed an evaluation that was embedded in two university courses. Each of these courses, listed in table 6.5, has been run for two years yielding four cohorts of students in total.

A learning corpus has been gathered for each cohort that consists of the mandatory papers the students have to read. A single paper in the undergraduate course was written in Dutch. All the other papers in both the graduate and undergraduate courses were written in English. In both courses, students read approximately between 16 and 23 papers in total which, as preliminary experiments suggest, is not enough to result in a proper data set for topic models and would make additional data analysis difficult. In order to address this, we consider each section in a paper as a separate *document* and have therefore split each paper via its section boundaries. The text in the documents (sections) in the corpus is tokenized on whitespace and lemmatized using the Wordnet-based lemmatizer in the NLTK-toolkit (Loper and Bird, 2002). The resulting documents are stored as a set of plain text files. We thus end up with four course-specific corpora each of which consists of a set of documents in the form of plain text files each corresponding to an individual section for the papers.

Finally, there is also an additional generic corpus that consists of the full text of the English Wikipedia. This generic corpus is used for comparative purposes in the evaluation of the topic

[13]The actual Dutch name of the course at Utrecht University in 2012 and 2013 was "Taal en Computer"

Name	year	papers	sections/documents	terms	unique terms
Undergraduate	2012	16	164	41218	7068
Undergraduate	2013	16	147	46894	7239
Graduate	2012	23	224	59449	6631
Graduate	2013	23	219	79699	8142
Wikipedia			3430629	1192829523	100000[14]

Table 6.6: Overall corpora statistics.

Shorthand	Description
`wiki`	LDA model of English Wikipedia
`lda`	LDA model of a learning corpus
`medlda course`	MedLDA model of a learning corpus and all ratings
`medlda user`	MedLDA model of a learning corpus and the ratings for each student

Table 6.7: Overview of different LDA-based models and their abbreviations.

models. More specifically, it is employed to confirm that the use of a learning corpus offers compelling advantages over a large generic corpus, i.e. that the generic corpus is less useful in a formative assessment than a course-specific learning corpus. A comparison between these generic and course-specific corpora is presented in section 7. Table 6.6 shows the overall statistics of each corpus and the number of terms and unique terms after stemming.

6.2 Topic models

In total, four different types of topic models have been trained on the five corpora previously introduced. All the topic models and their abbreviations are listed in table 6.7. Each topic model will be referred to using its shorthand, as listed in the first column of table 6.7.

The topic modeling algorithms have been used to construct topic models both for the course-specific learning corpora and for the generic Wikipedia corpus. The Wikipedia corpus is included in order to determine the impact of a generic corpus when compared to a domain-specific learning corpus. Terms which occur in over 90% of the documents have been excluded. I will now provide some more details as to how each corpus is constructed, the algorithms and the software that have been used to construct the topic models.

Wikipedia topic model The `wiki` topic model is trained using the online LDA algorithm, as implemented in the Gensim toolkit (Řehůřek and Sojka, 2010). Gensim is a Python framework designed for computational distributional semantics. "Gensim aims at processing raw,

[14]The number of term was limited to the 100000 most frequent terms due to memory constraints at the time.

unstructured digital texts ('plain text'). The algorithms in Gensim, such as Latent Semantic Analysis, Latent Dirichlet Allocation or Random Projections, discover semantic structure of documents, by examining word statistical co-occurrence patterns within a corpus of training documents."[15] An XML-dump of the full English Wikipedia[16] is used as the corpus. Wikipedia is a freely available resource with a large amount of content without a specific dominant domain bias. The fact that the Wikipedia corpus itself has extensive domain coverage makes it suitable to train a *generic* topic model.

Terms which occur in more than 40%[17] of the documents are discarded and each token is required to occur in at least 20 documents (Wikipedia articles). The dictionary was set to only include the first 100.000 most frequent terms. The topic model itself was trained using the online LDA algorithm (Hoffman et al., 2010) with 1000 topics and 100.000 documents for each iteration while using the default settings of the Gensim topic modeling toolkit for the other parameters.

The Wikipedia topic model was trained with 1000 topics in order to gain a sufficient level of detail for the individual topics. The number of topics is inherently arbitrary as far as the topic modeling itself is concerned and depends entirely on the end-user application. There are, however, two reasons why I chose 1000 topics instead of a higher or lower value. The first reason is that an increased number of topics results in each topic being more specific. Given the large Wikipedia corpus of diverse subjects the additional specificity of the topic model was required in order to achieve detailed feedback. A list of specific topics is assumed to be more useful to learners than a list of highly generic ones. The second reason is that 1000 topics, given the large vocabulary, was the largest number of topics that I could train realistically with the computer hardware that was available at the time.

The number of documents to use for each iteration of online LDA is also somewhat arbitrary, but the online LDA algorithm will perform poorly if there is a lot of topic drift, i.e. initial documents should not cover significantly different subjects than subsequent documents. Increasing the number of documents used for each iteration will result in a larger sampling of possible subjects and thus reduce the amount of topic drift in each iteration. The choice for 100.000 documents was also the largest amount of documents possible given the computational resources available at the time.

LDA topic model The regular `lda` topic model was constructed using the same procedure as the `wiki` model, but applied to the learning corpus. As a consequence, the restriction that a term had to occur in at least 20 documents was removed due to the limited size of the learning corpora. Second, only 20 topics[18] were inferred, because a highly specialized learning corpus was used which does not suffer from some of the problems that a large generic corpus such as

[15] `http://radimrehurek.com/gensim/intro.html` retrieved 19-01-2012

[16] The XML Wikipedia dump of article content dated 2011-10-07

[17] The default value for constructing a dictionary from Wikipedia in Gensim version 0.8.1

[18] 20 topics were inferred from the learning corpora as the topics seemed to be appropriate in the context of the course. A perplexity-based evaluation to justify this exact number of topics was not performed, because the number of topics is inherently tied to the manual topic labeling task, presented in section 5.4. To my knowledge, there is no research that suggests that an improvement in the fit of a model, e.g. perplexity, results in 'better' topics. Perplexity is a measure of how well a probabilistic model predicts on a novel test sample. Better prediction, in this context, means improved inter-annotator agreement of the proper label for a topic.

Wikipedia has. More specifically, recall that the choice for the inference of 1000 topics with the `wiki` model was due to the required amount of specificity of the topics and the generality of the corpus on which it was trained, i.e. the English Wikipedia. The `lda` model is trained on just the learning corpus and as a result does not require a large amount of topics in order to achieve the required amount of specificity. Finally, the entire corpus was processed in the first batch of the Online LDA (Blei and McAuliffe, 2010) algorithm available in Gensim (Řehůřek and Sojka, 2010) which basically makes the results the same as those obtained with regular LDA (Blei et al., 2003). As a result, topic drift in the learning corpus is not an issue.

MedLDA topic model The `medlda course` and `medlda user` topic models have been trained using the regression variant of MedLDA (Zhu et al., 2009). The number of topics was also set to 20 in order to make it comparable to the LDA-model. The use of the difficulty ratings in the topic modeling methodology can be used to personalize a topic model. More specifically, the full dataset of all ratings has been restricted in two ways: (1) the complete data of a cohort and (2) the data of a single student. In each case, the text corpus is exactly the same within a given cohort, but the ratings that are used to train the model are not. Each of these rating selections results in a different MedLDA model, because different sets of ratings are used (assuming substantial differences among students). There are thus two types of MedLDA models generated that allow us to compare the effects of personalization on different levels, i.e. the overall group (`medlda course`) and individual learners (`medlda user`)

The `medlda user` topic models are of primary interest, but because they are each unique they cannot be studied in detail statistically. We expect that the ratings of an individual student are more consistent with respect to the topic-rating correlation, but the `medlda user` models also suffer from problems related to sparsity. The `medlda course` and `medlda user` models provide evidence for the impact of increased personalization, because they incorporate the ratings in the topic model. The LDA-based models (`wiki`, `lda`) do not support taking the ratings into account, because they are constructed using the Online LDA algorithm (Hoffman et al., 2010), which is an unsupervised algorithm. The LDA and MedLDA-based topic models can be ordered by their amount of personalization: `lda`, `medlda course`, `medlda user`. We expect that the increased personalization of the MedLDA topic models has an impact on the relationship between their structure, how learners interact with them and how they relate to actual learning outcomes. These aspects are covered in more detail in section 7.

This concludes the discussion of the computational aspects of TOMOFF in the context of the learning corpora and topic modeling algorithms. The following sections present the relevant information related to the student tasks carried out during the course. More specifically, the rating of documents (section 6.3) and the topic labeling task (section 6.4).

6.3 Feedback

The first task that the students have been required to perform as part of the experiment was to submit a rating for every document read during the course. Recall that the documents in the corpus represent sections of papers. Each student was instructed to provide a rating for every

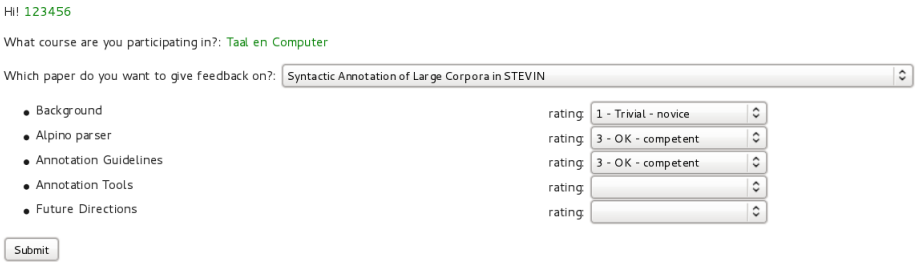


Figure 6.2: The rating web application used by the students to rate sections of papers.

Course	Year	students	ratings	avg rating	sd	avg. sd/document
Undergraduate	2012	31	1920	2.73	0.97	0.82
Undergraduate	2013	29	2290	3.02	0.95	0.74
Graduate	2012	13	2241	2.34	0.89	0.79
Graduate	2013	19	2569	2.82	1.14	1.05

Table 6.8: Descriptive rating statistics for each cohort.

document they read[19]. The topic modeling methodology, which was covered in section 5.3, treats each section as a separate document. Typically, students were required to read between 1 and 3 papers each week on a given theme. They submitted a rating between 1 and 5 that represents the subjective difficulty level of each document (see section 5.2 for details).

The students were assisted by a simple custom web-based rating system shown in fig. 6.2. The bottom of the feedback application lists an extended description of every rating option in case the student wants to consult them (not included in fig. 6.2). The ratings have been stored in a database along with an anonymous student identifier, course, time stamp and document identifier.

Table 6.8 displays the overall statistics of the ratings provided by the students. The average standard deviation per document has also been included in order to quantify the amount of variability between student ratings in each cohort. The average standard deviation per document is considerably higher in the graduate course (0.94) than in the undergraduate course (0.78) when taking the weighted average over the two years. This indicates increased disagreement on the appropriate rating for a document. Please keep this observation in mind when reviewing the results of the MedLDA topic models in section 7.

The amount of students that actually used the system to provide ratings throughout the course remained fairly constant. This is in part due to the instruction that their feedback would be

[19]The 'novelty of information' and 'writing style' were also gathered in 2013 in addition to the 'difficulty of information', because the results from 2012 alone were inconclusive. They were meant to provide additional information about the ratings in order to aid analysis. However, the statistical analysis of the results from 2012 and 2013 combined has revealed clear patterns and as a result the additional information about the novelty and writing style is not included in the results, in section 7.

used for further discussion on the topics addressed in the course. Results suggest that the students did not simply enter random ratings (junk ratings). I will present statistical evidence for this claim in section 7. However, the amount of ratings submitted is considerably lower than expected. An optimistic estimate, given the $\approx$ 10 sections in each paper, yields 10 * 20 = 200 ratings for each student and 200 * 50 students = 10000 ratings in total. However, one should expect a drop-out rate of 35% which should result in about 6500 ratings in total for both the graduate and undergraduate course combined in each year. However, we received only 4489 ratings in 2012 and 4810 ratings in 2013. The number of ratings is lower than predicted due to overall poor motivation and the fact that their course participation grade only contributed a little to their overall grade for the undergraduates and not at all for the graduate students. It is still the case that the overall majority of the students made use of the system during the course. Although it is disappointing that not every student made full use of the system, from a statistical and experimental perspective, it does make the whole evaluation more realistic and representative of real-world interactions.

6.4 Topic labeling

The second task that the students performed was a topic labeling task. The task requires the students to come up with a new topic label on the basis of the 10 most salient words of a topic. The topic labeling task was carried out by the students near the end of the course in preparation for the final exam. The labels provided by the students are automatically analyzed and a *topic label quality score* is calculated that quantifies the appropriateness of each label.

The topics in a topic model constitute an abstraction of the learning material. They also allow for making connections among the different concepts which have been acquired during the course, because a single topic spans a number of concepts. Four different types of topic models have been presented to university students, in order to evaluate their usefulness. First, we trained an LDA topic model on a generic corpus (`wiki`). Second, several LDA-based models have been trained on course specific learning corpora (`lda`). Third and fourth, two variants of a supervised topic modeling algorithm have been trained on learning corpora and the associated student ratings (`medlda course` & `medlda user`). Only the MedLDA models make use of the feedback (ratings) in the topic modeling process; the unsupervised LDA-based models can not.

The students provide a label for each of the four types of topic models (see tables 6.9 to 6.11 for examples), as previously explained. Junk topics that mostly consisted of Dutch determiners have been manually removed from the undergraduate topic models. The junk topics appeared, because only a single paper was in Dutch, whereas all the other papers were in English. Regular Topic models, because they are knowledge poor methods, cannot handle this particular situation gracefully and should therefore only be applied to monolingual corpora. Students were allowed refrain from providing a topic label if they were not able to quickly come up with an appropriate one. Therefore, one method of identifying poor performance on the task simply uses the number of absent topic labels for a topic model. Students' performance in topic labeling is expected to be indicative of knowledge acquisition and therefore to correlate with their overall exam performance.

For each of the topic models the students were asked to perform two tasks:

topic	salient words
1	edition, volume, translation, published, text, author, written, revision, manuscript, translated
2	republican, campaign, bill, senate, election, democratic, democrat, representative, senator, incumbent
3	question, answer, preceding, jack, thing, don, correct, asked, guess, give
4	product, market, marketing, customer, consumer, sale, norton, strategy, business, industry
5	feedback, output, input, rc, gain, threshold, pulse, trigger, balanced, quad
6	event, festival, annual, venue, participant, concert, host, hosted, live, competition
7	medium, web, online, blog, internet, content, facebook, forum, twitter, reliable
8	news, pm, reporter, coverage, morning, cnn, story, anchor, cbs, sport
9	software, application, network, data, process, technology, user, communication, component, interface
10	experience, relationship, positive, feel, sense, feeling, role, focus, character, audience
11	ship, submarine, patrol, navy, vessel, built, torpedo, launched, destroyer, class
12	brand, beer, drink, bottle, alcohol, brewery, drinking, beverage, ale, brewing
13	guitar, custom, gibson, thomson, dodge, neck, pickup, instrument, fury, string
14	section, dispute, mediation, issue, committee, resolution, additional, case, agree, involved
15	sailor, tin, sander, canens, kia, neon, reno, moonlight, be, moon
16	nick, cooper, rod, prop, fortune, hercules, holloway, leicester, camden, strongest
17	ring, module, vernon, comfort, ideal, drift, lookout, jacobson, cato, abram
18	theory, effect, difference, concept, process, term, idea, study, experiment, level
19	management, professional, business, development, training, program, skill, process, knowledge, level
20	data, table, database, test, statistic, analysis, error, sample, testing, method

Table 6.9: Example of a `wiki` topic model as it was presented to the students for the undergraduate course in 2013.

topic	salient words
1	corpus, annotation, tool, parse, alpino, syntactic, parser, par, number, lexical
2	expression, regular, operator, string, search, match, pattern, text, parenthesis, eliza
3	topic, corpus, term, culture, emotion, translation, text, language, love, unique
4	annotation, semantic, label, dependency, role, syntactic, al, approach, tree, based
5	verb, flight, phrase, noun, np, grammar, rule, vp, sentence, structure
6	element, attribute, type, declaration, entity, character, document, content, xml, tag
7	corpus, word, language, text, web, english, frequency, information, time, al
8	entity, xml, reference, character, document, declaration, processor, external, text, definition
9	conversion, sentence, format, xml, coi, word, text, file, processing, paragraph
10	question, bouma, answering, syntactic, automatically, information, dutch, der, clef, van
11	entity, task, ne, relation, annotation, muc, system, named, type, te
12	corpus, language, web, word, text, data, sample, type, large, kind
13	corpus, language, text, type, meaning, word, fact, study, pattern, model
14	expression, corpus, text, annotation, type, previous, match, character, regular, xml
15	match, expression, character, regular, pattern, string, word, woodchuck, digit, previous
16	keywords, keyword, language, learning, extractor, document, performance, system, result, test
17	corpus, annotation, data, word, project, dutch, transcription, treebanks, mw, cgn
18	word, transcription, annotation, orthographic, speech, student, prosodic, segmentation, corpus, protocol

Table 6.10: Example of an `lda` topic model as it was presented to the students for the undergraduate course.

topic	salient words
1	annotation, layer, dutch, semantic, reference, sonar, corpus, al, relation, temporal
2	verb, flight, np, noun, phrase, vp, rule, grammar, sentence, nominal
3	word, transcription, system, annotation, information, text, language, speech, user, corpus
4	entity, type, tag, name, ne, tagger, corpus, token, text, tagging
5	language, text, work, corpus, early, word, study, great, tradition, type
6	match, expression, occurrence, xn, char, regular, gb, character, string, number
7	annotation, syntactic, alpino, corpus, word, tool, parse, coi, parser, cgn
8	annotation, label, dependency, node, tree, corpus, semantic, role, argument, syntactic
9	focus, consistency, reference, english, discussion, section, space, strongly, aspect, structure
10	corpus, language, text, mw, translation, written, type, spoken, large, project
11	match, pattern, character, expression, string, regular, operator, woodchuck, word, line
12	xml, word, data, annotation, corpus, error, sentence, transcription, conversion, text
13	entity, type, tag, name, ne, tagger, corpus, token, text, tagging
14	entity, declaration, element, attribute, character, xml, document, reference, type, content
15	keywords, keyword, language, extractor, task, result, test, learning, document, evaluation
16	expression, regular, search, string, text, pattern, user, eliza, substitution, match
17	system, ontology, web, learning, instance, based, al, metadata, management, data
18	corpus, language, sample, time, text, data, spoken, design, english, word
19	word, phrase, grammar, noun, fly, september, seventeenth, atlanta, denver, sentence

Table 6.11: Example of a `medlda course` topic model as it was presented to the students for the undergraduate course in 2013.

1) Grade each topic model as a whole on a scale of 1 (very poor) to 10 (excellent). This scale mirrors the grading system in the Netherlands for school work. The question given to the students was "How well does each topic list represent the subjects covered during the course?".

2) Label each topic using a maximum of three words. The students were instructed to write down a label that represents the overall meaning of the topic. They were allowed to use any word they wanted, which includes words not part of the topic model. No further restrictions were placed on which words they could use as part of their label. Students were also explicitly allowed to refrain from providing topic labels for topics for which they failed to see a theme or failed to properly express the theme in words. These labels were either empty or marked with a '?'.

The first task informs us of the overall quality of the topic model. More specifically, we evaluate up to what extent the students believe that the topic model represents a complete summary of the subjects covered as part of the course. Topic models that are incomplete, or too generic, such as the Wikipedia topic model, are expected to get a lower grade compared to topic models that are more specific to the course. This is indeed what we have observed as part of the evaluation results presented in section 7.

