What's in an Index: Extracting Domain-specific Knowledge Graphs from Textbooks

Isaac Alpizar-Chacon
Utrecht University
Utrecht, The Netherlands

Sergey Sosnovsky
Utrecht University
Utrecht, The Netherlands

ABSTRACT

A typical index at the end of a textbook contains a manually-provided vocabulary of terms related to the content of the textbook. In this paper, we extend our previous work on extraction of knowledge models from digital textbooks. We are taking a more critical look at the content of a textbook index and present a mechanism for classifying index terms according to their domain specificity: a *core domain* concept, an *in-domain* concept, a concept from a *related domain*, and a concept from a *foreign domain*. We link the extracted models to DBpedia and leverage the aggregated linguistic and structural information from textbooks and DBpedia to construct and prune the domain-specific knowledge graphs. The evaluation experiments demonstrate (1) the ability of the approach to identify (with high accuracy) different levels of domain specificity for automatically extracted concepts, (2) its cross-domain robustness, and (3) the added value of the domain specificity information. These results clearly indicate the improved quality of the refined knowledge graphs and widen their potential applicability.

CCS CONCEPTS

• **Information systems** → **Document representation**; **Web searching and information discovery**; **Clustering and classification**; Information extraction; • **Applied computing** → **Document analysis**.

KEYWORDS

Knowledge modeling, Knowledge extraction, Knowledge graphs, Textbooks, Textbook indices, DBpedia, Domain specificity

ACM Reference Format:
Isaac Alpizar-Chacon and Sergey Sosnovsky. 2022. What's in an Index: Extracting Domain-specific Knowledge Graphs from Textbooks. In *Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3485447.3512140

This work was partially supported by the Ministry of Science, Innovation, Technology and Telecommunications of Costa Rica (grant 2-1-4-17-1-021).

WWW '22, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9096-5/22/04…$15.00
https://doi.org/10.1145/3485447.3512140

1 INTRODUCTION

Back-of-the-book indices are collections of terms that can help textbook readers in several ways. As a navigation tool, an index provides readers with hand-crafted shortcuts from a target notion to a place in the textbook that explains it or elaborates on it. As an information retrieval tool, it supports meaningful annotation of the textbook content with manually selected keywords. Finally, as a knowledge organization method, an index is a collection of important domain terms curated by an expert. We underline the fact that index terms and links between them and textbook pages are created manually. There have been a few research attempts to develop methods and tools for automated index construction [11, 61]; however, their results are far from reliable. More recently developed approaches on terminology extraction either require very large corpora [65] or utilize supervised methods [63]. Finally, existing commercial indexing software such as CINDEX[1] or Index Manager[2] can help automate some steps of index creation (such as creating a word list, or alphabetizing of an index), but still require manual supervision. At the end of the day, index development is a task that has to be done manually and cannot be done just by anyone. Multiple books (e.g., [6, 9, 30]) and guidelines (e.g., [1, 2]) are written to direct the index creation process. Dedicated associations (e.g., The American Society for Indexing[3] and The Society of Indexers[4]) develop and disseminate book-indexing methods and practices. Methodologies for indexing stipulate index length and style, good and bad candidates for index terms, how to maintain consistency when creating hierarchical indices, interrelationships among terms, etc. Usually, it is either a textbook author or a dedicated human indexer who creates the index. As a result, a typical textbook index is not just a collection of words but also a reference model produced by an expert according to a predefined set of rules. Each entry in this model is accompanied with one or more links to relevant textbook pages. Moreover, these pages do not simply mention index entries but provide meaningful references by either introducing corresponding terms or elaborating them.

In our recent work [4], we have described a procedure for automated extraction of knowledge models from PDF textbooks, based on their structure, formatting and organization patterns. We have paid special attention to the index sections of the processed textbooks as the source of fine-grained domain terminology and text annotations. We have been able to automatically link index terms to their corresponding resources in DBpedia[5], thus enriching our textbook models with additional semantic information and connecting them to the open linked data cloud [3].