The second task forces the students to make their interpretation of a topic explicit. As previously explained, I hypothesize that the ability to correctly interpret the individual topics of a topic model correlates with the actual level of knowledge acquisition. By having the students

topic	salient words
1	corpus,linguistics,web,computational,language,technology,issue,great,representative,practical
2	phrase, tree, treebanks, dependency, label, annotation, sentence, node, treebank, type
3	web, semantic, ontology, agent, page, information, user, service, social, data
4	language, annotator, human, score, keywords, agreement, system, result, text, inter
5	corpus, type, text, language, annotation, speech, word, event, information, tag
6	annotation, layer, dutch, al, semantic, sonar, scheme, temporal, amsterdam, time
7	transcription, word, orthographic, speech, phonetic, automatic, phoneme, dutch, layer, cgn
8	keywords, keyword, learning, language, extractor, word, tool, system, document, performance
9	entity, task, muc, ne, relation, te, st, type, ace, tr
10	translation, model, word, sentence, corpus, source, text, language, alignment, target
11	error, false, word, negative, za, ^a, positive, instance, pattern, tt
12	annotation, cgn, corpus, tool, alpino, word, evaluation, project, number, annotator
13	topic, term, culture, emotion, love, table, unique, india, medium, usa
14	word, web, language, corpus, page, english, text, translation, altavista, data
15	search, text, regular, expression, language, web, type, information, name, engine
16	verb, flight, np, noun, phrase, grammar, rule, vp, sentence, nominal
17	system, language, based, transfer, word, text, representation, approach, syntactic, rule
18	corpus, language, text, sample, data, spoken, kind, time, variety, english
19	expression, match, pattern, regular, character, string, word, operator, digit, woodchuck

Table 6.12: Example of a `medlda user` topic model as it was presented to the students for the undergraduate course in 2013.

generate a label for each of the topics, the label quality score (introduced in section 5.4) can be calculated and compared with the actual exam score. The exam score and the topic label quality score then allow one to check the hypothesis that the two measures correlate significantly. The inclusion of several topic models in the topic labeling process is only required in this evaluation of TOMOFF. In the actual roll-out of TOMOFF, learners would only have to label a single topic model. However, in the context of the evaluation several topic models have been used in order to determine which one is best suited to serve as part of formative feedback task and additionally to determine whether the use of supervised topic models offers an advantage over unsupervised topic models.

The topic labels that the students provided were automatically lemmatized using NLTK's WordNet-based lemmatization algorithm (Loper and Bird, 2002). Spelling corrections were manually performed for all labels. Acronyms provided by students were also retained, but abbreviations were expanded to their full form. For example, the topic label 'mt' for a topic about machine translation was not expanded into 'machine translation', but 'comp. ling.' was. Some students also inconsistently switched between Dutch and English topic labels which was not explicitly disallowed in the task description of 2012. For example, using the Dutch 'corpusannotatie' instead of the English "corpus annotation". Dutch topic labels have therefore been manually translated into English. This switch between Dutch and English is likely evidence for the ongoing acquisition of domain specific terminology which is to be expected in a course setting.

The results obtained through the rating and topic labeling tasks have been linked to the grade

of the mandatory written tests for each course. The written exams consisted of open-ended questions which were graded individually by a teacher. In the analysis in this chapter, only the total average exam grade is taken into account for each of the written tests. Group projects have been explicitly excluded from the dataset in order to make sure that only the individual performance of students is measured.

For each student, we aimed to elicit information on four types of topic models (see table 6.7 for details). However, it is important to note that due to an experimental error, this has not been fully accomplished in the actual evaluation. More specifically, the `wikipedia` and `lda` models were absent from the topic labeling task for the undergraduate course in 2012. All other topics models have been fully evaluated for the remaining cohorts. This experimental error affects the statistical analysis of the results and is taken into account in the analysis in section 7.

In summary, several topic models have been trained as part of the evaluation with and without the additional learner feedback, i.e. ratings gathered about the learning corpus. The topic labeling task and overall topic model grades gathered as part of the topic labeling task allows us to check the hypothesis that the actual learning outcomes correlate with a learner's ability to interpret the topic model successfully and whether different topic models vary significantly with respect to this correlation.

This concludes the discussion of the overall setup of the experiment. The following section presents the main results and their interpretation.

7 Results and Analysis

This section presents the experimental results of the student evaluation described in the previous section. Recall that each student submitted ratings during the course and performed a formative assessment task, i.e. manually labeled topics, near the end of the course, as previously discussed in section 6.3. In total, four different types of topic models have been evaluated, namely `wiki`, `lda`, `medlda course` & `medlda user`. Each of these models has been measured in four ways: (1) the grade assigned by the student (2) number of topics that have not been labeled (3) number of words used to label a topic and (4) a topic label quality score (see section 5.4 for details).

The analysis of the data will show that the amount of feedback from students does not reliably correspond to actual student involvement and learning outcomes. The data also suggests that learning outcome prediction based on just the overall grade (1) is insufficient and a more detailed analysis, such as metrics (2), (3) and (4) is necessary. The analysis suggests that graduate and undergraduate students vary with respect to what metrics best correspond to their level of conceptual knowledge. Students are expected to prefer the course-specific corpora and resulting topic models over generic ones based on Wikipedia, because they are more relevant with respect to the content of the course.

Two multilevel mixed linear effects models have been fitted to the data using the various measures as fixed effects and the student id as a random factor. The graduate model is based on 44 students and the undergraduate model is based on 61 students. The data actually con-

sists of 4 cohorts, but the individual cohorts of each course, i.e. 2012 and 2013, have been merged for because of their limited sample sizes. As a result, there are two separate data sets that each span two years. As previously illustrated in table 6.8, the samples are unbalanced for the graduate course which might have skewed the results towards the graduate cohort of 2013. Both models used an unstructured covariance structure and all probabilities that are listed are two-tailed. The data for all the metrics has been normalized in the range of [0,1]. A weighted average of all grades is used, because the graduate course in 2013 used four exams, whereas the one in 2012 used two. The correlations listed in tables 6.13 to 6.17 are Pearson product-moment correlation coefficients (r) between the metric and model combination that is listed and the *weighted average exam grade*. Details about the models and the confidence intervals of various parameters are described in appendix C.

It is important to note that there is at least one non-random cause for missing data in the dataset. This is due to the experimental error previously mentioned in section 6.4. Inspection of the means and variances identified deviations on two variables (number of missing labels for `medlda course` and `medlda user`) with and without this cohort. This might have skewed the results of the subsequent statistical analysis. Means and variances for the other cohorts were not found to be statistically different.

Additionally, there is some oversampling of students that did well on the course given the fact that poor students have dropped out with a corresponding loss of data points on later exams and feedback. More specifically, 44 graduate students were enrolled in the course and 20 (45%) did not pass their individual exams, i.e. had an average grade less than 5.5 out of 10. 39 graduate students participated in the topic labeling task of which 16 (41%) did not pass their exams. 60 undergraduate students were enrolled in total, and 13 (22%) did not pass their exams. 43 undergraduate students participated in the topic labeling task of which 7 (16%) did not pass their exams.

On a methodological level this study can only report significant findings for relatively large effect sizes, because the number of participants is quite low and the experiments were performed in the (virtual) classroom, as opposed to a well-controlled laboratory. A study with larger samples is probably required for effect sizes smaller than $r = 0.3$ (Cohen, 1988, p.102). As a result, a larger study might establish additional statistically significant weaker correlations, which are currently absent. No outliers of any kind have been removed from the dataset in order to improve or boost results.

I will now present the various types of metrics that have been gathered as part of the course immediately followed by an analysis of what the results are likely to indicate. Section 7.1 gives a broad overview of the feedback that students gave on the course corpus and how it relates to their average exam grade. The subsequent sections discuss the four different metrics that are used to analyze the results of the topic labeling task. These four metrics are: grade assigned to a topic model (section 7.2), number of absent topic labels (section 7.3), number of words used to label a topic (section 7.4) and the topic label quality score (section 7.5).

7.1 Ratings

The correlation between the number of ratings a student submitted during the course and the average rating are listed in table 6.13. Naively, one might believe that students which rate a

lot of papers as difficult are likely to be correlated with poor exam grades. Similarly, students which rate most papers as easy would be expected to get good grades. However, the results in table 6.13 show no such correspondence for both the graduate and undergraduate courses. This is likely due to the fact that overestimation and underestimation, presented in section 5.4, cancel each other out in the student population as a whole.

Course	average rating	number of ratings
Graduate	-0.26	0.18
Undergraduate	-0.19	0.55***

Table 6.13: Correlations between overall rating statistics and the average exam grade.

A possible explanation for this fact is that explicit self-assessments are not so reliable, as observed by Baker (1989) and many others. More specifically, that overestimation and underestimation of one's understanding of a resource is considerably more common than people expect. As a result, the over- and underestimation in the data can cancel each other out in the statistical analysis and hence the lack of any significant correlation for the average rating with the learning outcomes.

The second part of table 6.13 concerns the total number of ratings a student submitted during the course using the online rating system. Only the undergraduate course had a strong correlation (r=0.55***), between the number of ratings and the average exam grade. This means that when a student provides a rating, it reflects the student actually reading the paper thoroughly. This result is not surprising by itself. Students which read more of the documents in the learning corpus are expected to do better on the exam, but it is good to see this confirmed nonetheless. However, interestingly enough this correlation is absent for the graduate course. A possible explanation could be that the graduate students are only skimming the text, normal behavior for advanced readers, for interesting parts or not reading them at all.

7.2 Overall grade

The first metric with respect to the topic models is the overall grade that the students assigned to a topic model as a whole. These results are shown in table 6.14. No significant correlation was found for any of the grades that the students assigned to the topic models and their average exam grade. It is, however, worthwhile to look at how the students graded each of the different topic models. More specifically, a significantly poorer grade for the Wikipedia-based model was expected in comparison to the other learning corpus-based models, i.e. `lda`, `medlda course` and `medlda user`. This is due to the assumption that the use of a course-specific corpus results in more relevant and specific topic models. The distribution of grades that students' assigned is depicted in fig. 6.3a for the graduate course and in fig. 6.3b for the undergraduate course. It is clear that the Wikipedia-based models are rejected by students since they believe this model does not capture the important topics of the respective course. Based on the overall grades of the three remaining models, no clear differences emerge with respect to the assigned grades. In conclusion, the Wikipedia-based LDA model is, as expected, rejected by students, but none of the overall grades that the students assigned

to the topic models correlate significantly with the average exam grade. This result was expected, because the overall grade is much too noisy and thus suggests that a more detailed analysis, presented in the following sections, between the topic model and a learner is in order.

Course	wiki	lda	medlda course	medlda user
Graduate	-0.14	-0.22	-0.02	0.17
Undergraduate	-0.17	0.20	-0.23	0.16

Table 6.14: Correlations between the overall grade that students assigned to a topic model and their average exam grade.

7.3 Absent topic labels

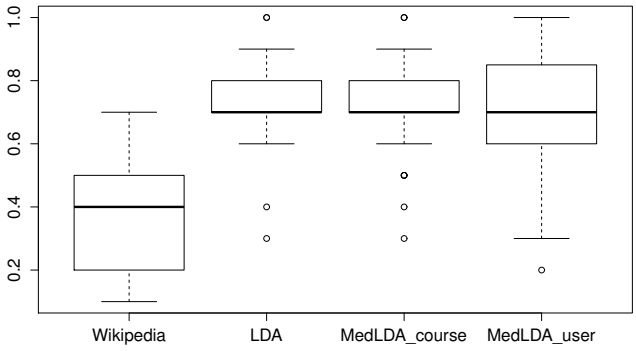
The second metric is the number of absent/missing topic labels in the topic labeling task (see section 5.4 for details). The overall results are presented in table 6.15. For the graduate course the number of absent topic labels for the `medlda course` (r=0.34*) and `medlda user` (r=0.59**) models correlates significantly with the average exam grade, i.e. a smaller number of absent topic labels correlates strongly with a higher exam grade. The undergraduate course also has a strong correlation (r=0.37**) with the exam grade on the number of absent `medlda user` model's topic labels and a significant correlation (r=0.30*) for the `lda` model. The `wiki` model's lack of any correlation is to be expected, because it is relatively poor and too general.

The inconsistency between the `lda` and `medlda course` models with respect to significant correlations is surprising. One would expect the `lda` and `medlda course` models to perform similarly with respect to absent topic labels, because there is no significant difference in how the students appreciated them, as shown in figs. 6.3a and 6.3b. However, the undergraduate course correlates significantly with the `lda` model, whereas the graduate course correlates significantly with the `medlda course` model. This suggests that the two types of models differ in a way that does not affect the quality of the topic model as a whole, but only the relationship between the topic labeling task and the conceptual knowledge of the students. It is likely that these differences between the `lda` and `medlda course` model are, in part, due to the small number of students that participated in the experiments, because the effect size is relatively small (Cohen, 1988; Schönbrodt and Perugini, 2013).

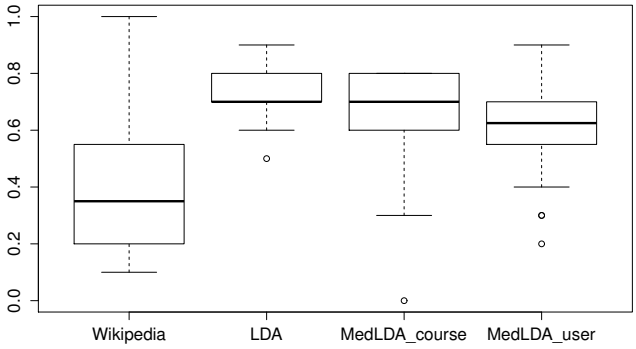
Course	wiki	lda	medlda course	medlda user
Graduate	0.23	0.31*	0.34*	0.59**
Undergraduate	0.16	0.30*	0.18	0.37**

Table 6.15: Correlations between the number of absent topic labels and a student's average exam grade.

Figures 6.4a and 6.4b plot the normalized number of absent topic labels for both the graduate



(a) Graduate



(b) Undergraduate

Figure 6.3: A boxplot of the normalized grades assigned to each of the 4 different topic models for the graduate and undergraduate course.

(fig. 6.4a) and undergraduate (fig. 6.4b) courses. A normalized score of 1.0 reflects no absent topic labels and 0.0 reflects that all topic labels are absent. The normalization is required, because the undergraduate topic model's number of topics varies, because of the post-hoc filtering of junk topics, i.e. topics that only consist of Dutch determiners due to the inclusion of one Dutch paper in an English corpus. Again, it is clear that the Wikipedia model stands out and that there are only marginal differences between the three other models with respect to the number of absent topic labels.

7.4 Number of words

The third metric is the number of words that were used by a student to label a topic. The results from the statistical analysis are presented in table 6.16. The students for the undergraduate course used ≈ 1.62 words per topic which translates into ≈ 32 words for a topic model of 20 topics[20]. The graduate students used a considerably higher amount of words per topic. They used, on average, ≈ 2.03 words per topic label which translates into ≈ 41 words for the topic model as a whole. A boxplot of the normalized amount of words used for each topic model is shown in figs. 6.5a and 6.5b. The y-axis should be limited to 1.0, i.e. the maximum number of words for a topic label[21]. However, some of the graduate students ignored the limit imposed on the length of topic labels in their feedback.

It is striking that, for the undergraduates, the number of words significantly correlates for all models, except the `wiki` model. Yet, these correlations are completely absent for the graduate course. When we contrast these results with those in table 6.15, it is surprising that we did establish reasonable correlations for missing topic labels. However, there are no correlations for the graduate course between the number of words used to label topics and their average exam grade in table 6.16. This result is unexpected, because an absent topic label, a label of 0 words, should have a negative impact on the total number of words. These results suggest that when an undergraduate student labels a topic, it is likely to correspond to actual domain knowledge irrespective of what the actual topic label is. This correlation does not hold for the graduates. This suggests that they attempt to label a topic even if they do not have the corresponding domain knowledge. One possible interpretation is that this reflects the result from Baker (1989), i.e. experts are more likely to overestimate their knowledge than novices.

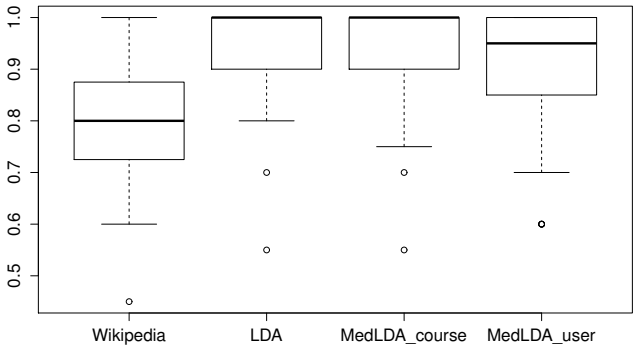
are more restrained in the words that they use when compared to graduate students.

Course	wiki	lda	medlda_course	medlda_user
Graduate	0.03	0.08	0.13	0.18
Undergraduate	0.06	0.42*	0.30*	0.46***

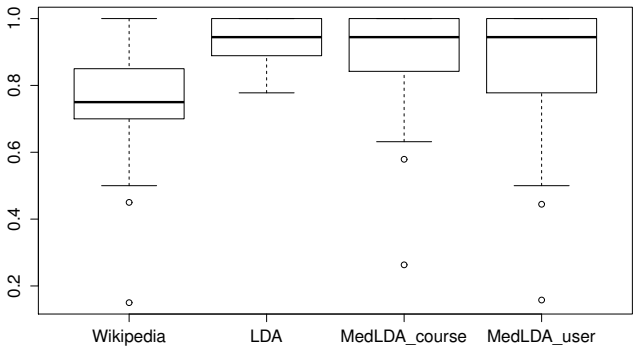
Table 6.16: Correlations between the number of words used to label a topic and a student's average exam grade.

[20] These figures do not take into account the number of words used for the `wiki` model.

[21] 3 words per topic label, which totals to 3*20=60 words for a topic model of 20 topics as a whole.

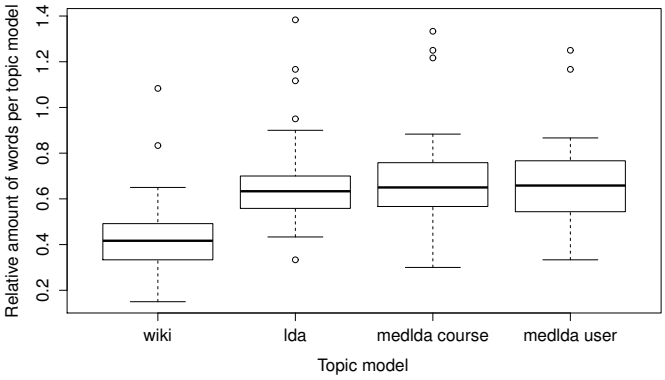


(a) Graduate

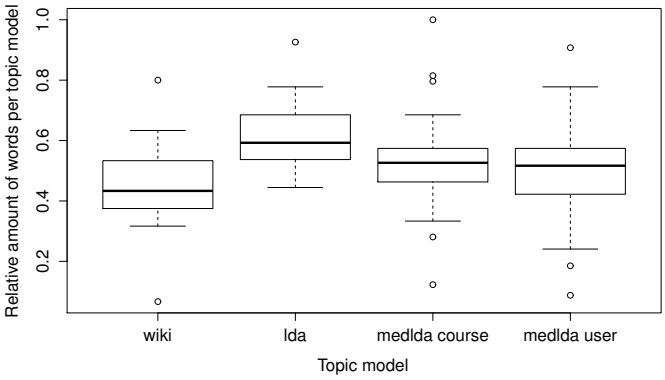


(b) Undergraduate

Figure 6.4: A boxplot of the normalized number of absent topic labels for each of the four types of topic models for the graduate and undergraduate course.



(a) Graduate



(b) Undergraduate

Figure 6.5: A boxplot of the normalized number of words used to label each of the four types of topic models for the graduate and undergraduate course.

7.5 Topic label quality

The fourth and final metric is the topic label quality score for each of the four types of topic models. This metric is automatically calculated and is based on the descriptions that the students have assigned to the individual topics of a model. The results from the linear mixed effects analysis are shown in table 6.17. Again, as expected, the Wikipedia-based model does not correlate significantly with the average exam grade for both the graduate and undergraduate courses. However, the `lda` model does correlate significantly for the undergraduate (r=0.49*) course. This is a clear improvement over the number of absent topic labels in table 6.15 for the `lda` model. This suggests that the topic label quality uncovers additional information based on the quality of a topic label that is different from the number of absent topic labels in section 7.3.

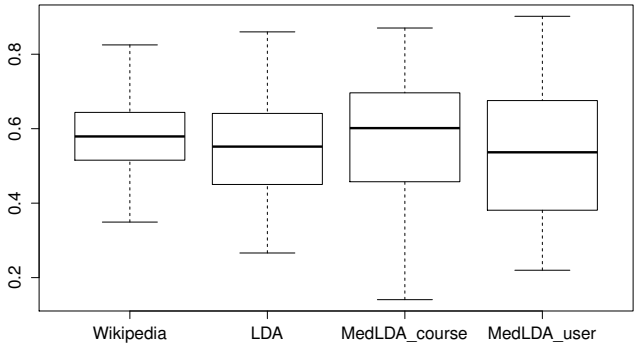
Course	wiki	lda	medlda course	medlda user
Graduate	-0.18	0.26	0.37*	0.36*
Undergraduate	-0.13	0.49**	0.17	0.47**

Table 6.17: Correlations between the topic label quality score and a student's average exam grade.

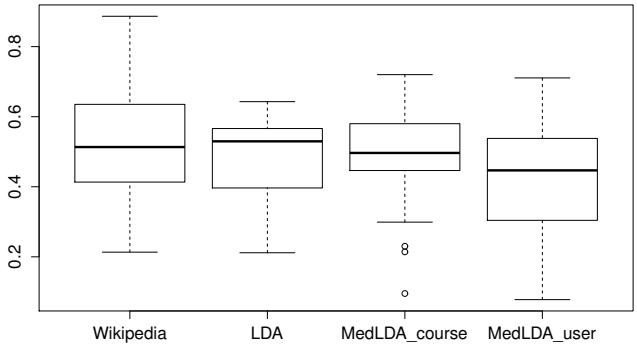
The `medlda course` only correlates significantly for the graduate course (r=0.37*). This mirrors the results from the absent topic labels analysis in table 6.15 and it is likely that this is in some part due to the same underlying reason, because, as expected, the two are moderately correlated (r=0.37*). Finally, the `medlda user` model continues to perform well with significant correlations for both the graduate (r=0.36*) and undergraduate (r=0.47**) courses. However, the correlations are considerably less than those obtained through the absent topic label metric in table 6.15.

Figures 6.6a and 6.6b depict boxplots of the average topic label quality scores for each of the four types of topic models for the graduate (fig. 6.6a) and undergraduate (fig. 6.6b) courses. Again, as previously in figs. 6.4a and 6.4b, the variance for the `medlda user` model is greatest which is the expected result due to the effects of personalization.

Conclusion In summary, student performance on the topic labeling task strongly correlates with exam performance when measured using the topic label quality. The results of the analysis suggest that the outcomes are not specific to a particular course or learning corpus. This is due to the fact that the data set has been obtained under a number of different settings, i.e. two different courses, four cohorts of students over a timespan of two years and four different small learning corpora. Additionally, the topic model for each cohort is also unique due to differences in the learning corpora. This suggests that the topic labeling task and associated evaluation methodology is a suitable method of measuring the level knowledge acquisition that generalizes to other courses and (small) learning corpora. The students in the evaluation assessed the quality of the resulting topic models as excellent. They did not blindly assign high grades, because considerably lower grades were assigned to the less appropriate Wikipedia-based model (figs. 6.3a and 6.3b).



(a) Graduate



(b) Undergraduate

Figure 6.6: A boxplot of the average topic label quality score for each of the four types of topic models for the graduate and undergraduate course.

The ability of a student to interpret a topic model is strongly correlated to the acquired domain knowledge taught as part of a (university) course, as evidenced in tables 6.15 and 6.17. The number of times a student has submitted feedback strongly correlates with their exam grade for the undergraduate course suggesting that the ratings provided during that course are more representative of students actually studying the material than during the graduate course. No significant correlation has been established between the grades that students assigned to the topic models and their exam grades, but this was an expected result. However, students did identify, as expected, the Wikipedia-based topic model as a poor summary of the subjects taught during the course (figs. 6.3a and 6.3b).

The most important results with respect to the evaluation of TOMOFF are those presented in tables 6.15 to 6.17. They indicate strong correlations between the ability to interpret a topic and actual learning outcomes as measured through the exams. This was measured using either the number of absent topic labels or a quality measure for the labels that the students provided. Both result in strong correlations and the added advantage of the topic label quality measure becomes especially clear when one compares the results in table 6.15 and table 6.17 with respect to the `lda` topic model.

8 Conclusion

Topic modeling is traditionally used to model the content of a corpus, but ignores individual differences in how documents are interpreted, i.e. the relevant topics in the exact same corpus are likely to be different from learner to learner. TOMOFF exploits this fact and has shown that the ability to correctly interpret a topic model is indicative of knowledge acquisition. In result, TOMOFF integrates the macro-perspective of corpus analysis with the micro-perspective of personal knowledge assessment.

The experimental results show that a student's average exam grade, i.e. learning outcome, correlates well with their performance on a topic labeling task as measured through either the number of absent labels or the average topic label quality score. There are small differences between graduate and undergraduate students and the underlying reasons for that might be interesting to explore in future work. Additionally, the evaluation shows that small corpora can yield topic models that are of sufficient semantic intelligibility and that they are to be preferred over topic models trained on large, generic corpora, such as Wikipedia.

Detailed analysis of the various topic models shows that the use of ratings in combination with supervised topic models can increase performance, e.g. as can be observed with the `medlda course` and `medlda user` models in tables 6.15 and 6.17. The results suggest that personalization becomes more important when there is increased disagreement between learners on the difficulty of learning objects, as shown in table 6.8. More specifically, the `medlda course` model outperforms the `lda` model for the graduate course in tables 6.15 and 6.17, but this does not apply to the undergraduate course which has less disagreement between ratings, as evidenced by the results in table 6.8. Additionally, the added value of the topic label quality score over the number of absent topic labels for the `lda` model is considerable when one compares tables 6.15 and 6.17.

These observations warrants additional research, because the impact of supervised topic mod-

els, i.e. MedLDA, on the resulting model's semantics and subsequently a learner's interpretation process is unclear. The TOMOFF approach works remarkably well considering the amount of information elicited from the students, i.e. eliciting about 32 to 41 words, on average per student, achieved a correlation with the exam grade of up to 0.59**. It is relevant to note, that even though the required number of words is small, the effort required to generate them is relatively large. The TOMOFF approach is much more feasible than many other current approaches based on topic models, such as essay writing. The minimalistic task generated by TOMOFF is domain independent, user friendly and easy to carry out in today's time constrained learning environments.

Chapter 7

Conclusion & Discussion

1 Overall summary

Social constructivism suggests that learning is an inherently social phenomenon. The approaches presented in this dissertation emphasize the social nature of information, knowledge and language, using a combination of ontologies, reference repositories, folksonomy analysis and topic modeling. Differences between the vocabulary of an individual and that of a Community of Practice present both a problem and an opportunity. The problem is that it can impede access to appropriate resources for learning, e.g. keyword-based search can be very sensitive to subtle lexical differences. I have shown that poor alignment of an individual's vocabulary with that of a community can be addressed using enriched ontologies in combination with semantic search. The opportunity is that analysis of the vocabulary and 'lexical competence' of an individual reveals his or her conceptual knowledge and allows computer algorithms to assist individuals with reflection and critical thinking. Uptake of the proper community vocabulary is a good predictor of an individual's integration with a community, because it signifies the acceptance and acquisition of the community's conceptualization and vocabulary (Danescu-Niculescu-Mizil et al., 2013). The tight integration between conceptual knowledge and vocabulary usage is useful in the context of assessment and personalization in order to tailor resources and individuals to the appropriate level of knowledge by means of topic models.

2 Contributions

This section highlights the primary technical and methodological contributions of this dissertation. More specifically, this concerns contributions in the area of ontology enrichment, disambiguation, semantic search, topic models and e-learning.

Social Ontology enrichment Static domain ontologies, which become disconnected from their respective community, are reconnected by means of a novel methodology that I have called *Social Ontology Enrichment* (SOE), by adding relevant concepts and terms such that they reflect the knowledge that is embodied by a specific Community of Practice. SOE contributes to the integration of two complementary aspects of knowledge representation and collaborative learning systems: domain ontologies in the context of the Semantic Web and folksonomies as formalizations of Social Media. The fact that tags, and by extension reference concepts, are deeply embedded in the social structure of a folksonomy allows one to determine their relevancy from the perspective of an online knowledge community using a similarity measure, more specifically resource coocurrence with Jaccard-based normalization. This is important because a domain ontology that is out of sync with its respective community is incapable of mediating between a novice learner and the content that is produced by the relevant online Community of Practice (Wenger and Snyder, 2000).

Other approaches attempt to develop (light) ontologies from sets of tags (Mika, 2005; Schmitz, 2006; Heymann and Garcia-Molina, 2006; Tang et al., 2009). The SOE approach is complementary, because it relies on existing domain ontologies and uses external data to enrich them. SOE does not make use of other domain ontologies for enrichment such as in Specia and Motta (2007); Angeletou (2010), but instead relies on reference repositories such as DBpedia (Bizer et al., 2009b). These reference repositories suffer from fewer coverage issues (Lin et al., 2009) than for example WordNet (Miller, 1995; Vossen, 2002), i.e. more terms can be resolved to concepts. SOE also takes the lexical enrichment of domain ontologies into account, because these are crucial for end user applications (Monachesi et al., 2008, 2009).

In an abstract sense, the proposed ontology enrichment methodology extends the tripartite model from Mika (2005) specifically with regard to concepts and their lexicalizations. Whereas Mika (2005) conflated tags and concepts, SOE links tags to corresponding reference concepts which results in an explicit integration of a folksonomy with formal ontologies and reference repositories. SOE is designed to apply ontology enrichment entirely automatically in contrast to Alves and Santanche (2011, 2012).

SOE demonstrates the value of ontology-based crawling (Luong et al., 2009) in order to construct a seeded corpus. The use of a seeded corpus allows SOE to target a specific Community of Practice (Wenger and Snyder, 2000) by means of their use of characteristic unambiguous terms, as provided by a domain ontology. The lexical layer of an ontology in combination with reference repositories can be used to automatically extract domain-specific information from a folksonomy in spite of the presence of multiple communities with overlapping vocabularies. SOE achieves a balance between the structure of the original domain ontology and new conceptualizations, entities and domain structure, as present in a folksonomy. The use of reference repositories and disambiguation not only establishes an explicit integration between domain ontologies and Social Media, but also with the Semantic Web at large via the use of shared identifiers for concepts and instances.

Disambiguation Ideally, one would like to represent the information available in a folksonomy on the conceptual level instead of being limited to the lexical level, i.e. a vocabulary of terms. However, concepts are not explicitly available in a folksonomy, only tags (terms) are.

With the help of reference repositories such as DBpedia (Bizer et al., 2009b) and automated disambiguation, tags can be linked to reference concepts. This allows one to distinguish between synonyms, homonyms and other relevant lexical and conceptual relationships and bridge the gap between ontologies and folksonomies.

The disambiguation algorithm in this dissertation builds on the work presented in Monachesi and Markus (2010b). It introduces a graph-based disambiguation algorithm that exploits the community structure of graphs (Girvan and Newman, 2002). It supports the unsupervised and automatic disambiguation of tags by constructing a network of concepts. The proposed disambiguation algorithm also deals with disjoint sets of concepts, such as resources annotated with terms from multiple domains or mutually ambiguous terms. The disambiguation approach exploits reference repositories the scope and size of which is much greater than traditional resources, such as WordNet. However, their scope and size does come at a cost with respect to quality and consistency (Bizer et al., 2009b; Bollacker et al., 2008). My disambiguation algorithm uses heuristics to exploit the semi-structured from reference repositories to accomplish its goal.

The disambiguation algorithm makes use of graph-clustering techniques and is strongly related to other approaches to disambiguation (Anaya-Sánchez et al., 2007; Specia and Motta, 2007; Tomuro and Shepitsen, 2009). However, an important difference is that the graphs are generated using a large sense inventory similar to Garca-Silva et al. (2009); Han and Zhao (2010). The proposed algorithm identifies clusters of concepts instead of terms or term vectors in a similar fashion as Tomuro and Shepitsen (2009); Specia and Motta (2007); Mendes et al. (2011). The major difference with respect to these related approaches is the use of links between articles and the use of community-based clustering to enforce global coherence between word senses of different terms. Degree, the number edges to or from a certain node, is used as the measure for selecting concepts within individual clusters, because it is a transparent and elegant measure. My evaluation confirms the effectiveness of degree as previously reported in Navigli and Lapata (2007). The resulting approach has similarities with that of Anaya-Sánchez et al. (2007) in the sense that both approaches share a sense clustering and cluster selection/filtering stage, but my approach is much simpler, not WordNet-specific and it does not require recursive clustering of candidate clusters. The approach is also much simpler than that of Mendes et al. (2011) through its use of standard DBpedia datasets as opposed to manual extraction of surface forms, while achieving better performance.

Although only simple features from a large noisy reference repository are used, performance is excellent, i.e. it achieves an accuracy of 0.84, which is competitive to related state of the art approaches to disambiguation. The evaluation clearly shows that it outperforms default sense heuristics based on lexical overlap such as LESK and the 'most frequent sense'-heuristic (Kilgarriff and Rosenzweig, 2000; Edmonds and Cotton, 2001; Mihalcea et al., 2004).

Semantic Search The SOSEM approach to semantic search has been shown to be effective for improving access to information in Social Media and clearly demonstrates the value of reference-repository-based disambiguation in combination with ontology enrichment. The SOSEM semantic search approach has in common with dos Reis et al. (2011) that both aim to improve resource retrieval in Social Media using ontology-supported semantic search. However, SOSEM only relies on the presence of tags instead of the document content. The added

value of exploiting tags is that non-textual resources such as images or videos can also be retrieved using the exact same methodology. The approaches from Egozi et al. (2011); Bonino et al. (2004); Castells et al. (2007); Tran et al. (2007) also operate only on the textual content of documents and do not presuppose a Social Media setting with tags.

Both SOSEM and the approach of Egozi et al. (2011) share the assumption that Wikipedia-derived concepts can be used to improve resource retrieval. However, SOSEM employs the graph structure of Wikipedia for disambiguation and term selection, and uses a domain ontology to improve resource retrieval. Egozi et al. (2011) do not employ an ontology, but use Wikipedia concepts to construct a latent semantic space based on a term's concept associations. Similarly, Wang et al. (2008) recognize the value of the lexical information of Wikipedia to bridge the gap from keywords to ontological concepts. SOSEM is supported by a domain ontology and achieves a goal which is the exact opposite of Santamaría et al. (2010), who also used Wikipedia: more specific and unambiguous search results instead of more diverse ones. Finally, ambiguity is explicitly support as opposed to Tran et al. (2007) who assume that terms in an ontology are unambiguous.

Ontology enrichment increases the performance of semantic search with respect to identifying the appropriate resources in the context of the domain ontology. The evaluation of SOSEM reaffirms the added value of ontology enrichment in the context of improved information retrieval. Both precision and recall substantially improve when the enriched ontology is employed instead of the original ontology. The increased lexical coverage and conceptual enrichment help in identifying relevant resources in tag-based Social Media. An enriched domain ontology is better able to provide structured access to domain resources from Social Media, i.e. relevant resources are identified with an average F-score of 0.83.

Topic models TOMOFF is an approach that provides "scaffolds that enhance critical thinking and reflection" (Lin et al., 1999, p.44) automatically on an abstraction level close to concept maps using modern topic modeling techniques. The methodology I propose is supported by robust topic modeling (Blei et al., 2003; Zhu et al., 2009). It allows for the generation of a highly structured task that both reduces input requirements and enforces appropriate domain coverage. The use of structured tasks for knowledge elicitation is compatible with the observation that "semantic associations are not per se the optimum in terms of word knowledge development [...] the free association task [is] less suitable as a test of word knowledge development and emphasizes the need for more structured tasks that specifically target (the recognition of) semantic meaning aspects of words" (Cremer, 2013, p.190).

TOMOFF is designed to be complementary to other techniques based on essay assessment (Wolfe et al., 1998; Lemaire and Dessus, 2001; Landauer, 2003; Hasan, 2012; Jorge-Botana et al., 2010), short answer assessment (Ziai et al., 2012) and the assessment of personal blog entries (Wild et al., 2010), because it only requires 30-40 words of input from a learner. TOMOFF's objective is not to maximize agreement with human graders using a supervised learning approach, but is instead intentionally limited to the use of only a learning corpus, i.e. without any explicit teacher or expert feedback. This allows TOMOFF to be used in today's time constrained and just-in-time (collaborative) learning environments that lack institutional support. An important difference between TOMOFF and related work on essay-grading is the fact that the trained topic model is presented to learners as opposed to only using it as

a machine learning tool in the background (Wolfe et al., 1998; Lemaire and Dessus, 2001; Landauer, 2003; Hasan, 2012; Jorge-Botana et al., 2010). Learner's ability to interpret topic models within TOMOFF significantly correlates with actual learning outcomes ($r \approx 0.48 - 0.59$) and can thus be used as a tool in support of self-assessment in lifelong informal learning.

e-learning The integration of the teacher practice of mediation between knowledge sources and peers needs to take the leap towards the dominant online environment used by students: the social networks and online search and recommendation systems (Dalsgaard, 2006; Marenzi et al., 2008). However, in this context access to knowledge is sensitive to the use of a learner's vocabulary. Additionally, knowledge available to learners might not match their current level of understanding and learning objectives.

Domain ontologies that have been integrated with folksonomies can be used by learners to navigate an unfamiliar domain and improve their access to relevant resources and improve domain understanding. The integration of the ontology with the community occurs at both the conceptual and lexical layers of an ontology. Lexical enrichment is important, because an improved alignment between the lexicon of a domain ontology and its respective community increases the ability of novices and non-experts to exploit the domain structure of ontologies. The ontology also compensates for the lack of lexical competence of a learner, because learners can navigate the domain using the conceptual structure of the ontology and acquire proper domain terms, concepts and interrelations in an efficient and structured fashion (Westerhout et al., 2010, 2011). Domain ontologies are thus effectively re-established as continuously evolving formalizations of expert communities that assist and mediate in the access to resources and domain knowledge.

This dissertation has illustrated how formative feedback can be generated using topic models in a way that supports critical thinking and reflection. This approach is based on an automatic computational analysis of the language employed in learning objects in the same spirit as Kinchin and Cabot (2010); Gog et al. (2009), but with the important difference that the feedback is automatically inferred from the learning objects and does not require the learner to invest much time in an evaluation task. Lexical analysis support by LDA-based topic models allows one to identify which specific elements of domain knowledge require additional attention and which are sufficiently acquired. Topic models represent the relevant lexico-semantic associations in a corpus of expert texts and can be used to summarize content. Personalization of the topic model via integration of learner feedback in the topic modeling process can be used to improve the correlation between a learner's ability to interpret the topic model and learning outcomes. The work on topic models has identified a clear relation between the domain knowledge of learners and their ability to interpret topic models. In brief, the TOMOFF approach makes the latent patterns in a learner's learning process explicit and available for reflection (Markus and Westerhout, 2011). My work suggests that topic models do not necessarily help learners access complex information, because their ability to interpret the topic model is inherently linked to their knowledge of the underlying corpus. I would therefore recommend that topic models should be used with restraint as aids for navigating unfamiliar information.