[1] http://www.indexres.com
[2] http://index-manager.net
[3] https://www.asindexing.org/
[4] https://www.indexers.org.uk/
[5] http://dbpedia.org

At the same time, even a surface analysis of a typical textbook index can show that not all index entries are equally representative of the domain of the textbook [6]. From the epistemological perspective, an index of any document reflects not only the expertise and efforts of its creator, but also the needs of a group of users for whom the index is created, and the task that these users are engaged in [27]. Hence, on one hand, we can never expect a textbook index to represent a fully objective and neutral model with consistent granularity of cohesive terms covering the domain of the textbook exclusively, completely and unambiguously. At the same time, we can expect that the purpose of an index follows the purpose of the textbook itself, namely, to present important notions that help readers better understand a certain subject. Many of the index entries will introduce the core concepts from the target domain, yet there will also be terms included in the index to represent the unique view of the textbooks author, make connections to relevant domains, and present potential use cases, applications and examples. In other words, even a good index is likely to include a large number of entries that refer to concepts that either mildly relevant to the target domain, or not relevant at all. For example, in a statistics textbook, the terms *mean* and *hypotheses testing* will belong to the core of the main subject. There will also be other, more niche statistical terms, such as *five-number summary* and *cross-validation*. Terms like *factorial* and *De Morgan's laws* are likely to be present as well, yet they are associated with domains related to statistics: mathematics and set theory, respectively. Finally, terms like *Euro coin* and *Bovine Spongiform Encephalopathy* are from entirely different domains, included to enrich the textbook with examples.

This means that knowledge models that we extract from textbooks contain concepts with low domain specificity. This can seriously reduce the value of such models as intelligent services built on top of them would have hard time distinguishing between relevant and irrelevant domain knowledge. For example, an adaptive learning environment [26] using such a model could misjudge the importance of a certain concept and mistakenly guide students to irrelevant educational material.

Since manual assessment of domain specificity of large knowledge graphs is a time-consuming and complex task [40], we have focused on developing a method for *automated* analysis of index terms and identification of their relevance to the domain of a textbook. We integrate information extracted from textbooks and DBpedia. Textbooks supply index terms and referenced text fragments, while DBpedia provides structural (categories and links) and textual (abstracts) information associated with the resources linked to the index terms. The contributions of this paper are two-fold: (1) an approach for identifying the domain specificity relation of index terms; (2) an evaluation of the accuracy and applicability of the proposed approach.

2 PRELIMINARIES

2.1 Domain Specificity

Generally speaking, domain specificity classification refers to the task of assigning to a term the label *used* or *not used* concerning a domain D of interest [32]. In this paper, we extend the traditional classification into a set L of four domain specificity labels to annotate the index terms. Each label $l \in L$ is one of the following:

- *core-domain*: key index terms that represent the most important and frequently used concepts in D;
- *in-domain*: additional index terms that belong to D;
- *related-domain*: index terms from domains related to D;
- *out-of-domain*: index terms not related to D (often used for pedagogical reasons, e.g., examples, use-cases, summaries).

2.2 DBpedia

DBpedia is a knowledge graph [22] extracted from Wikipedia [7]. Each Wikipedia entry/page is represented as a DBpedia resource. Currently, its English version describes over 6 million resources, uniquely identified by URIs[7]. Knowledge in DBpedia can be queried through its SPARQL endpoint or downloaded as a full RDF model. Each DBpedia resource has an abstract, a category, and a set of links to other resources. Abstracts are extracted from the texts of Wikipedia pages preceding tables of contents. Categories are special kind of resources used to classify and group regular resources on similar subjects. Each resource has one or more categories associated, and each category has a set of sub-categories and super-categories. We use the symbols $\subseteq^c$ and $\supseteq^c$ to indicate that a category is a sub-category or a super-category, respectively. For example, *dbc:Statistics* $\subseteq^c$ *dbc:Probability_and_statistics* and *dbc:Statistics* $\supseteq^c$ *dbc:Applied_statistics*. Also, when $\subseteq^c$ and $\supseteq^c$ are used between a resource and a category, they show the direct category of the resource. For example, *dbr:Mean* $\subseteq^c$ *dbc:Means*. This allows navigate the non-strict hierarchy of categories using a number of hops (n-hops) to connect categories and resources. Finally, each resource is associated with other resources using the hyperlinks between the corresponding Wikipedia pages of the resources. Figure 1 shows five DBpedia resources and how they are connected to *dbc:Statistics* using the categorization system.

2.3 Knowledge Model Extraction and Linking to DBpedia

As mentioned, we have previously developed a method for the automated extraction of knowledge models from textbooks [5]. Figure 2.A presents a summarized version of this process. During the first stage, a knowledge model is extracted from a textbook using its formatting, structure, and organization. Then, the model is enriched with additional semantic information from DBpedia by linking identified index terms into DBpedia resources. The second stage is more relevance for our current work, therefore we provide a more detailed description of it.

First, the index section of the textbook is processed, all terms are extracted and added to a glossary. Then, during the term linking step, each term is queried against DBpedia. When a result is retrieved, it can be either (1) a resource or (2) a list of candidate resources with the same or similar names. Suppose the result is a

[6] e.g. while analyzing textbooks on statistics, we have observed a rather stable ratio of about 2/3 of all index entries categorizable as relevant to the domain of statistics

[7] For example, the resource about the *arithmetic mean* has the full URI *http://dbpedia.org/resource/Arithmetic_mean* corresponding to the shorter (namespaced) version *dbr:Arithmetic_mean*. For simplicity, in the rest of the paper we use namespaced URIs of resources and properties. Namespace prefixes can be found at https://dbpedia.org/sparql?help=nsdecl

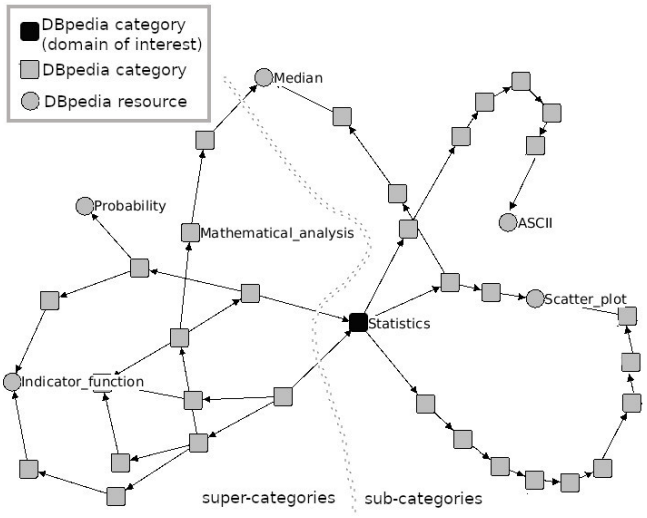

Figure 1: Resources connected using the categorization system.

single resource r, and it belongs to a sub-category that is at most 3-hops away from the target category (representing D). r is considered to belong to D unambiguously and is linked to the target term. After all index terms have been tossed to the DBpedia SPARQL interface as queries, we obtain a set of resources called *SEED*[8]. Each $r \in SEED$ represents a *seed resource* in D. When multiple resources are retrieved, we compute a similarity function between each candidate resource' abstract and the cumulative abstract of *SEED*. The candidate resource with the highest similarity, which is also above a threshold, is linked to the target term. This second set of resources is called *RES*. In the final step, each linked term is enriched with semantic information extracted from DBpedia.