The technical solutions in this dissertation have shown how some of the problems related to the vocabulary problem (Furnas et al., 1987) and lack of lexical competence (Marconi,

1995) can be overcome using enriched ontologies and topic models. It is thus important for future e-learning systems to provide a similar integration with dominant user platforms in Social Media. However, adoption of these practices is notoriously slow in many academic institutions. The abundance of resources in Social Media drives learners towards fact finding and surface learning (Beattie IV et al., 1997) with little motivation towards deep learning, i.e. to perform *informational search* (Jansen et al., 2008). This makes the contributions in this dissertation all the more relevant and important. The presented technical solutions contribute to guiding learners towards deep learning through the domain model of an ontology and help moderate their learning strategies in a way that fits formal and informal, fast-paced learning environments.

In summary, language and knowledge are tightly interlinked and interdependent. Integration of the huge amount of information generated by Social Media can be used in combination with ontologies from the Semantic Web to enhance the access to resources and provide an overview of a domain. This is accomplished via ontology enrichment, disambiguation and semantic search. The vocabulary used by learners in relation to the resources used in a Community of Practice can be used for automatic assessment by means of topic models. Increased divergence between a learner's vocabulary and that of a community indicates poorer knowledge acquisition and this has implications for the design and future of learning-support systems in complex online environments, the importance of computational lexical semantics for e-learning and the interaction between institutional knowledge management, online learning and search.

3 Further research

There are plenty of opportunities to expand in new directions with regard to the subject that this dissertation covers. However, in order to complete something, one must choose one path instead of trying to follow all possible paths of interest. I have included some of these alternative paths of research which includes; topic modeling for e-learning, the role of ontologies in the Social Web and graph-based disambiguation with reference repositories.

Domain ontologies and ontology learning Tag-based Social Media have since long proven their ability to disseminate and provide access to arbitrary information. In the past, many have argued that formal domain models, i.e. ontologies, were vital for the mediation between large amounts of knowledge and users. However, the simple vocabulary and community-oriented information management of Social Media has been shown to be more flexible and popular in many respects. At this point, it is unclear what the long term role of detailed domain ontologies is going to be if folksonomies are indeed suitable for most information management tasks. The Social Ontology Enrichment (chapter 3) approach in this dissertation can transfer some of the advantages of Social Media to existing domain ontologies. This is especially relevant, because of the fact that it is possible to construct domain models from linguistic resources themselves, albeit with a lower amount of precision and or recall (see chapter 3, section 2.2).

However, it is likely that some abstract aspects of domain knowledge are rarely spelled out,

because they are assumed to be 'obvious', 'trivial' or part of a domain's dogmas. It is exactly these types of knowledge that ontologies can represent, because these aspects are unlikely to arise, due to the abstractness and implicitness, from the automated analysis of domain resources. This seems to suggest that upper level ontologies in combination with automatic ontology learning for domain ontologies will prove to be more important than automatically learning very detailed domain ontologies in the long term.

The implication for future work is that the focus of researchers should move away from the manual construction of domain ontologies and towards automatic ontology learning from various sources. Domain ontologies are clearly here to stay, due to their benefits with respect to information management, interoperability and knowledge discovery. The impact of user generated content on knowledge management is already considerable, but needs to move further forward with respect to the creation and maintenance of knowledge-rich artifacts. The work on ontology enrichment presented in this thesis is a small step in that direction during this transition phase. At present, large scale knowledge extraction that is based on collaborative processes are geared towards the creation of reference repositories such as Freebase (Bollacker et al., 2008) and DBpedia (Bizer et al., 2009b), but it still needs to make the leap towards the expression level of domain ontologies: automatically constructed domain ontologies need to improve on quality and detail, in particular with respect to the relations between concepts, instances, and their properties. This has not been achieved yet, as most approaches reviewed in chapter 3 are limited to rough taxonomies.

Dynamic Topic Models for e-learning The topic modeling approach from chapter 6 has used a corpus of documents that is spread out over time, i.e. the duration of a course. During this period it is likely that students learned new information and, as a result, their conceptual knowledge and vocabulary has changed. One would thus expect that 'difficulty ratings' assigned at the beginning of the course are less relevant than those presented more recently. In more formal terms, the relationship between topics and their average difficulty is likely to decline over time, i.e. introductory material at the beginning of the course should become trivial near the end. Preliminary experiments have been preformed with linear regression on the interaction of time and a topic with respect to prediction of the difficulty rating of a topic. However, no consistent statistically significant results have been obtained.

I expect that the recently introduced family of dynamic topic models (Wang et al., 2012) is able to capture the changes in vocabulary over time. This is relevant, because during a course, a learner makes considerable conceptual and lexical adjustments with respect to the subject matter. In the experiments presented in chapter 6, no discounting of older information was performed. The topic model composition of the course material would naturally change over time as different topics are discussed in the course, but it is currently unknown how the lexical changes over time should be accounted for in the final topic labeling task of TOMOFF, i.e. the latest information is the most reliable in representing a learner's current understanding, but is usually dependent on the proper acquisition of previous information in the course. A dynamic topic model could make this distinction between past and present information and further improve the correlation between domain understanding and topic modeling.

Personalization support to disambiguation Disambiguation is not necessarily about objectively linking a term in a context to a word sense. One can imagine that the word senses are actually predetermined by the social context in which they appear. For example, the term 'java' will always refer to the programming language sense in a community of computer programmers. It would thus be sensible to add an amount of personalization to the disambiguation process for a knowledge community.

Some personalization was added to the disambiguation approach presented in chapter 4 via the automatic generation of a MOAT RDF fragment (Passant and Laublet, 2008). MOAT allows one to establish a link between terms, senses and users. As a result, community-specific biases with respect to term-sense pairs can be inferred. These term-concept pairs are then stored using the MOAT ontology in order to differentiate between *global meanings*, as the list of all meanings that could be related to a tag in a folksonomy space and *personal meanings*, related to a specific user or Community of Practice. As a result, not only the meanings that can be assigned to a given tag are available, but also which person or community 'assigned' this meaning.

The disambiguation approach presented in this dissertation does not explicitly take the social context of the tagging data into account. Including this information should yield considerable improvements to the disambiguation accuracy once enough community-specific preferences are established. However, it could also introduce negative side-effects related to greedy personalization (Markus and Westerhout, 2011) and the filter bubble (Bozdag and Timmermans, 2011). Nevertheless, a social bias for certain word senses should improve disambiguation performance for situations where the available disambiguation alone is insufficient. One can see this as a sophisticated variant of the most frequent word sense, i.e. the biased word sense for an entire community. The primary challenge with respect to community or individual biases for word disambiguation is to detect topic or context switches. For example, an expert on snakes who is also a part-time Python programmer, could get frustrated when his disambiguation bias for snakes is incorrectly applied to programming language resources and vice-versa. These community or individual biases for word senses cannot be studied using generic newspaper corpora, but require the social context in which they appear.

Personalized Topic labeling Most approaches to topic modeling take the topic model itself as central and optimize the presentation of individual topics along dimensions of topic distinctiveness or coverage. They start from the assumption that there is a single, optimal way to represent the information in a topic model (Mei et al., 2007; Lau et al., 2011) for all users. For example, Mei et al. (2007) use phrases extracted from the corpus as opposed to simple terms and Lau et al. (2011) selects an appropriate Wikipedia article title for a topic to improve its intelligibility. However, I propose to explicitly include the fact that learners vary in their background knowledge as my evaluation of topic models in chapter 6 has shown.

In the context of TOMOFF, the topic model itself is only used as an effective abstraction method for a collection of documents. The fact that documents are rated allows one to optimize the presentation of the topics based on the lexical differences between the difficulty level of a document. More specifically, the topic model can tell what terms are associated with a lower difficult rating. This allows one to select terms for a difficult topic that are known to be easier to understand for a specific learner. More specifically, a personalized representation of

a topic should use terms that both represent that topic sufficiently and maximize the overlap with terms which occur in easy to understand documents. For instance, in order to explain a physics experiment using an analogy, one should select an analogy that is appropriate for the audience. The right analogy to use for the explanation is likely to vary between audiences.

Similarly, topics should become clearer when there is increased overlap with the vocabulary that the learner is familiar with. This constitutes a trade-off between comprehensible terms on the one hand and topic-representative terms on the other. Making this trade-off successfully can yield more intelligible topics for unfamiliar information. In a sense, the representation of a topic model adapts to a learner's specific background knowledge. This constitutes an extension of the existing methodologies for topic labeling.

Semantic representations of supervised topic models In chapter 6, supervised (MedLDA) and unsupervised (LDA) topic models have been presented. Additionally, it has been established that there is a match between how humans cluster documents into topics and how this same task is performed by the LDA topic modeling algorithm. MedLDA has been proposed as a hybrid of LDA and Support Vector Machines (SVM) in order to allow for supervised topic modeling. SVM models have been successful at a wide range of tasks with respect to classification, but the model that they establish is opaque, i.e. a black box. MedLDA is a combination of a transparent, semantically meaningful model, LDA, combined with a black box model. The MedLDA model needs to be transparent to be useful in the topic labeling task of TOMOFF, but it is unclear what the effect of the inclusion of the SVM-part of the MedLDA is on the semantics of the resulting topics.

We do know that MedLDA generates topic distributions which are both optimized for prediction and the lexico-semantic patterns in a corpus. In the evaluation in chapter 6 we also established that users are still able to identify semantically meaningful information from the MedLDA topic model. One would expect the semantic transparency of the MedLDA model to suffer from the increase in classification performance afforded by the SVM component. However, no large variations in the grades assigned by students to either LDA or MedLDA-based topic models have been observed (see chapter 6, figs. 6.3a and 6.3b). It is not clear how the topics inferred through MedLDA differ semantically from those inferred using the classic LDA algorithm.

The open question surrounding the semantics of the MedLDA-style models also affect the topic label quality score. The topic label quality score algorithm, presented in section 5.4, is based on an LDA-based topic model structure. The evaluation results show that it provides an advantage over simply counting the number of absent topic labels, e.g. compare table 6.15 and table 6.17 in chapter 6 for the `lda` model. However, there is no consistent performance gain observed when using the topic label quality scores for MedLDA-based models. This suggests that the assumptions on which the topic labeling scoring algorithm is based may not hold for MedLDA-based models, which would explain the inconsistent results. The study in this dissertation is too limited to provide any definitive answers with respect to the underlying cause of the difference between MedLDA and LDA models.

Effect of lexical enrichment It has been argued in chapter 3 that an enriched ontology can mediate between an individual and a community using its lexicalizations. An individual

enters a search query and the ontology, through its domain model and lexicalizations, can automatically switch to more appropriate lexicalizations if required. An individual that has the vocabulary of a novice might not be able to retrieve resources that use an expert's vocabulary, but the ontology can automatically compensate. This contribution is a first step in the direction of automatic mediation between individual and community-specific vocabularies. Future work on this issue might establish that, although the ontology can mediate, there might be situations where this in itself is not enough.

At present, comprehension differences are often modeled at the conceptual level, thereby ignoring subtle lexical cues. Comprehension is identified with moving from generic concepts down to more specific ones in a hierarchical fashion or the conceptual coverage of a domain increases in a breadth-first manner. When one incorporates the lexical cues that community members use to retrieve resources at their own level of comprehension, search queries might actually become more useful and relevant for learners at different levels of expertise. This is accomplished in this dissertation via the use of lexical enrichment of the domain ontology and semantic search.

We know that the vocabulary that an individual employs correlates with the knowledge that the individual has. In a sense, the use of a novice's vocabulary will increase that individual's chances of encountering resources on a similar level of expertise. In some cases, this might actually be preferable if this means that a novice retrieves comprehensible, as opposed to, incomprehensible, but 'high quality' resources. Automatic selection of lexical alternatives is still appropriate if it is used to align an individual's vocabulary with that of a similarly knowledgeable community.

One way to address this, is to extend the lexicon of an ontology with the comprehension level to which each term is associated. One can imagine that this can be accomplished by building on the work presented in this dissertation, i.e. to include the interrelation of communities, lexicalizations and a learner's integration in a community in an ontology's lexicon. I implicitly assume that greater integration in a Community of Practice corresponds with increased knowledge of that community's concepts and vocabulary. Each term in the lexicon would have to be annotated with its respective community. This can already be realized with MOAT (Passant and Laublet, 2008), and a measure of community integration that is associated with the term's use. As such, the ontology's lexicon can be used to mediate between a learner and knowledge community, while taking into account the relevant lexical cues and their associated level of comprehension.

This concludes the list of possible avenues for future research. I am confident that researchers will continue to reap the benefits of comprehensive integration between large amounts of knowledge-poor and small amounts of knowledge-rich artifacts. It is clear that there are many alternative, initially plausible, approaches to the problems I addressed in this dissertation. However, whether they will be better able to help the uncle learn his nephew's world of ~~dinosaurs~~ Pterosauria than the specific approach I have taken in this dissertation remains to be seen.

Bibliography

Adamic, L. A. and Huberman, B. A. (2002). Zipf's law and the internet. Glottometrics, 3(1):143–150.

Agirre, E., De Lacalle, O., Fellbaum, C., Hsieh, S., Tesconi, M., Monachini, M., Vossen, P., and Segers, R. (2010). Semeval-2010 task 17: All-words word sense disambiguation on a specific domain. In Proceedings of the 5th International Workshop on Semantic Evaluation, pages 75–80. Association for Computational Linguistics.

Agirre, E. and Rigau, G. (1996). Word sense disambiguation using conceptual density. In Proceedings of the 16th conference on Computational linguistics-Volume 1, pages 16–22. Association for Computational Linguistics.

Alves, H. and Santanche, A. (2011). Folksonomized ontologies — from social to formal. In XVII Brazilian Symposium on Multimedia and the Web, pages 58–65.

Alves, H. and Santanche, A. (2012). Folksonomized ontology and the 3e steps technique to support ontology evolvement. Web Semantics: Science, Services and Agents on the World Wide Web.

Ames, M. and Naaman, M. (2007). Why we tag: motivations for annotation in mobile and online media. In Proceedings of the SIGCHI conference on Human factors in computing systems, pages 971–980. ACM.

Anaya-Sánchez, H., Pons-Porrata, A., and Berlanga-Llavori, R. (2007). Tkb-uo: using sense clustering for wsd. In 4th International Workshop on Semantic Evaluations (SemEval), Prague, Czech Republic, Association for Computational Linguistics.

Andrews, P., Pane, J., and Zaihrayeu, I. (2010). Semantics disambiguation in folksonomy: a case study. Technical report, Dipartimento di Ingegneria e Scienza Dell'Informazione,Universit degli studi di Trento. http://eprints.biblio.unitn.it/archive/00001933/01/063.pdf.

Andrzejewski, D. and Buttler, D. (2011). Latent topic feedback for information retrieval. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 600–608. ACM.

Angeletou, S. (2010). Semantic enrichment of folksonomy tagspaces. The Semantic Web-ISWC 2008, pages 889–894.

Angeletou, S., Sabou, M., and Motta, E. (2008). Semantically enriching folksonomies with flor. CISWeb 2008, page 65.

Angeletou, S., Sabou, M., Specia, L., and Motta, E. (2007). Bridging the gap between folksonomies and the semantic web: An experience report. In Workshop: Bridging the Gap between Semantic Web and Web, volume 2.

Antoniou, G. and Harmelen, F. (2009). Web ontology language: Owl. Handbook on ontologies, pages 91–110.

Antoniou, G. and Van Harmelen, F. (2004). A semantic web primer. MIT press.

Artiles, J., Gonzalo, J., and Sekine, S. (2007). The semeval-2007 weps evaluation: Establishing a benchmark for the web people search task. Proceedings of Semeval, pages 64–69.

Artiles, J., Gonzalo, J., and Sekine, S. (2009). Weps 2 evaluation campaign: overview of the web people search clustering task. In 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference.

Auer, S. and Herre, H. (2007). RapidOWL—An Agile Knowledge Engineering Methodology. Perspectives of Systems Informatics, pages 424–430.

Baker, L. (1989). Metacognition, comprehension monitoring, and the adult reader. Educational Psychology Review, 1(1):3–38.

Bakhtin, M., Holquist, M., and Emerson, C. (1986). Speech genres and other late essays. Univ of Texas Pr.

Bao, J., Caragea, D., and Honavar, V. (2006). Towards collaborative environments for ontology construction and sharing. In Collaborative Technologies and Systems, 2006. CTS 2006. International Symposium on, pages 99–108. IEEE.

Basca, C., Corlosquet, S., Cyganiak, R., Fernández, S., and Schandl, T. (2008). Neologism: Easy Vocabulary Publishing. In Proceedings of the Workshop on Scripting for the Semantic Web, in conjunction with ESWC, volume 2008.

Beattie IV, V., Collins, B., and Mcinnes, B. (1997). Deep and surface learning: a simple or simplistic dichotomy? Accounting Education, 6(1):1–12.

Bergman, M. K. (2010). What is a reference concept? `http://www.mkbergman.com/938/what-is-a-reference-concept/`.

Berlanga, A., Van Rosmalen, P., Trausan-Matu, S., Monachesi, P., and Burek, G. (2009). The language technologies for lifelong learning project. In Advanced Learning Technologies, 2009. ICALT 2009. Ninth IEEE International Conference on, pages 624–625. IEEE.

Berners-Lee, T. (1998). Why rdf model is different from the xml model. http://www.w3.org/DesignIssues/RDF-XML.

Berners-Lee, T., Hendler, J., Lassila, O., et al. (2001). The semantic web. Scientific american, 284(5):28–37.

Bhagdev, R., Chapman, S., Ciravegna, F., Lanfranchi, V., and Petrelli, D. (2008). Hybrid search: Effectively combining keywords and semantic searches. In Proceedings of the 5th European semantic web conference on The semantic web: research and applications, pages 554–568. Springer-Verlag.

Bischoff, K., Firan, C., Nejdl, W., and Paiu, R. (2008). Can all tags be used for search? In Proceeding of the 17th ACM conference on Information and knowledge management, pages 193–202. ACM.

Bizer, C., Heath, T., and Berners-Lee, T. (2009a). Linked data-the story so far. International Journal on Semantic Web and Information Systems (IJSWIS), 5(3):1–22.

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., and Hellmann, S. (2009b). DBpedia-A crystallization point for the Web of Data. Web Semantics: Science, Services and Agents on the World Wide Web, 7(3):154–165.

Blei, D. (2011). Introduction to probabilistic topic models. CACM.

Blei, D. and Lafferty, J. (2006). Dynamic topic models. In Proceedings of the 23rd international conference on Machine learning, pages 113–120. ACM.

Blei, D. and McAuliffe, J. (2010). Supervised topic models. Arxiv preprint arXiv:1003.0783.

Blei, D., Ng, A., and Jordan, M. (2003). Latent dirichlet allocation. The Journal of Machine Learning Research, 3:993–1022.

Blei, D. M. (2012). Probabilistic topic models. Communications of the ACM, 55(4):77–84.

Boehm, B. (1988). A spiral model of software development and enhancement. Computer, 21(5):61–72.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data, pages 1247–1250. ACM.

Bonino, D., Corno, F., Farinetti, L., and Bosca, A. (2004). Ontology driven semantic search. WSEAS Transaction on Information Science and Application, 1(6):1597–1605.

Bontcheva, K. and Rout, D. (2012). Making sense of social media streams through semantics: a survey. Semantic Web Journal.

Borthwick, A. (1999). A maximum entropy approach to named entity recognition. PhD thesis, New York University.

Bouma, G. (2010). Cross-lingual ontology alignment using eurowordnet and wikipedia. In Proceedings of the 7th Conference on International Language Resources and Evaluation (LREC 2010), pages 1023–1028.

Bozdag, E. and Timmermans, J. (2011). Values in the filter bubble ethics of personalization algorithms in cloud computing. In 1st International Workshop on Values in Design–Building Bridges between RE, HCI and Ethics, page 7.

Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. Computer networks and ISDN systems, 30(1):107–117.

Broekstra, J. and Kampman, A. (2003). Serql: a second generation rdf query language. In Proc. SWAD-Europe Workshop on Semantic Web Storage and Retrieval, pages 13–14.

Brown, J., Collins, A., and Duguid, P. (1989). Situated cognition and the culture of learning. Educational Researcher, 18(1):32–42.

Brusilovsky, P. (2001). Adaptive hypermedia. User modeling and user-adapted interaction, 11(1):87–110.

Buitelaar, P. and Cimiano, P. (2008). Ontology Learning and Population: Bridging the Gap between Text and Knowledge, volume 167. IOS Press Amsterdam, The Netherlands, The Netherlands.

Buitelaar, P., Declerck, T., Frank, A., Racioppa, S., Kiesel, M., Sintek, M., Engel, R., Romanelli, M., Sonntag, D., Loos, B., et al. (2006). Linginfo: Design and applications of a model for the integration of linguistic information in ontologies. In Proceedings of OntoLex 2006. Citeseer.

Butcher, K. and Sumner, T. (2011). Self-directed learning and the sensemaking paradox. Human–Computer Interaction, 26(1-2):123–159.

Candlish, S. and Wrisley, G. (2012). Private language. In Zalta, E. N., editor, The Stanford Encyclopedia of Philosophy. Summer 2012 edition.

Castano, S., Ferrara, A., Hess, G., et al. (2006). Discovery-driven ontology evolution. In The Semantic Web Applications and Perspectives (SWAP), 3rd Italian Semantic Web Workshop, PISA, Italy, pages 18–20. Citeseer.

Castells, P., Fernandez, M., and Vallet, D. (2007). An adaptation of the vector-space model for ontology-based information retrieval. IEEE Transactions on Knowledge and Data Engineering, 19(2):261–272.

Cattuto, C., Benz, D., Hotho, A., and Stumme, G. (2010). Semantic grounding of tag relatedness in social bookmarking systems. The Semantic Web-ISWC 2008, pages 615–631.

Cha, S. (2007). Comprehensive survey on distance/similarity measures between probability density functions. International journal of mathematical models and methods in applied sciences, 1(2):1.

Chaiklin, S. (2003). The zone of proximal development in vygotsky's analysis of learning and instruction. Vygotsky's educational theory in cultural context, pages 39–64.

Chang, J., Gerrish, S., Wang, C., and Blei, D. (2009). Reading tea leaves: How humans interpret topic models. New York, 31:1–9.

Chi, E. and Mytkowicz, T. (2008). Understanding the efficiency of social tagging systems using information theory. In Proceedings of the nineteenth ACM conference on Hypertext and hypermedia, pages 81–88. ACM.

Chi, F. and Yang, N. (2010). Twitter in congress: Outreach vs transparency. MPRA Paper.

Chi, M., De Leeuw, N., Chiu, M., and LaVancher, C. (1994). Eliciting self-explanations improves understanding. Cognitive science, 18(3):439–477.

Cho, H., Gay, G., Davidson, B., and Ingraffea, A. (2007). Social networks, communication styles, and learning performance in a cscl community. Computers & Education, 49(2):309–329.

Cilibrasi, R. and Vitanyi, P. (2007). The google similarity distance. IEEE Transactions on Knowledge and Data Engineering, pages 370–383.

Cimiano, P. (2006). Ontology learning and population from text: algorithms, evaluation and applications. Springer Verlag.

Cohen, J. (1988). Statistical power analysis for the behavioral sciences. Routledge Academic.

Collis, B. and Moonen, J. (2002). Flexible learning in a digital world. Open Learning: The Journal of Open and Distance Learning, 17(3):217–230.

Cremer, M. (2013). Accessing word meaning: Semantic word knowledge and reading comprehension in Dutch monolingual and bilingual fifth-graders. PhD thesis, Netherlands Graduate School of Linguistics.

Crestani, F. (1997). Application of spreading activation techniques in information retrieval. Artificial Intelligence Review, 11(6):453–482.

Cucerzan, S. (2007). Large-scale named entity disambiguation based on Wikipedia data. In Proceedings of EMNLP-CoNLL, volume 2007, pages 708–716.

Cucerzan, S. (2011). Tac entity linking by performing full-document entity extraction and disambiguation. Proc. of TAC.

Cucerzan, S. (2012). The msr system for entity linking at tac 2012. In Proceedings of the Text Analysis Conference 2012.

Cuzzocrea, A., Papadimitriou, A., Katsaros, D., and Manolopoulos, Y. (2011). Edge betweenness centrality: A novel algorithm for qos-based topology control over wireless sensor networks. Journal of Network and Computer Applications.

Daelemans, W., Zavrel, J., Van der Sloot, K., and Van den Bosch, A. (2007). Timbl: Tilburg memorybased learner. Version, 6:07–03.

Dalsgaard, C. (2006). Social software: E-learning beyond learning management systems. European Journal of Open, Distance and E-Learning, 2006(2).

Damme, C. V., Hepp, M., and Siorpaes, K. (2007). FolksOntology: an integrated approach for turning folksonomies into ontologies. Bridging the Gap between Semantic Web and Web, 2:57–70.

Danescu-Niculescu-Mizil, C., West, R., Jurafsky, D., Leskovec, J., and Potts, C. (2013). No country for old members: User lifecycle and linguistic change in online communities. WWW'13.

d'Aquin, M., Motta, E., Sabou, M., Angeletou, S., Gridinoc, L., Lopez, V., and Guidi, D. (2008). Toward a new generation of semantic web applications. Intelligent Systems, IEEE, 23(3):20–28.

Dascalu, M., Dessus, P., Trausan-Matu, S., Bianco, M., and Nardy, A. (2013). Readerbench, an environment for analyzing text complexity and reading strategies. In Proceedings of the 16th International Conference on Artificial Intelligence in Education, page 379–388. Australian Computer Society, Inc., Springer, LNAI 7926.

Davies, J., Duke, A., and Kiryakov, A. (2009). Semantic search. Information Retrieval, pages 179–213.

Decker, S., Melnik, S., Van Harmelen, F., Fensel, D., Klein, M., Broekstra, J., Erdmann, M., and Horrocks, I. (2000). The semantic web: The roles of xml and rdf. Internet Computing, IEEE, 4(5):63–73.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., and Harshman, R. (1990). Indexing by latent semantic analysis. Journal of the American society for information science, 41(6):391–407.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society. Series B (Methodological), pages 1–38.

Dessus, P. (2009). An overview of lsa-based systems for supporting learning and teaching. Artificial Intelligence in Education. Building learning systems that care: From knowledge representation to affective modelling (AIED2009), pages 157–164.

DeVries, R. (2000). Vygotsky, piaget, and education: A reciprocal assimilation of theories and educational practices. New Ideas in Psychology, 18(2-3):187–213.

Diederich, J. and Iofciu, T. (2006). Finding communities of practice from user profiles based on folksonomies. In Proceedings of the 1st International Workshop on Building Technology Enhanced Learning solutions for Communities of Practice, pages 288–297. Citeseer.

dos Reis, J., Bonacin, R., and Baranauskas, M. (2011). Beyond the social search: personalizing the semantic search in social networks. Online Communities and Social Computing, pages 345–354.

Dredze, M., Wallach, H. M., Puller, D., and Pereira, F. (2008). Generating summary keywords for emails using topics. In Proceedings of the 13th international conference on Intelligent user interfaces, pages 199–206. ACM.

Dreyfus, S. (2004). The five-stage model of adult skill acquisition. Bulletin of science, technology & society, 24(3):177–181.

Dunn, K. and Mulvenon, S. (2009). A critical review of research on formative assessment: The limited scientific evidence of the impact of formative assessment in education. Practical Assessment, Research & Evaluation, 14(7):1–11.

Dutta, B. and Giunchiglia, F. (2009). Semantics are actually used. In International Conference on Semantic Web and Digital Libraries, Trento, Italy, pages 62–78.

Edmonds, P. and Cotton, S. (2001). senseval-2: Overview. In Proceedings of, pages 1–6.

Egozi, O., Markovitch, S., and Gabrilovich, E. (2011). Concept-based information retrieval using explicit semantic analysis. ACM Transactions on Information Systems (TOIS), 29(2):8.

Eraut, M. (1994). Developing professional knowledge and competence. Routledge.

Faatz, A. and Steinmetz, R. (2002). Ontology enrichment with texts from the WWW. Semantic Web Mining, page 20.

Fader, A., Soderland, S., Etzioni, O., and Center, T. (2009). Scaling wikipedia-based named entity disambiguation to arbitrary web text. In Proceedings of the IJCAI Workshop on User-contributed Knowledge and Artificial Intelligence: An Evolving Synergy, Pasadena, CA, USA, pages 21–26. Citeseer.

Falconer, S., Noy, N., and Storey, M. (2007). Ontology mapping-a user survey. In Proceedings of the Workshop on Ontology Matching (OM2007) at ISWC/ASWC2007, Busan, South Korea, pages 113–125.

Fellbaum, C. (2010). Wordnet. Theory and Applications of Ontology: Computer Applications, pages 231–243.

Flavell, J. H. (1979). Metacognition and cognitive monitoring: A new area of cognitive–developmental inquiry. American psychologist, 34(10):906.

Fogarolli, A. (2009). Word sense disambiguation based on wikipedia link structure. In 2009 IEEE International Conference on Semantic Computing, pages 77–82. IEEE.

Fortuna, B., Lavrač, N., and Velardi, P. (2008). Advancing topic ontology learning through term extraction. PRICAI 2008: Trends in Artificial Intelligence, pages 626–635.

Fowler, M. and Highsmith, J. (2001). The agile manifesto. Software Development, 9(8):28–35.

Fruchterman, T. and Reingold, E. (1991). Graph drawing by force-directed placement. Software- Practice and Experience, 21(11):1129–1164.

Furnas, G., Landauer, T., Gomez, L., and Dumais, S. (1987). The vocabulary problem in human-system communication. Communications of the ACM, 30(11):964–971.

Gabrilovich, E. and Markovitch, S. (2006). Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In Proceedings of the National Conference on Artificial Intelligence, volume 21, page 1301. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Gangemi, A., Navigli, R., and Velardi, P. (2003). The OntoWordNet Project: extension and axiomatization of conceptual relations in WordNet. On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE, pages 820–838.

Garca-Silva, A., Szomszor, M., Alani, H., and Corcho, O. (2009). Preliminary results in tag disambiguation using dbpedia. In International Conference on Knowledge Capture-Workshop on Collective Knowledge Capturing and Representation, Redondo Beach, CA, USA. Citeseer.

García-Silva, A., Corcho, Ó., Alani, H., and Gómez-Pérez, A. (2011). Review of the state of the art: Discovering and associating semantics to tags in folksonomies. The Knowledge Engineering Review, 26(4).

Gemmell, J., Shepitsen, A., Mobasher, B., and Burke, R. (2008). Personalizing navigation in folksonomies using hierarchical tag clustering. Data Warehousing and Knowledge Discovery, pages 196–205.

Genesereth, M. and Nilsson, N. (1987). Logical foundations of artificial intelligence. Kaufmann.

Gennari, J., Musen, M., Fergerson, R., Grosso, W., Crubézy, M., Eriksson, H., Noy, N., and Tu, S. (2003). The evolution of Protégé: an environment for knowledge-based systems development. International Journal of Human-Computer Studies, 58(1):89–123.

Ghidini, C., Kump, B., Lindstaedt, S., Mahbub, N., Pammer, V., Rospocher, M., and Serafini, L. (2009). Moki: The enterprise modelling wiki. The Semantic Web: Research and Applications, pages 831–835.

Gil-García, R., Badía-Contelles, J., and Pons-Porrata, A. (2003). Extended star clustering algorithm. Progress in Pattern Recognition, Speech and Image Analysis, pages 480–487.

Girvan, M. and Newman, M. (2002). Community structure in social and biological networks. Proceedings of the National Academy of Sciences, 99(12):7821.

Giunchiglia, F., Dutta, B., and Maltese, V. (2009). Faceted lightweight ontologies. Conceptual Modeling: Foundations and Applications, pages 36–51.

Glassman, M. (1995). The difference between piaget and vygotsky: A response to duncan. Developmental Review, 15(4):473–482.

Gog, T., Kester, L., Nievelstein, F., Giesbers, B., and Paas, F. (2009). Uncovering cognitive processes: Different techniques that can contribute to cognitive load research and instruction. Computers in Human Behavior, 25(2):325–331.

Golder, S. and Huberman, B. (2006). Usage patterns of collaborative tagging systems. Journal of information science, 32(2):198.

Griffiths, T. and Steyvers, M. (2004). Finding scientific topics. Proceedings of the National Academy of Sciences of the United States of America, 101(Suppl 1):5228.

Gruber, T. (1993). What is an Ontology? Knowledge Acquisition, 5(2):199–220.

Gruber, T. (2007). Ontology of folksonomy: A mash-up of apples and oranges. International Journal on Semantic Web and Information Systems (IJSWIS), 3(1):1–11.

Guha, R., McCool, R., and Miller, E. (2003). Semantic search. In Proceedings of the 12th international conference on World Wide Web, pages 700–709. ACM.

Guo, S. and Ramakrishnan, N. (2010). Finding the storyteller: automatic spoiler tagging using linguistic cues. In Proceedings of the 23rd International Conference on Computational Linguistics, pages 412–420. Association for Computational Linguistics.

Haase, P., Lewen, H., Studer, R., Tran, D., Erdmann, M., d'Aquin, M., and Motta, E. (2008). The neon ontology engineering toolkit. In WWW.

Han, X. and Zhao, J. (2010). Structural semantic relatedness: a knowledge-based method to named entity disambiguation. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 50–59. Association for Computational Linguistics.

Harvey, M., Baillie, M., Ruthven, I., and Carman, M. (2010). Tripartite hidden topic models for personalised tag suggestion. Advances in Information Retrieval, pages 432–443.

Hasan, Y. C. S. A. (2012). Automatically assessing free texts. In 24th International Conference on Computational Linguistics, page 9.

Hatcher, J. and Bringle, R. (1997). Reflection: Bridging the gap between service and learning. College teaching, 45(4):153–158.

Hein, J. and Hendler, J. (2000). Searching the web with shoe. Proceedings of the Artificial Intelligence for Web Search, AAAI Press, CA., pages 35–40.

Hellmann, S., Stadler, C., Lehmann, J., and Auer, S. (2009). Dbpedia live extraction. On the Move to Meaningful Internet Systems: OTM 2009, pages 1209–1223.

Hepp, M. (2007). Possible Ontologies: How Reality Constrains the Development of Relevant Ontologies. IEEE Internet Computing, pages 90–96.

Heymann, P. and Garcia-Molina, H. (2006). Collaborative creation of communal hierarchical taxonomies in social tagging systems. Technical Report 10, InfoLab.

Heymann, P., Koutrika, G., and Garcia-Molina, H. (2008). Can social bookmarking improve web search? In Proceedings of the international conference on Web search and web data mining, pages 195–206. ACM.

Hildebrand, M., Ossenbruggen, J., and Hardman, L. (2007). An analysis of search-based user interaction on the semantic web. CWI. Information Systems [INS], (E0706).

Hirschman, L. and Gaizauskas, R. (2001). Natural language question answering: The view from here. Natural Language Engineering, 7(04):275–300.

Hobbs, J. (1990). Topic drift. Conversational organization and its development, 38:3–22.

Hoehndorf, R. (2010). What is an upper level ontology? `http://ontogenesis.knowledgeblog.org/740` [Online. last-accessed: 2012-06-19 16:13:42].

Hoffman, M., Blei, D., and Bach, F. (2010). Online learning for latent dirichlet allocation. Advances in Neural Information Processing Systems, 23:856–864.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, pages 50–57. ACM.

Hotho, A., Jäschke, R., Schmitz, C., and Stumme, G. (2006a). Information retrieval in folksonomies: Search and ranking. The Semantic Web: Research and Applications, pages 411–426.

Hotho, A., Jäschke, R., Schmitz, C., and Stumme, G. (2006b). Information Retrieval in Folksonomies: Search and Ranking. In The Semantic Web: Research and Applications, volume 4011 of Lecture Notes in Computer Science, pages 411–426, Berlin/Heidelberg. Springer-Verlag.

Huhta, A. (2007). Diagnostic and formative assessment. The handbook of educational linguistics, pages 469–482.

Hulpus, I., Hayes, C., Karnstedt, M., and Greene, D. (2013). Unsupervised graph-based topic labelling using dbpedia. In Proceedings of the sixth ACM international conference on Web search and data mining, pages 465–474. ACM.

Hung, D. and Chen, D. (2001). Situated cognition, vygotskian thought and learning from the communities of practice perspective: Implications for the design of web-based e-learning. Educational Media International, 38(1):3–12.

Hunter, J. and Lagoze, C. (2001). Combining rdf and xml schemas to enhance interoperability between metadata application profiles. In Proceedings of the 10th international conference on World Wide Web, pages 457–466. ACM.

Jansen, B., Booth, D., and Spink, A. (2008). Determining the informational, navigational, and transactional intent of web queries. Information Processing & Management, 44(3):1251–1266.

Janssen, J. (2008). Using visualizations to support collaboration and coordination during computer-supported collaborative learning. PhD thesis, Universiteit Utrecht.

Ji, Q., Gao, Z., and Huang, Z. (2011). Reasoning with noisy semantic data. The Semanic Web: Research and Applications, pages 497–502.

John-Steiner, V. and Mahn, H. (1996). Sociocultural approaches to learning and development: A vygotskian framework. Educational Psychologist, 31(3):191–206.

Jordan, S. (2011). Student engagement with assessment and feedback: some lessons from short-answer free-text e-assessment questions. Computers & Education.

Jorge-Botana, G., Leon, J. A., Olmos, R., and Escudero, I. (2010). Latent semantic analysis parameters for essay evaluation using small-scale corpora*. Journal of Quantitative Linguistics, 17(1):1–29.

Jäschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., and Stumme, G. (2007). Tag recommendations in folksonomies. In Kok, J., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenic, D., and Skowron, A., editors, Knowledge Discovery in Databases: PKDD 2007, volume 4702 of Lecture Notes in Computer Science, pages 506–514. Springer Berlin / Heidelberg.

Kalfoglou, Y. and Schorlemmer, M. (2003). Ontology mapping: the state of the art. The knowledge engineering review, 18(01):1–31.

Kaplan, A. and Haenlein, M. (2010). Users of the world, unite! the challenges and opportunities of social media. Business horizons, 53(1):59–68.

Kato, M., Ohshima, H., Oyama, S., and Tanaka, K. (2008). Can social tagging improve web image search? Web Information Systems Engineering-WISE 2008, pages 235–249.

Keet, C. (2011). The use of foundational ontologies in ontology development: an empirical assessment. The Semantic Web: Research and Applications, pages 321–335.

Kemps-Snijders, M., Windhouwer, M., Wittenburg, P., and Wright, S. (2009). Isocat: remodelling metadata for language resources. International Journal of Metadata, Semantics and Ontologies, 4(4):261–276.

Khattak, A., Latif, K., Lee, S., and Lee, Y. (2009). Ontology evolution: A survey and future challenges. U-and E-Service, Science and Technology, pages 68–75.

Khattak, A., Pervez, Z., Lee, S., and Lee, Y. (2010). After effects of ontology evolution. In Future Information Technology (FutureTech), 2010 5th International Conference on, pages 1–6. IEEE.

Kilgarriff, A. and Rosenzweig, J. (2000). Framework and results for english senseval. Computers and the Humanities, 34(1):15–48.

Kim, H., Breslin, J., Yang, S., and Kim, H. (2008a). Social semantic cloud of tag: semantic model for social tagging. In Proceedings of the 2nd KES International conference on Agent and multi-agent systems: technologies and applications, pages 83–92. Springer-Verlag.