Once the model is fully linked and enriched, it is serialized as an XML file using the Text Encoding Initiative (TEI)[9]. Up to this point, we have not experimented with further improvement of our models and considered all index terms that we extract equally important for the subject of the textbook, i.e. for the domain model extracted from it. In the next section, we perform a deeper analysis of the textbook indices and refine extracted models by labeling their concepts according to their domain specificity. The star in fig. 2.A indicates where the approach presented in this paper fits within the general workflow.

3 APPROACH

To decide whether a resource (and an index term linked to it) belongs to a domain, our approach heavily relies on the structure of the DBpedia category graph that provides an easily navigatable web of (sub)domains for any root domain category. However, some properties of this graph present challenges. For example, it is possible for a resource to be connected to a root category using both sub- and super-categories. The former connection denotes belonging to the main domain, and the latter indicates that the resource is related indirectly through a different domain. When both connections exist, it is not possible to discern which one is stronger. In fig. 1, *Median* is connected to *Statistics* both directly to the root and indirectly through *Mathematical_analysis*. Another important consideration is that a presence of a direct hierarchical path from a domain category to a resources is not enough to assume that the resource belongs to the domain. In fig. 1, *ASCII* is connected to *Statistics* by a chain of 6 sub-categories, yet it is not a statistical concept. The third challenge is to decide if resources connected only through super-categories are sufficiently relevant to be considered as a part of a related domain or not (e.g., in fig. 1 do resources *Probability* and *Indicator_function* belong to domains related to *Statistics*?). Finally, a resource could have multiple paths to a domain using different categories with varying degrees of relevance, making it necessary to identify the most related one. For example, *Scatter_plot* has two direct paths to *Statistics* using only sub-categories in fig. 1.

To combat these challenges, we need to combine information from textbook models with the content and structural properties of the DBpedia knowledge graph. Figure 2.B shows the overall process for identifying domain specificity of individual DBpedia resources / index terms. The final result is the refined model, which is essentially a domain specificity graph where each individual vertex represents a concept and each concept is marked with an individual specificity label. It is essential to mention that each domain specificity graph indicates only the specificity of concepts extracted from a source textbook (or a set of textbooks) regarding the target domain; it does not try to label the specificity of all possible resources in DBpedia regarding this domain. The following subsections explain the stages of the proposed approach. Additionally, the algorithmic representations of the main methods of the approach can be found in the Appendix.

3.1 Initialization

The first stage of the approach is preparatory. The DBpedia resources matched to the textbook's index terms are divided into two sets: *SEED* and *RES* (see Section 2.3). An empty domain specificity graph is created. The graph is denoted by $DSG = (V, E)$, where V is a set of vertices representing concepts and categories, and $E \subseteq V \times V$ is a set of unweighted and directed edges representing the hierarchical relations of resources in the categorization system of DBpedia. There are one special vertex, $root \subseteq V$, and two subsets of vertices, $C \subseteq V$ and $CAT \subseteq V$. *Root* represents the main DBpedia category of D (e.g., *dbc:Statistics*). C is composed of **concepts** - DBpedia resources linked to index terms whose domain specificity has been identified. Finally, CAT vertices are DBpedia categories with an identified domain specificity and linked to the concepts in C. Elements in C and CAT are annotated with one of the domain specificity labels from L. A path denoted by $p_i = (root \supseteq^c cat_1 \supseteq^c \ldots \supseteq^c cat_n \supseteq^c c) \mid cat_x \in CAT, c \in C,$ and $i \in \mathbb{N}$, connects c to $root$. $P_c = \{p_1, \ldots, p_g\}$ is a set of g paths connecting c to $root$. For a p_i, if $\forall cat_x \in p_i, root \supseteq^c cat_x$, c is a *core-domain* or *in-domain* concept. On the contrary, if $\exists cat_x \in p_i, root \subseteq^c cat_x$, c is a *related-domain* concept. Finally, if $\nexists p_i \mid c \in p_i$, c is an *out-of-domain* concept and it is disconnected from *DSG*. Figure 3 shows a fragment of the domain specificity graph used in Section 3.7.