Kim, H., Passant, A., Breslin, J., Scerri, S., and Decker, S. (2008b). Review and alignment of tag ontologies for semantically-linked data in collaborative tagging spaces. In Semantic Computing, 2008 IEEE International Conference on, pages 315–322. IEEE.

Kinchin, I. and Cabot, L. (2010). Reconsidering the dimensions of expertise: from linear stages towards dual processing. London Review of Education, 8(2):153–166.

Kirkwood, A. (2009). E-learning: you don't always get what you hope for. Technology, Pedagogy and Education, 18(2):107–121.

Kirshenblatt-Gimblett, B. (1996). Topic drift: Negotiating the gap between the field and our name. Journal of Folklore Research, 33(3):245–254.

Klasnja-Milicevic, A., Vesin, B., Ivanovic, M., and Budimac, Z. (2011). E-learning personalization based on hybrid recommendation strategy and learning style identification. Computers & Education, 56(3):15.

Kleinberg, J. (1999). Authoritative sources in a hyperlinked environment. Journal of the ACM (JACM), 46(5):604–632.

Krestel, R., Fankhauser, P., and Nejdl, W. (2009). Latent dirichlet allocation for tag recommendation. In Proceedings of the third ACM conference on Recommender systems, pages 61–68. ACM.

Krötzsch, M., Vrandečić, D., and Völkel, M. (2006). Semantic mediawiki. The Semantic Web-ISWC 2006, pages 935–942.

Kwon, S. and Cifuentes, L. (2009). The comparative effect of individually-constructed vs. collaboratively-constructed computer-based concept maps. Computers & Education, 52(2):365–375.

Lai, E. R. (2011). Metacognition: A literature review. Technical report, Research Report.

Landauer, T. (2003). Automatic essay assessment. Assessment in education: Principles, policy & practice, 10(3):295–308.

Landauer, T., Foltz, P., and Laham, D. (1998). An introduction to latent semantic analysis. Discourse processes, 25:259–284.

Lau, J., Grieser, K., Newman, D., and Baldwin, T. (2011). Automatic labelling of topic models. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pages 1536–1545. Association for Computational Linguistics.

Lee, L. (2001). On the effectiveness of the skew divergence for statistical language analysis. In Artificial Intelligence and Statistics, volume 2001, pages 65–72.

Lemaire, B. and Dessus, P. (2001). A system to assess the semantic content of student essays. Journal of Educational Computing Research, 24(3):305–320.

Lemnitzer, L. and Monachesi, P. (2008). Extraction and evaluation of keywords from learning objects–a multilingual approach. In Proceedings of the Language Resources and Evaluation Conference (LREC 2008).

Lemnitzer, L., Mossel, E., Simov, K., Osenova, P., and Monachesi, P. (2008). Using a Domain-Ontology and Semantic Search in an E-Learning Environment. Innovative Techniques in Instruction Technology, E-learning, E-assessment, and Education, pages 279–284.

Lemnitzer, L., Vertan, C., Killing, A., Simov, K., Evans, D., Cristea, D., and Monachesi, P. (2007). Improving the search for learning objects with keywords and ontologies. Creating New Learning Experiences on a Global Scale, pages 202–216.

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In Proceedings of the 5th annual international conference on Systems documentation, pages 24–26. ACM.

Lewis, S., Pea, R., and Rosen, J. (2010). Beyond participation to co-creation of meaning: mobile social media in generative learning communities. Social Science Information, 49(3):351–369.

Ley, T., Kump, B., and Gerdenitsch, C. (2010). Scaffolding Self-directed Learning with Personalized Learning Goal Recommendations. User Modeling, Adaptation, and Personalization, pages 75–86.

Li, C., Sun, A., and Datta, A. (2011). A generalized method for word sense disambiguation based on wikipedia. Advances in Information Retrieval, pages 653–664.

Lim, S. (2009). How and why do college students use wikipedia? Journal of the American Society for Information Science and Technology, 60(11):2189–2202.

Lin, D. (1998). An information-theoretic definition of similarity. In Proceedings of the 15th international conference on machine learning, volume 1, pages 296–304. Citeseer.

Lin, H., Davis, J., and Zhou, Y. (2009). An integrated approach to extracting ontological structures from folksonomies. The Semantic Web: Research and Applications, pages 654–668.

Lin, X., Hmelo, C., Kinzer, C., and Secules, T. (1999). Designing technology to support reflection. Educational Technology Research and Development, 47(3):43–62.

Liu, S., Zhou, M., Pan, S., Qian, W., Cai, W., and Lian, X. (2009). Interactive, topic-based visual text summarization and analysis. In Proceeding of the 18th ACM conference on Information and knowledge management, pages 543–552. ACM.

Loiseau, M., Dupré, D., Dessus, P., et al. (2011). Pensum, un système d'aide à la compréhension de cours à distance. EIAH'2011: A la recherche des convergences entre les acteurs des EIAH, pages 287–299.

Loper, E. and Bird, S. (2002). Nltk: The natural language toolkit. In Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1, pages 63–70. Association for Computational Linguistics.

Luczak-Rösch, M. (2009). Towards Agile Ontology Maintenance. The Semantic Web-ISWC 2009, pages 965–972.

Luong, H., Gauch, S., and Wang, Q. (2009). Ontology-based Focused Crawling. In International Conference on Information, Process, and Knowledge Management, pages 123–128. IEEE.

Manandhar, S., Klapaftis, I., Dligach, D., and Pradhan, S. (2010). Semeval-2010 task 14: Word sense induction & disambiguation. In Proceedings of the 5th International Workshop on Semantic Evaluation, pages 63–68. Association for Computational Linguistics.

Mangold, C. (2007). A survey and classification of semantic search approaches. International Journal of Metadata, Semantics and Ontologies, 2(1):23–34.

Marconi, D. (1995). On the structure of lexical competence. Proceedings of the Aristotelian Society, 95:pp. 131–150.

Marenzi, I., Demidova, E., and Nejdl, W. (2008). LearnWeb 2.0. Integrating Social Software for Lifelong Learning. In Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications, pages 1793–1802.

Markham, K. M., Mintzes, J. J., and Jones, M. G. (1994). The concept map as a research and evaluation tool: Further evidence of validity. Journal of research in science teaching, 31(1):91–101.

Markus, T. and Westerhout, E. (2011). Hidden patterns in learner feedback. Proceedings of the Third International Conference on Computer Supported Education, 3.

Marlow, C., Naaman, M., Boyd, D., and Davis, M. (2006). Position paper, tagging, taxonomy, flickr, article, toread. In In Collaborative Web Tagging Workshop at WWWÕ06. Citeseer.

Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A., Oltramari, R., Schneider, L., Istc-cnr, L., and Horrocks, I. (2002). Wonderweb deliverable d17. the wonderweb library of foundational ontologies and the dolce ontology.

Matuszek, C., Cabral, J., Witbrock, M., and DeOliveira, J. (2006). An introduction to the syntax and content of cyc. In Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering, page 44–49.

Means, B., Toyama, Y., Murphy, R., Bakia, M., and Jones, K. (2010). Evaluation of Evidence-Based Practices in Online Learning: A Meta-Analysis and Review of Online Learning Studies. Technical report, Department of Education, Office of Planning, Evaluation, and Policy Development.

Mei, Q., Shen, X., and Zhai, C. (2007). Automatic labeling of multinomial topic models. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 490–499. ACM.

Mendes, P. N., Jakob, M., García-Silva, A., and Bizer, C. (2011). Dbpedia spotlight: Shedding light on the web of documents. In Proceedings of the 7th International Conference on Semantic Systems (I-Semantics).

Mihalcea, R., Chklovski, T., and Kilgarriff, A. (2004). The senseval-3 english lexical sample task. In Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, pages 25–28. Barcelona, Spain, Association for Computational Linguistics.

Mihalcea, R. and Csomai, A. (2007). Wikify!: linking documents to encyclopedic knowledge. In Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, pages 233–242. ACM.

Mika, P. (2005). Ontologies are us: A unified model of social networks and semantics. The Semantic Web–ISWC 2005, pages 522–536.

Miles, A., Matthews, B., Wilson, M., and Brickley, D. (2005). SKOS Core: Simple knowledge organisation for the web. Proceedings of the International Conference on Dublin Core and Metadata Applications, 5:12–15.

Miller, G. (1995). WordNet: a lexical database for English. Communications of the ACM, 38(11):39–41.

Milne, D., Medelyan, O., and Witten, I. (2006). Mining domain-specific thesauri from wikipedia: A case study. In Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, pages 442–448. IEEE Computer Society.

Mimno, D. and McCallum, A. (2007). Organizing the oca: learning faceted subjects from a library of digital books. In Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries, pages 376–385. ACM.

Monachesi, P. and Markus, T. (2010a). Socially driven ontology enrichment for eLearning. In Proceedings of The Language Resources and Evaluation Conference, LREC 2010.

Monachesi, P. and Markus, T. (2010b). Using social media for ontology enrichment. The Semantic Web: Research and Applications, pages 166–180.

Monachesi, P., Markus, T., and Mossel, E. (2009). Ontology enrichment with social tags for elearning. In Cress, U., Dimitrova, V., and Specht, M., editors, Learning in the Synergy of Multiple Disciplines, Proceedings of the EC-TEL 2009, volume 5794 of Lecture Notes in Computer Science. Springer.

Monachesi, P., Markus, T., Osenova, P., Posea, V., Simov, K., and Trausan-Matu, S. (2010). Supporting knowledge discovery in an elearning environment having social components. In Elleithy, K., Sobh, T., Iskander, M., Kapila, V., Karim, M., and Mahmood, A., editors, Technological Developments in Networking, Education and Automation. Springer.

Monachesi, P., Markus, T., Westerhout, E., Osenova, P., and Simov, K. (2011). Supporting formal and informal learning through domain ontologies. International Conference on e-Education, e-Business, e-Management and e-Learning.

Monachesi, P., Simov, K., Mossel, E., Osenova, P., and Lemnitzer, L. (2008). What ontologies can do for eLearning. In Proceedings of the International Conference on Interactive Mobile and Computer Aided Learning, IMCL08.

Naaman, M., Boase, J., and Lai, C. (2010). Is it really about me?: message content in social awareness streams. In Proceedings of the 2010 ACM conference on Computer supported cooperative work, pages 189–192. ACM.

Nanni, M. (2005). Speeding-up hierarchical agglomerative clustering in presence of expensive metrics. Advances in Knowledge Discovery and Data Mining, pages 137–138.

Navigli, R. (2009). Word sense disambiguation: A survey. ACM Computing Surveys (CSUR), 41(2):10.

Navigli, R. and Lapata, M. (2007). Graph connectivity measures for unsupervised word sense disambiguation. In Proceedings of the 20th International Joint Conference on Artificial Intelligence, pages 1683–1688.

Navigli, R. and Velardi, P. (2003). An analysis of ontology-based query expansion strategies. In Proceedings of the 14th European Conference on Machine Learning, Workshop on Adaptive Text Extraction and Mining, Cavtat-Dubrovnik, Croatia, pages 42–49.

Navigli, R. and Velardi, P. (2006). Ontology enrichment through automatic semantic annotation of on-line glossaries. Managing Knowledge in a World of Networks, pages 126–140.

Newman, D., Asuncion, A., Smyth, P., and Welling, M. (2007). Distributed inference for latent dirichlet allocation. Advances in Neural Information Processing Systems, 20(1081-1088):17–24.

Niles, I. and Pease, A. (2001). Towards a standard upper ontology. In Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001, pages 2–9. ACM.

Noh, Y., Hagedorn, K., and Newman, D. (2011). Are learned topics more useful than subject headings. In Proceeding of the 11th annual international ACM/IEEE joint conference on Digital libraries, pages 411–412. ACM.

Novak, J. and Cañas, A. (2008). The theory underlying concept maps and how to construct and use them. Florida Institute for Human and Machine Cognition, 2008.

Novak, J. D. and Cañas, A. J. (2006). The theory underlying concept maps and how to construct them. Florida Institute for Human and Machine Cognition, 1.

Noy, N. (2009). Ontology mapping. Handbook on Ontologies, pages 573–590.

Orme, A. M., Yao, H., and Etzkorn, L. H. (2007). Indicating ontology data quality, stability, and completeness throughout ontology evolution. Journal of Software Maintenance and Evolution: Research and Practice, 19(1):49–75.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: bringing order to the web.

Palincsar, A. (1998). Social constructivist perspectives on teaching and learning. Annual review of psychology, 49(1):345–375.

Pan, J., Taylor, S., and Thomas, E. (2009). Reducing ambiguity in tagging systems with folksonomy search expansion. The Semantic Web: Research and Applications, pages 669–683.

Paris, S. G., Winograd, P., et al. (1990). How metacognition can promote academic learning and instruction. Dimensions of thinking and cognitive instruction, 1:15–51.

Passant, A. and Laublet, P. (2008). Meaning of a tag: A collaborative approach to bridge the gap between tagging and linked data. In Proceedings of the WWW 2008 Workshop Linked Data on the Web (LDOW2008), Beijing, China. Citeseer.

Pollock, J. (1987). Defeasible reasoning. Cognitive science, 11(4):481–518.

Ponzetto, S. and Navigli, R. (2010). Knowledge-rich word sense disambiguation rivaling supervised systems. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 1522–1531. Association for Computational Linguistics.

Porteous, I., Newman, D., Ihler, A., Asuncion, A., Smyth, P., and Welling, M. (2008). Fast collapsed gibbs sampling for latent dirichlet allocation. In Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 569–577. ACM.

Posea, V. and Trausan-Matu, S. (2009). Bridging ontologies and folksonomies using dbpedia.

Prakken, H. and Vreeswijk, G. (2002). Logics for defeasible argumentation. Handbook of philosophical logic, 4:218–319.

Ramirez, E., Brena, R., Magatti, D., and Stella, F. (2011). Topic model validation. Neurocomputing.

Ratinov, L., Roth, D., Downey, D., and Anderson, M. (2011). Local and global algorithms for disambiguation to wikipedia. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pages 1375–1384. Association for Computational Linguistics.

Rebedea, T., Dascalu, M., and Trausan-Matu, S. (2010a). Polycafe: Polyphony-based system for collaboration analysis and feedback generation. Natural Language Processing in Support of Learning: Metrics, Feedback and Connectivity, page 21.

Rebedea, T., Dascalu, M., Trausan-Matu, S., Banica, D., Gartner, A., Chiru, C., and Mihaila, D. (2010b). Overview and preliminary results of using polycafe for collaboration analysis and feedback generation. Sustaining TEL: From Innovation to Learning and Practice, pages 420–425.

Řehůřek, R. and Sojka, P. (2010). Software framework for topic modelling with large corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, pages 45–50, Valletta, Malta. ELRA.

Resnik, P. (1993). Selection and information: a class-based approach to lexical relationships. IRCS Technical Reports Series, page 161.

Rizzo, G. (2011). Nerd: Evaluating named entity recognition tools in the web of data.

Ronzano, F., Marchetti, A., Tesconi, M., and Minutoli, S. (2008). Tagpedia: a semantic reference to describe and search for web resources. In WWW 2008 Workshop on Social Web and Knowledge Management, Beijing, China. Citeseer.

Rubin, T., Chambers, A., Smyth, P., and Steyvers, M. (2011). Statistical topic models for multi-label document classification. Machine Learning, pages 1–52.

Rugg, G. and McGeorge, P. (1997). The sorting techniques: a tutorial paper on card sorts, picture sorts and item sorts. Expert Systems, 14(2):80–93.

Santamaría, C., Gonzalo, J., and Artiles, J. (2010). Wikipedia as sense inventory to improve diversity in web search results. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 1357–1366. Association for Computational Linguistics.

Schmitz, P. (2006). Inducing ontology from flickr tags. In Collaborative Web Tagging Workshop at WWW2006, Edinburgh, Scotland, pages 210–214.

Schönbrodt, F. D. and Perugini, M. (2013). At what sample size do correlations stabilize? Journal of Research in Personality.

Schraw, G. (1998). Promoting general metacognitive awareness. Instructional science, 26(1):113–125.

Schraw, G. and Sperling Dennison, R. (1994). Assessing metacognitive awareness. Contemporary educational psychology, 19:460–460.

Schutz, A. and Buitelaar, P. (2005). Relext: A tool for relation extraction from text in ontology extension. The Semantic Web–ISWC 2005, pages 593–606.

Schütze, H. and Pedersen, J. (1995). Information retrieval based on word senses.

Seale, J. and Cann, A. (2000). Reflection on-line or off-line: the role of learning technologies in encouraging students to reflect. Computers & Education, 34(3-4):309–320.

Semeraro, G., Degemmis, M., Lops, P., and Basile, P. (2007). Combining learning and word sense disambiguation for intelligent user profiling. In Proceedings of the 20th international joint conference on Artifical intelligence, pages 2856–2861. Morgan Kaufmann Publishers Inc.

Shadbolt, N., Hall, W., and Berners-Lee, T. (2006). The semantic web revisited. Intelligent Systems, IEEE, 21(3):96–101.

Shen, W., Wang, J., Luo, P., and Wang, M. (2012). Linden: linking named entities with knowledge base via semantic knowledge. In Proceedings of the 21st international conference on World Wide Web, pages 449–458. ACM.

Sheridan, J. and Tennison, J. (2010). Linking uk government data. BHBL+].: http://events. linkeddata. org/ldow2010/.(Cit. on p.).

Shi, X. (2007). Social network analysis of web search engine query logs. Ann Arbor, 1001:48109.

Siersdorfer, S. and Sizov, S. (2009). Social recommender systems for web 2.0 folksonomies. In Proceedings of the 20th ACM conference on Hypertext and hypermedia, pages 261–270. ACM.

Sigurbjörnsson, B. and Van Zwol, R. (2008). Flickr tag recommendation based on collective knowledge. In Proceeding of the 17th international conference on World Wide Web, pages 327–336. ACM.

Simon, J. (2010). The entanglement of trust and knowledge on the web. Ethics and information technology, pages 1–13.

Simperl, E. (2009). Reusing ontologies on the semantic web: A feasibility study. Data & Knowledge Engineering, 68(10):905–925.

Sinclair, J. and Cardew-Hall, M. (2008). The folksonomy tag cloud: when is it useful? Journal of Information Science, 34(1):15.

Sinha, R. and Mihalcea, R. (2007). Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In Semantic Computing, 2007. ICSC 2007. International Conference on, pages 363–369. IEEE.

Sirin, E., Parsia, B., Grau, B., Kalyanpur, A., and Katz, Y. (2007). Pellet: A practical owl-dl reasoner. Web Semantics: science, services and agents on the World Wide Web, 5(2):51–53.

Smith, A. J. (2002). Applications of the self-organising map to reinforcement learning. Neural Networks, 15(8-9):1107–1124.

Smith, D. B. B. O. N. A. (2013). Learning latent personas of film characters. In ACL 2013 proceedings. The Association for Computational Linguistics.

Specia, L. and Motta, E. (2007). Integrating folksonomies with the semantic web. The semantic web: research and applications, pages 624–639.

Spitkovsky, V. and Chang, A. (2012). A cross-lingual dictionary for english wikipedia concepts. In Eighth International Conference on Language Resources and Evaluation (LREC 2012) Open access.

Stahl, G. (2006). Group Cognition: Computer Support for Collaborative Knowledge Building. The MIT Press, Cambridge, MA.

Stahl, G., Koschmann, T., and Suthers, D. (2006). Computer-supported collaborative learning: An historical perspective. Cambridge handbook of the learning sciences, 2006.

Stojanovic, L., Staab, S., and Studer, R. (2001). elearning based on the semantic web. In WebNet2001-World Conference on the WWW and Internet, Orlando, Florida, USA.

Studer, R., Benjamins, V. R., and Fensel, D. (1998). Knowledge engineering: principles and methods. Data & knowledge engineering, 25(1):161–197.

Suchanek, F., Kasneci, G., and Weikum, G. (2007). Yago: a core of semantic knowledge. In Proceedings of the 16th international conference on World Wide Web, pages 697–706. ACM.

Sussna, M. (1993). Word sense disambiguation for free-text indexing using a massive semantic network. In Proceedings of the second international conference on Information and knowledge management, pages 67–74. ACM.

Szomszor, M., Alani, H., Cantador, I., O'Hara, K., and Shadbolt, N. (2008). Semantic modelling of user interests based on cross-folksonomy analysis. In Sheth, A., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., and Thirunarayan, K., editors, The Semantic Web - ISWC 2008, volume 5318 of Lecture Notes in Computer Science, pages 632–648. Springer Berlin / Heidelberg.

Tang, J., Leung, H., Luo, Q., Chen, D., and Gong, J. (2009). Towards ontology learning from folksonomies. In Proceedings of the 21st international jont conference on Artifical intelligence, pages 2089–2094. Morgan Kaufmann Publishers Inc.

Taylor, R. (1962). The process of asking questions. American Documentation, 13(4):391–396.

Tesconi, M., Ronzano, F., Marchetti, A., and Minutoli, S. (2008). Semantify del. icio. us: Automatically turn your tags into senses. In Proceedings of the First Social Data on the Web Workshop (SDoW2008). Citeseer.

Tomuro, N. and Shepitsen, A. (2009). Construction of disambiguated folksonomy ontologies using wikipedia. In Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources, pages 42–50. Association for Computational Linguistics.

Töpper, G., Knuth, M., and Sack, H. (2012). Dbpedia ontology enrichment for inconsistency detection. In Proceedings of the 8th International Conference on Semantic Systems, pages 33–40. ACM.

Tran, T., Cimiano, P., Rudolph, S., and Studer, R. (2007). Ontology-based interpretation of keywords for semantic search. The Semantic Web, pages 523–536.

Tran, T., Mika, P., Wang, H., and Grobelnik, M. (2011). Semsearch'11: the 4th semantic search workshop. In Proceedings of the 20th international conference companion on World wide web, pages 315–316. ACM.

Trausan-Matu, S., Dessus, P., Lemaire, B., Mandin, S., Villiot-Leclercq, E., Rebedea, T., Chiru, C., Mihaila, D., Gartner, A., and Zampa, V. (2008). Ltfll - d5.1: Writing support and feedback design.

Tsatsaronis, G., Varlamis, I., and Nørvåg, K. (2010). An experimental study on unsupervised graph-based word sense disambiguation. Computational Linguistics and Intelligent Text Processing, pages 184–198.

Turney, P. D., Pantel, P., et al. (2010). From frequency to meaning: Vector space models of semantics. Journal of artificial intelligence research, 37(1):141–188.

Valenti, S., Neri, F., and Cucchiarelli, A. (2003). An overview of current research on automated essay grading. Journal of Information Technology Education, 2(319330):3–118.

Van de Cruys, T. (2010). Mining for Meaning - The Extraction of Lexicosemantic Knowledge from Text. PhD thesis, Groningen University.

Van de Cruys, T., Apidianaki, M., et al. (2011). Latent semantic word sense induction and disambiguation. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL/HLT), pages 1476–1485.

Van den Boom, G., Paas, F., van Merriënboer, J., and Van Gog, T. (2004). Reflection prompts and tutor feedback in a web-based learning environment: Effects on students' self-regulated learning competence. Computers in Human Behavior, 20(4):551–567.

Vander Wal, T. (2007). Folksonomy. online posting, Feb, 7.

Vasilescu, F., Langlais, P., and Lapalme, G. (2004). Evaluating variants of the lesk approach for disambiguating words. In Proceedings of the Conference of Language Resources and Evaluations (LREC 2004), pages 633–636.

Villalon, J. and Calvo, R. A. (2009). Single document semantic spaces. In Proceedings of the Eighth Australasian Data Mining Conference-Volume 101, pages 175–181. Australian Computer Society, Inc.

Voorhees, E. (1994). Query expansion using lexical-semantic relations. In Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, pages 61–69. Springer-Verlag New York, Inc.

Voorhees, E. and Harman, D. (1999). Overview of the eighth text retrieval conference (trec-8). Voorhees et Harman.

Voss, J. (2007). Tagging, folksonomy & co-renaissance of manual indexing? Arxiv preprint cs/0701072.

Vossen, P. (2002). EuroWordNet: general document.

Wang, C., Blei, D., and Heckerman, D. (2012). Continuous time dynamic topic models. arXiv preprint arXiv:1206.3298.

Wang, H., Zhang, K., Liu, Q., Tran, T., and Yu, Y. (2008). Q2semantic: a lightweight keyword interface to semantic search. The Semantic Web: Research and Applications, pages 584–598.

Wang, X. and Grimson, E. (2007). Spatial latent dirichlet allocation. Advances in Neural Information Processing Systems, 20:1577–1584.

Wei, W., Barnaghi, P., and Bargiela, A. (2008). Search with meanings: an overview of semantic search systems. Int. J. Communications of SIWN, 3:76–82.

Weller, K. et al. (2007). Folksonomies and ontologies: two new players in indexing and knowledge representation. Applying web, 2:108–115.

Wenger, E. and Snyder, W. (2000). Communities of practice: The organizational frontier. Harvard business review, 78(1):139–146.

Westerhout, E., Markus, T., Posea, V., and Monachesi, P. (2011). Integrating social media and semantic structure in the learning process. Towards Ubiquitous Learning, pages 489–494.

Westerhout, E., Monachesi, P., Markus, T., and Posea, V. (2010). Enhancing the Learning Process: Qualitative Validation of an Informal Learning Support System Consisting of a Knowledge Discovery and a Social Learning Component. Sustaining TEL: From Innovation to Learning and Practice, pages 374–389.

Wild, F., Haley, D., and Bülow, K. (2010). Conspect: monitoring conceptual development. Advances in Web-Based Learning–ICWL 2010, pages 299–308.

Winters, F., Greene, J., and Costich, C. (2008). Self-regulation of learning within computer-based learning environments: A critical analysis. Educational Psychology Review, 20(4):429–444.

Wolfe, M. B., Schreiner, M., Rehder, B., Laham, D., Foltz, P. W., Kintsch, W., and Landauer, T. K. (1998). Learning from text: Matching readers and texts by latent semantic analysis. Discourse Processes, 25(2-3):309–336.

Wu, X., Zhang, L., and Yu, Y. (2006). Exploring social annotations for the semantic web. In Proceedings of the 15th international conference on World Wide Web, page 426. ACM.

Yano, T., Cohen, W., and Smith, N. (2009). Predicting response to political blog posts with topic models. In Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 477–485. Association for Computational Linguistics.

Yao, Y., Zeng, Y., Zhong, N., and Huang, X. (2007). Knowledge retrieval (kr). In Web Intelligence, IEEE/WIC/ACM International Conference on, pages 729 –735.

Yi, X. and Allan, J. (2009). A comparative study of utilizing topic models for information retrieval. Advances in Information Retrieval, pages 29–41.

Zhao, P., Han, J., and Sun, Y. (2009). P-rank: a comprehensive structural similarity measure over information networks. In Proceedings of the 18th ACM conference on Information and knowledge management, pages 553–562. ACM.

Zhu, J., Ahmed, A., and Xing, E. (2009). Medlda: maximum margin supervised topic models for regression and classification. In Proceedings of the 26th Annual International Conference on Machine Learning, pages 1257–1264. ACM.

Ziai, R., Ott, N., and Meurers, D. (2012). Short answer assessment: Establishing links between research strands. In Proceedings of the Seventh Workshop on Building Educational Applications Using NLP, pages 190–200. Association for Computational Linguistics.

Zipf, G. K. (1932). Selected studies of the principle of relative frequency in language.

Appendix A

Concept filtering list

A list of DBpedia resources that were automatically discarded as part of the ontology enrichment process presented in chapter 3.

- Marketing
- Literature
- Leisure
- Vocabulary
- Aesthetics
- Abstraction
- Design
- Creativity
- Humanities
- Knowledge
- Cognition
- Skill
- Phenomenon
- Phenomena Activity
- Museology
- Economics
- Sociology
- Academia
- Recreation
- Computing
- TO
- Cool
- Sociology
- Entertainment
- Main_topic_classifications
- Electronics
- List_of_decades
- Concepts
- Intellectual_works
- Interdisciplinary_fields
- Social_sciences
- Society
- Thought
- Business
- Production_and_manufacturing
- Visual_Arts
- Language
- Information
- Media_Technology
- Personal_life
- Manufactured_goods
- Applied_sciences
- Structure
- Scientific_disciplines
- Tutorial
- How-to
- Public_utility
- Science
- Industries
- Industry
- Technology
- Categories_requiring_diffusion
- Culture
- Cultures
- Non-transitive_categories
- Cultural_history
- Engineering
- Arts
- Art_media
- Engineering_disciplines
- Formal_sciences
- Equipment
- Fundamental_categories
- Human_behavior
- Human_communication
- Humans

In addition, all resources that contained either *_by_*, *_of_* or *maintenance* were also automatically removed.

Appendix B

Disambiguation example

The example in this appendix illustrates what happens when two clusters represent disjoint domains, i.e. multiple topics, multi-disciplinarity. More specifically, it lists the full output of the disambiguation algorithm for a multi-disciplinary situation using four terms: `[ruby, python, mona, leonardo]`, i.e. the domains of computing and art.

During disambiguation, the first cluster, with the highest average degree, disambiguates the first two terms. However, the second cluster contains a concept that conflicts with a concept assignment already made using the first cluster. The second cluster is therefore disregarded in favor of another cluster that does not contain concepts that conflict with already disambiguated terms.

The raw output of the disambiguation algorithm applied to the terms: *python, ruby, mona, leonardo*:

```
[python, ruby, mona, leonardo]
python -> Python_%28programming_language%29
python -> Python_II
python -> Python_of_Aenus
python -> Python_Lowracer
python -> Python_%28Efteling%29
python -> Python_%28genus%29
python -> Python_missile
python -> Colt_Python
python -> Armstrong_Siddeley_Python
python -> Python
python -> Python_%28film%29
python -> Python_of_Catana
python -> Python_%28roller_coaster%29
python -> Monty_Python
python -> Python_Automobile
python -> Python_%28mythology%29
python -> Pythonidae
python -> Python_of_Byzantium
concepts: 18

ruby -> Ruby%2C_My_Dear
ruby -> Ruby_%28hardware_description_language%29
ruby -> Ruby_%28programming_language%29
ruby -> Lloyd_Ruby
ruby -> Ruby_Dandridge
ruby -> According_to_Jim
ruby -> The_Tribe_%28TV_series%29
ruby -> Pok%C3%A9mon_Ruby
ruby -> Ruby_%28Andrews_novel%29
ruby -> Sam_Ruby
ruby -> Ruby_Gloom
ruby -> Ruby_%28Egyptian_singer%29
ruby -> Ruby_%28Pok%C3%A9mon%29
ruby -> Ruby_%28band%29
```

```
ruby -> Ruby_Creek_%28disambiguation%29
ruby -> Cockney_Rhyming_Slang
ruby -> Curry
ruby -> Ruby_%28annotation_markup%29
ruby -> Ruby_the_Galactic_Gumshoe
ruby -> Ruby_%28TV_series%29
ruby -> Ruby_Crescent
ruby -> Ruby_Baby
ruby -> Ruby_Records
ruby -> Ruby_%28The_Land_Before_Time%29
ruby -> Ruby%2C_Arizona
ruby -> Ruby_%281970s_band%29
ruby -> Ruby_Rose
ruby -> Ruby_character%23Ruby_markup
ruby -> Ruby_%28Supernatural%29
ruby -> List_of_The_Tribe_characters%23Ruby
ruby -> Ruby_%28elephant%29
ruby -> Ruby_Wax
ruby -> Tom_Fogerty
ruby -> Ruby_Lin
ruby -> Ruby_%28given_name%29
ruby -> Ruby%2C_the_digital_superstar
ruby -> Style_Network
ruby -> Ruby_Creek
ruby -> Ruby%2C_Don%27t_Take_Your_Love_to_Town
ruby -> Jack_Ruby
ruby -> Ruby_pistol
ruby -> London
ruby -> Ruby_Creek%2C_British_Columbia
ruby -> Ruby%2C_Alaska
ruby -> Ruby_%28V._C._Andrews_novel%29
ruby -> Ruby_Dee
ruby -> Ruby_Dome
ruby -> Ruby_laser
ruby -> Ruby_character
ruby -> Ruby_%28film_1977%29
ruby -> Ruby_%28film%29
ruby -> Dear_Mr._Wonderful
ruby -> Rubi_%28disambiguation%29
ruby -> Ruby_Mountains
ruby -> Ruby_Gentry
ruby -> Rubies_of_Eventide
ruby -> Ruby_%28tv_show%29
ruby -> Ruby_%28song%29
ruby -> Ruby_Tuesday
ruby -> Ruby_Murray
ruby -> Ruby_Trollman
ruby -> Ruby_Creek_%28Canada%29
ruby -> Ruby_Soho
ruby -> Ruby_Creek_2
ruby -> Karine_Ruby
ruby -> Ray_Charles
ruby -> Ruby_Mountain
ruby -> Ruby_MRI
ruby -> Ruby_Walsh
ruby -> Ruby
concepts: 70

mona -> Mona_%28opera%29
mona -> Mona%2C_Puerto_Rico
mona -> Mona_%28deity%29
mona -> Mona_%28song%29
mona -> Mona
mona -> Mona%2C_Utah
mona -> Mona_%28operating_system%29
mona -> Mona_%28elephant%29
mona -> Mona_Simpson_%28The_Simpsons%29
mona -> Mona_%28name%29
mona -> Mona_%28wrestler%29
mona -> Mona_the_Virgin_Nymph
mona -> Mona_the_Vampire
mona -> Mona_%28WarioWare%29
mona -> Mona_Lisa
mona -> Mona_%28film%29
mona -> Szl:Mona
mona -> Mona%2C_Jamaica
mona -> Mona%2C_Anglesey
mona -> Mona_Mahmudnizhad
mona -> Lifeboat_Mona
concepts: 21

leonardo -> Leonardo_dos_Santos_Silva
leonardo -> Leonardo_Andr%C3%A9_Pimenta_Faria
leonardo -> Leonardo_%28robot%29
leonardo -> Leonardo_Ara%C3%BAjo
leonardo -> Leonardo_Cardona
leonardo -> Leonardo_da_Vinci
leonardo -> Leonardo_%28dinosaur%29
leonardo -> Leonardo_Rodriguez_Pereira
leonardo -> Leonardo_Renan_Sim%C3%B5es_de_Lacerda
leonardo -> San_Leonardo%2C_Nueva_Ecija
```

```
leonardo -> Leandro_e_Leonardo
leonardo -> Leonardo_Fibonacci
leonardo -> Leonardo_DiCaprio
leonardo -> Leonardo_Jos%C3%A9_Aparecido_Moura
leonardo -> King_Leonardo_and_His_Short_Subjects
leonardo -> Leonardo_%28Teenage_Mutant_Ninja_Turtles%29
leonardo -> Leonardo_Santiago
leonardo -> San_Leonardo%2C_Italy
leonardo -> Leonardo_%28TV_channel%29
leonardo -> Leonardo_Acropolis
leonardo -> Leonardo_Leonardo
leonardo -> Leonardo_%28St._Louis%2C_Missouri%29
leonardo -> Leonardo_Narv%C3%A1ez
leonardo -> Leonardo
leonardo -> Leonardo_Farkas
leonardo -> Leonardo%2C_New_Jersey
leonardo -> Leonardo_the_Musical:_A_Portrait_of_Love
leonardo -> Leonardo_Vetra
leonardo -> Leonardo_Bruni
leonardo -> Leonardo_da_Vinci_%28European_Union_programme%29
leonardo -> Leonardo_Louren%C3%A7o_Bastos
leonardo -> Leonardo_Journal
leonardo -> Leonardo_Sottani
concepts: 33

pagelink found: Python_%28programming_language%29 - Ruby_%28programming_language%29
pagelink found: Monty_Python - London
pagelink found: Ruby_%28programming_language%29 - Python_%28programming_language%29
pagelink found: Ruby_MRI - Python_%28programming_language%29
pagelink found: Mona_Lisa - Leonardo_da_Vinci
pagelink found: Leonardo_da_Vinci - London
pagelink found: Leonardo_da_Vinci - Mona_Lisa
pagelink found: Leonardo_the_Musical:_A_Portrait_of_Love - London
pagelink found: Leonardo_the_Musical:_A_Portrait_of_Love - Mona_Lisa
category found: Ruby_Baby <- Category:The_Beach_Boys_songs -> Mona_%28song%29
category found: Ruby_Lin <- Category:Living_people -> Leonardo_Sottani
category found: Ruby_Rose <- Category:Living_people -> Leonardo_Narv%C3%A1ez
category found: Ruby_Lin <- Category:Living_people -> Leonardo_Cardona
category found: Ruby_Lin <- Category:Living_people -> Leonardo_Narv%C3%A1ez
category found: Sam_Ruby <- Category:Living_people -> Leonardo_DiCaprio
category found: Ruby_Wax <- Category:Living_people -> Leonardo_Renan_Sim%C3%B5es_de_Lacerda
category found: Sam_Ruby <- Category:Living_people -> Leonardo_Santiago
category found: Ruby_Wax <- Category:Living_people -> Leonardo_dos_Santos_Silva
category found: Ruby_Wax <- Category:Living_people -> Leonardo_Andr%C3%A9_Pimenta_Faria
category found: Ruby_Wax <- Category:Living_people -> Leonardo_Farkas
category found: Ruby_Dee <- Category:Living_people -> Leonardo_Cardona
category found: Ruby_Walsh <- Category:Living_people -> Leonardo_Farkas
category found: Ruby_Rose <- Category:Living_people -> Leonardo_Cardona
category found: Ruby_Dee <- Category:Living_people -> Leonardo_Santiago
category found: Ruby_Dee <- Category:Living_people -> Leonardo_Ara%C3%BAjo
category found: Ruby_Rose <- Category:Living_people -> Leonardo_Rodriguez_Pereira
category found: Ruby_%28Egyptian_singer%29 <- Category:Living_people -> Leonardo_Renan_Sim%C3%B5es_de_Lacerda
category found: Sam_Ruby <- Category:Living_people -> Leonardo_Narv%C3%A1ez
category found: Ruby_%28Egyptian_singer%29 <- Category:Living_people -> Leonardo_Santiago
category found: Ruby_%28Egyptian_singer%29 <- Category:Living_people -> Leonardo_Jos%C3%A9_Aparecido_Moura
category found: Ruby_Dee <- Category:Living_people -> Leonardo_dos_Santos_Silva
category found: Sam_Ruby <- Category:Living_people -> Leonardo_Cardona
category found: Ruby_Wax <- Category:Living_people -> Leonardo_Jos%C3%A9_Aparecido_Moura
category found: Ruby_Lin <- Category:Living_people -> Leonardo_Renan_Sim%C3%B5es_de_Lacerda
category found: Ruby_Rose <- Category:Living_people -> Leonardo_Farkas
category found: Ruby_Walsh <- Category:Living_people -> Leonardo_Narv%C3%A1ez
category found: Sam_Ruby <- Category:Living_people -> Leonardo_dos_Santos_Silva
category found: Ruby_Dee <- Category:Living_people -> Leonardo_Rodriguez_Pereira
category found: Ruby_Walsh <- Category:Living_people -> Leonardo_Jos%C3%A9_Aparecido_Moura
category found: Ruby_Lin <- Category:Living_people -> Leonardo_Andr%C3%A9_Pimenta_Faria
category found: Ruby_Walsh <- Category:Living_people -> Leonardo_dos_Santos_Silva
category found: Python_%28programming_language%29 <- Category:Free_compilers_and_interpreters -> Ruby_MRI
category found: Ruby_Dee <- Category:Living_people -> Leonardo_Andr%C3%A9_Pimenta_Faria
category found: Python_%28film%29 <- Category:20th_Century_Fox_films -> Ruby_Gentry
category found: Sam_Ruby <- Category:Living_people -> Leonardo_Ara%C3%BAjo
category found: Ruby_Walsh <- Category:Living_people -> Leonardo_Rodriguez_Pereira
category found: Ruby_Lin <- Category:Living_people -> Leonardo_Rodriguez_Pereira
category found: Ruby_%28Egyptian_singer%29 <- Category:Living_people -> Leonardo_Louren%C3%A7o_Bastos
category found: Ruby_Dee <- Category:Living_people -> Leonardo_Renan_Sim%C3%B5es_de_Lacerda
category found: Ruby_Wax <- Category:Living_people -> Leonardo_DiCaprio
category found: Ruby_Rose <- Category:Living_people -> Leonardo_Renan_Sim%C3%B5es_de_Lacerda
category found: Ruby_Wax <- Category:Living_people -> Leonardo_Santiago
category found: Ruby_Wax <- Category:Living_people -> Leonardo_Louren%C3%A7o_Bastos
category found: Sam_Ruby <- Category:Living_people -> Leonardo_Rodriguez_Pereira
category found: Sam_Ruby <- Category:Living_people -> Leonardo_Farkas
category found: Ruby_Walsh <- Category:Living_people -> Leonardo_Renan_Sim%C3%B5es_de_Lacerda
category found: Ruby_Lin <- Category:Living_people -> Leonardo_Ara%C3%BAjo
category found: Ruby_Walsh <- Category:Living_people -> Leonardo_Sottani
category found: Ruby_%28Supernatural%29 <- Category:Fictional_characters_with_superhuman_strength ->
    Leonardo_%28Teenage_Mutant_Ninja_Turtles%29
category found: Ruby_Dee <- Category:American_film_actors -> Leonardo_DiCaprio
category found: Ruby_Gloom <- Category:Canadian_children%27s_television_series -> Mona_the_Vampire
category found: Ruby_Dee <- Category:Living_people -> Leonardo_Narv%C3%A1ez
category found: Ruby_Walsh <- Category:Living_people -> Leonardo_Andr%C3%A9_Pimenta_Faria
category found: Ruby_Dee <- Category:Living_people -> Leonardo_Farkas
category found: Ruby_Rose <- Category:Living_people -> Leonardo_DiCaprio
```

```
category found: Sam_Ruby <- Category:Living_people -> Leonardo_Andr%C3%A9_Pimenta_Faria
category found: Ruby_Dee <- Category:Living_people -> Leonardo_Louren%C3%A7o_Bastos
category found: Ruby_Dee <- Category:Living_people -> Leonardo_Sottani
category found: Ruby_Rose <- Category:Living_people -> Leonardo_dos_Santos_Silva
category found: Ruby_Lin <- Category:Living_people -> Leonardo_Farkas
category found: Ruby_Lin <- Category:Living_people -> Leonardo_Louren%C3%A7o_Bastos
category found: Ruby_Dee <- Category:Living_people -> Leonardo_Jos%C3%A9_Aparecido_Moura
category found: Ruby_%28elephant%29 <- Category:Famous_elephants -> Mona_%28elephant%29
category found: Sam_Ruby <- Category:Living_people -> Leonardo_Sottani
category found: Ruby_%28Egyptian_singer%29 <- Category:Living_people -> Leonardo_Rodriguez_Pereira
category found: Ruby_%28Egyptian_singer%29 <- Category:Living_people -> Leonardo_Farkas
category found: Ruby_Wax <- Category:Living_people -> Leonardo_Sottani
category found: Ruby_Walsh <- Category:Living_people -> Leonardo_Louren%C3%A7o_Bastos
category found: Ruby_Rose <- Category:Living_people -> Leonardo_Andr%C3%A9_Pimenta_Faria
category found: Python_%28film%29 <- Category:English-language_films -> Dear_Mr._Wonderful
category found: Ruby_Wax <- Category:Living_people -> Leonardo_Narv%C3%A1ez
category found: Ruby_Lin <- Category:Living_people -> Leonardo_Jos%C3%A9_Aparecido_Moura
category found: Ruby_Lin <- Category:Living_people -> Leonardo_DiCaprio
category found: Ruby_%28Egyptian_singer%29 <- Category:Living_people -> Leonardo_Andr%C3%A9_Pimenta_Faria
category found: Ruby_%28given_name%29 <- Category:Feminine_given_names -> Mona_%28name%29
category found: Sam_Ruby <- Category:Living_people -> Leonardo_Renan_Sim%C3%B5es_de_Lacerda
category found: Ruby_Lin <- Category:Living_people -> Leonardo_Santiago
category found: Ruby_Walsh <- Category:Living_people -> Leonardo_Cardona
category found: Python_II <- Category:English-language_films -> Dear_Mr._Wonderful
category found: Sam_Ruby <- Category:Living_people -> Leonardo_Louren%C3%A7o_Bastos
category found: Ruby_Rose <- Category:1986_births -> Leonardo_Jos%C3%A9_Aparecido_Moura
category found: Ruby_Walsh <- Category:Living_people -> Leonardo_DiCaprio
category found: Ruby_%28Egyptian_singer%29 <- Category:Living_people -> Leonardo_Sottani
category found: Ruby_Walsh <- Category:Living_people -> Leonardo_Santiago
category found: Ruby_%28Egyptian_singer%29 <- Category:Living_people -> Leonardo_DiCaprio
category found: Sam_Ruby <- Category:Living_people -> Leonardo_Jos%C3%A9_Aparecido_Moura
category found: Ruby_Rose <- Category:Living_people -> Leonardo_Santiago
category found: Python_%28programming_language%29 <- Category:Object-oriented_programming_languages ->
    Ruby_%28programming_language%29
category found: Ruby_Wax <- Category:Living_people -> Leonardo_Rodriguez_Pereira
category found: Ruby_Rose <- Category:Living_people -> Leonardo_Louren%C3%A7o_Bastos
category found: Ruby_Wax <- Category:Living_people -> Leonardo_Cardona
category found: Ruby_%28Egyptian_singer%29 <- Category:Living_people -> Leonardo_dos_Santos_Silva
category found: Ruby_Wax <- Category:Living_people -> Leonardo_Ara%C3%BAjo
category found: Ruby_%28Egyptian_singer%29 <- Category:Living_people -> Leonardo_Ara%C3%BAjo
category found: Ruby_Walsh <- Category:Living_people -> Leonardo_Ara%C3%BAjo
category found: Ruby_Lin <- Category:1976_births -> Leonardo_dos_Santos_Silva
category found: Ruby_%28Egyptian_singer%29 <- Category:Living_people -> Leonardo_Cardona
category found: Ruby_Rose <- Category:Living_people -> Leonardo_Sottani
category found: Ruby_Rose <- Category:Living_people -> Leonardo_Ara%C3%BAjo
category found: Ruby_%28Egyptian_singer%29 <- Category:Living_people -> Leonardo_Narv%C3%A1ez
category found: Ruby_Dandridge <- Category:American_film_actors -> Leonardo_DiCaprio

Number of edges in graph: 55
Unpruned cluster sized 6: [Python_II,
        Ruby_Gentry, Category:English-language_films, Python_%28film%29,
        Category:20th_Century_Fox_films, Dear_Mr._Wonderful]

Unpruned cluster sized 3: [Mona_%28name%29, Category:Feminine_given_names, Ruby_%28given_name%29]

Unpruned cluster sized 3: [Ruby_%28elephant%29, Category:Famous_elephants, Mona_%28elephant%29]

Unpruned cluster sized 25: [Leonardo_dos_Santos_Silva, Leonardo_Andr%C3%A9_Pimenta_Faria, Category:1986_births,
         Ruby_Dandridge, Leonardo_Ara%C3%BAjo, Leonardo_Cardona, Leonardo_Rodriguez_Pereira,
         Leonardo_Renan_Sim%C3%B5es_de_Lacerda, Sam_Ruby, Ruby_Dee, Ruby_%28Egyptian_singer%29,
         Leonardo_DiCaprio, Leonardo_Jos%C3%A9_Aparecido_Moura, Leonardo_Santiago,
         Category:1976_births, Leonardo_Narv%C3%A1ez, Leonardo_Farkas, Ruby_Rose,
         Leonardo_Louren%C3%A7o_Bastos, Category:American_film_actors, Category:Living_people, Ruby_Wax,
         Ruby_Lin, Ruby_Walsh, Leonardo_Sottani]

Unpruned cluster sized 4: [London, Mona_Lisa, Leonardo_da_Vinci, Leonardo_the_Musical:_A_Portrait_of_Love]

Unpruned cluster sized 3: [Ruby_Gloom, Mona_the_Vampire, Category:Canadian_children%27s_television_series]

Unpruned cluster sized 3: [Ruby_Baby, Mona_%28song%29, Category:The_Beach_Boys_songs]

Unpruned cluster sized 5: [Python_%28programming_language%29, Category:Object-oriented_programming_languages,
        Ruby_%28programming_language%29, Category:Free_compilers_and_interpreters, Ruby_MRI]

Unpruned cluster sized 3: [Ruby_%28Supernatural%29, Category:Fictional_characters_with_superhuman_strength,
        Leonardo_%28Teenage_Mutant_Ninja_Turtles%29]

Clusters after pruning....

Average cluster degree = 2.8
Python_%28programming_language%29,
Ruby_%28programming_language%29

Average cluster degree = 2.75
London,
Mona_Lisa
Leonardo_da_Vinci

Average cluster degree = 2.16
Leonardo_dos_Santos_Silva,
Ruby_Dee
```

```
Average cluster degree = 1.6666666666666667
Ruby_Gentry
Python_%28film%29

Average cluster degree = 1.3333333333333333
Mona_%28name%29
Ruby_%28given_name%29

Average cluster degree = 1.3333333333333333
Ruby_%28elephant%29
Mona_%28elephant%29

Average cluster degree = 1.3333333333333333
Ruby_Gloom
Mona_the_Vampire

Average cluster degree = 1.3333333333333333
Ruby_Baby
Mona_%28song%29

Average cluster degree = 1.3333333333333333
Ruby_%28Supernatural%29,
Leonardo_%28Teenage_Mutant_Ninja_Turtles%29

Average cluster degree = 1.0
Monty_Python,


TRYING A CLUSTER OF SIZE: 5( edgecount = 7)
Adding a new mapping: python -> Python_%28programming_language%29
Adding a new mapping: ruby -> Ruby_%28programming_language%29
TRYING A CLUSTER OF SIZE: 4( edgecount = 5)
Adding a new mapping: mona -> Mona_Lisa
Adding a new mapping: leonardo -> Leonardo_da_Vinci

Final mapping:
python (18) -> Python_%28programming_language%29
ruby (70) -> Ruby_%28programming_language%29
mona (21) -> Mona_Lisa
leonardo (33) -> Leonardo_da_Vinci
```

Appendix C

Topic Models - Mixed models analysis

1 Undergraduate model

	# Levels	Covariance Structure	# Parameters	Subject Variables	# Subjects
Fixed Effects	19		19		
Repeated Effects	19	Unstructured	190	student	61
Total	38		209		

Table C.1: Undergraduate model dimensions

Measure	Value
-2 Restricted Log Likelihood	-1097.031
Akaike's Information Criterion (AIC)	-717.031
Hurvich and Tsai's Criterion (AICC)	-568.909
Bozdogan's Criterion (CAIC)	332.446
Schwarz's Bayesian Criterion (BIC)	142.446

Table C.2: Undergraduate model fit

Numerator df	Denominator df	F	Sig.
19	22.590	4895.870	.000

Table C.3: Undergraduate model: Type III Tests of Fixed Effects

Parameter	Estimate	Std. Error	Wald Z	Sig.	Correlation	95% Confidence Interval	
						Lower bound	Upper bound
Average grade	-.016217	.011661	-1.391	.164	-.193	-.039073	.006638
# ratings	.030086	.007754	3.880	.000	.554	.014888	.045285
# absent wiki	.008729	.010447	.836	.403	.157	-.011746	.029204
# absent lda	.007098	.003457	2.053	.040	.303	.000322	.013874
# absent medlda course	.005359	.003580	1.497	.134	.182	-.001658	.012377
# absent medlda user	.014305	.004357	3.283	.001	.374	.005765	.022845
grade wiki	-.009826	.009001	-1.092	.275	-.168	-.027468	.007816
grade lda	.006903	.005894	1.171	.242	.203	-.004650	.018455
grade medlda course	-.007486	.005100	-1.468	.142	-.230	-.017481	.002510
grade medlda user	.005262	.004969	1.059	.290	.155	-.004477	.015001
quality wiki	-.005951	.007483	-.795	.426	-.131	-.020617	.008714
quality lda	.012935	.004809	2.690	.007	.487	.003510	.022360
quality medlda course	.004481	.003695	1.213	.225	.168	-.002761	.011722
quality medlda user	.018089	.005330	3.394	.001	.468	.007642	.028535
# words wiki	.004010	.008653	.463	.643	.063	-.012950	.020970
# words lda	.009092	.003599	2.526	.012	.418	.002038	.016146
# words medlda course	.008769	.003604	2.433	.015	.300	.001705	.015832
# words medlda user	.014365	.003488	4.119	.000	.459	.007529	.021200

Table C.4: Undergraduate model: covariance parameters and correlations of variables with the average exam grade.

2 Graduate model

	# Levels	Covariance Structure	# Parameters	Subject Variables	# Subjects
Fixed Effects	19		19		
Repeated Effects	19	Unstructured	190	student	44
Total	38		209		

Table C.5: Graduate model dimensions

Measure	Value
-2 Restricted Log Likelihood	-1026.229
Akaike's Information Criterion (AIC)	-646.229
Hurvich and Tsai's Criterion (AICC)	-502.506
Bozdogan's Criterion (CAIC)	407.387
Schwarz's Bayesian Criterion (BIC)	217.387

Table C.6: Graduate model fit

Numerator df	Denominator df	F	Sig.
19	29.757	5563.779	.000

Table C.7: Graduate model: Type III Tests of Fixed Effects

Parameter	Estimate	Std. Error	Wald Z	Sig.	Correlation	95% Confidence Interval	
						Lower bound	Upper bound
Average grade	.010213	.008602	1.187	.235	.183	-.006647	.027073
# ratings	-.033520	.020290	-1.652	.099	-.256	-.073289	.006248
# absent wiki	.004757	.003207	1.483	.138	.227	-.001528	.011041
# absent lda	.005032	.002532	1.988	.047	.310	.000007	.009994
# absent medlda course	.006473	.002988	2.166	.030	.342	.000616	.012329
# absent medlda user	.015873	.004680	3.392	.001	.592	.006700	.025045
grade wiki	-.004387	.004698	-.934	.350	-.140	-.013594	.004821
grade lda	-.005177	.003628	-1.427	.154	-.217	-.012288	.001933
grade medlda course	-.000469	.003886	-.121	.904	-.018	-.008085	.007148
grade medlda user	.006195	.005414	1.144	.252	.171	-.004415	.016806
quality wiki	-.003013	.002519	-1.196	.232	-.182	-.007949	.001924
quality lda	.006244	.003656	1.708	.088	.264	-.000922	.013410
quality medlda course	.010088	.004338	2.325	.020	.373	.001586	.018591
quality medlda user	.013006	.006076	2.141	.032	.335	.001098	.024914
# words wiki	.001731	.004130	0.419	.675	.063	-.006363	.009825
# words lda	.003074	.005174	0.594	.552	.089	-.007066	.013214
# words medlda course	.004859	.005447	0.892	.372	.134	-.005818	.015535
# words medlda user	.006195	.005216	1.188	.235	.179	-.004029	.016418

Table C.8: Graduate model: covariance parameters and correlations of variables with the average exam grade.

Index

Een samenvatting in het Nederlands

Volgens het sociaal constructivisme is leren en jezelf ontwikkelen een sociaal fenomeen. Sociale media worden in toenemende mate gebruikt om informatie en kennis uit te wisselen binnen online kennisgemeenschappen. Echter, het zoeken naar nieuwe informatie om te leren is afhankelijk van de individuele woordenschat. De kwaliteit van online zoekresultaten is gevoelig voor subtiele lexicale verschillen in de gekozen sleutelwoorden. Door een leerder gekozen zoektermen, kunnen worden beschouwd als een indicatie van het kennisniveau. Verschillen tussen het vocabulair van een leerder en dat van de bijbehorende kennisgemeenschap kan de toegang tot leermateriaal beperken.

Deze problemen tussen leerders en kennisgemeenschappen kunnen worden aangepakt door gebruik te maken van een domeinontologie; een formalisatie van een stukje domeinkennis. Deze stelt een leerder in staat om een goed beeld te krijgen van een kennisgebied zonder dat daarbij wordt verondersteld dat het jargon al bekend is. Een kenmerkend verschil tussen ontologieën en sociale media is dat de informatie in een ontologie zeer gestructureerd en van hoge kwaliteit is, terwijl sociale media juist weinig tot geen structuur opleggen. Ook moet een domeinontologie nagenoeg altijd handmatig worden gebouwd en bijgewerkt. Dit heeft in de praktijk tot gevolg dat ontologieën achterlopen op de laatste ontwikkelingen en daardoor minder relevant zijn. Sociale media hebben dit probleem niet en zijn door hun eenvoud en schaal een uitstekende bron van nieuwe relevante informatie.

Dit proefschrift beschrijft daarom een automatische methode die relevante informatie uit sociale media extraheert en deze gebruikt om een domeinontologie bij te werken. Automatische disambiguatie (het koppelen van woorden aan hun betekenis) met behulp van semantische netwerken, speelt in dit proces een grote rol. Een domeinontologie die is bijgewerkt met lexicale, conceptuele en relationele gegevens op basis van informatie uit sociale media, kan gebruikt worden om de kwaliteit van zoekresultaten met behulp van semantisch zoeken te verbeteren. De verrijkte domeinontologie is daarmee in staat om te bemiddelen tussen het kennisniveau en bijbehorend vocabulair van een leerder en dat van een kennisgemeenschap op conceptueel niveau.

Lexicale verschillen kunnen, ondanks hun voorgenoemde negatieve effect op informatieontsluiting, ook waardevol zijn in een leercontext. Ze geven namelijk een beeld van het vermogen van een leerder om te communiceren over domeinspecifieke kennis. Het vermogen of onvermogen om te communiceren in het vocabulair van een kennisgemeenschap blijkt een goede indicatie te zijn van de daadwerkelijk opgedane kennis. Dit proefschrift beschrijft een methode die het kennisniveau van een leerder volautomatisch kan inschatten op basis van een kleine collectie leermateriaal, computationele topic modellering en slechts 30 à 40 woorden van een leerder om een topic model samen te vatten.

Dit proefschrift beschrijft een innovatieve methodologie om domeinontologieën automatisch te verrijken met lexicale en conceptuele informatie uit sociale media. Ook introduceert het een domeinonafhankelijk disambiguatie-algoritme dat werkt op basis van semigestructureerde gegevens uit zogenoemde 'reference repositories'. Verder stelt het een nieuwe methodologie voor semantisch zoeken voor die niet beperkt is tot semantisch geannoteerde corpora. Met de voorgestelde methode, een combinatie van een verrijkte ontologie en automatische disambiguatie, kunnen sociale media volautomatisch doorzocht worden, waardoor de kwaliteit van zoekresultaten aanzienlijk toeneemt. Dit werk laat ook zien dat de interpretatie van topic modellen door mensen een indicator kan zijn voor hun kennis op een bepaald domein.

Curriculum Vitae

Thomas Markus was born in Zeist on the 3rd of December 1983. He studied Cognitive Artificial Intelligence at Utrecht University. This was seamlessly followed by a research position in the EU FP7 Language Technology for Lifelong Learning project. After this project he started working on his PhD interspersed with a language technology web services project and lecturing and organizing a course on logic programming. The results of some of the research performed during this period at Utrecht University is reflected in this book.