Resources in *SEED* and *RES* are called concepts in the next stages since their domain specificity is being determined.

[8]In our previous work, this set was called the *core set*
[9]https://tei-c.org/

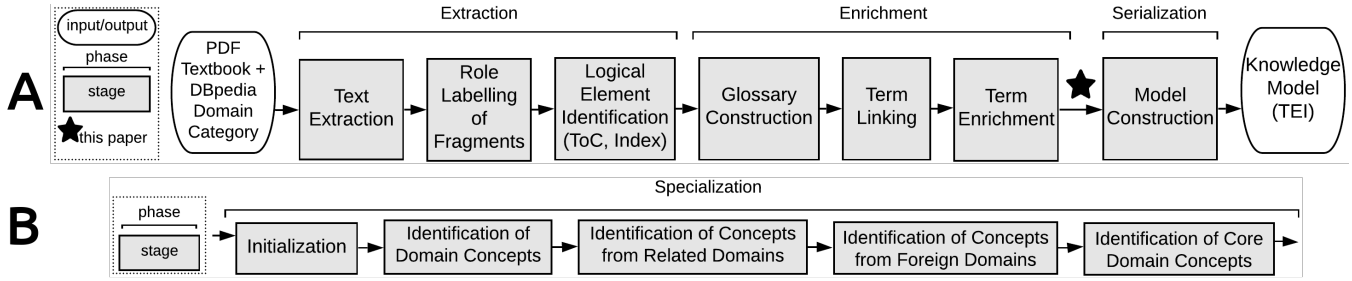

Figure 2: A. Knowledge model creation. B. Identification of domain specificity.

3.2 Identification of Domain Concepts

The second stage of the approach identifies concepts that belong to the main domain of the textbook. First, concepts from *SEED* are added to the *DSG*. All $c \in SEED$ are considered part of D because they have at least one category three or fewer hops from the *root*. This number of hops was selected after running several experiments. It is also in line with previous work [17, 42]. For each $c \in SEED$, the **path finder** method discovers P_c. This method is a depth-first search (DFS) [10, chap. 22.3] algorithm that starts with all direct categories of c to find all sequences of form $c \subseteq^c cat_n \subseteq^c \ldots \subseteq^c cat_1 \subseteq^c root$. A path p_i is constructed and added to P_c when the *root* is found within the maximum number of hoops (3) in a sequence. A graph traversal algorithm is necessary because although SPARQL 1.1 supports property paths [51], it does not return arbitrary paths of variable length. Instead, in practice, it only allows testing reachability [13, 50].

Then, the **path scoring** method assess each $p_i \in P_c$ to assign a score (s_{cat_n}) according to their belonging to D. Each category in a p_i is scored as the average of three sub-scores: s_1 - the similarity between all the category's resources and *SEED*; s_2 - the percentage of category's resources that have a direct link to *root*; and s_3 - the similarity between the category and its super-category along the path. The final score of a p_i is the score of cat_n, which is the direct parent category of c. It is important to note that the same category will have different scores in different p_i since each category score incorporates all its super-categories up to the *root*. After the scoring is finished, the p_i with the highest score is selected. Finally, c is added to C with $l = in\text{-}domain$. The categories in the selected p_i are added to CAT with $l = in\text{-}domain$. All pairs of vertices in p_i are added to E. By selecting the p_i with the highest score, only the most relevant categories to D are added initially to *DSC*.

In the second part of the current stage, more *in-domain* resources are discovered. First, since at this point CAT contains only *in-domain* categories, we can directly add the concepts that belong to any of those categories. Each $c \in RES$ is added to C if: $c \subseteq^c cat_n \in CAT$. Then, for each of the the remaining $c \in RES$, all corresponding p_i are discovered with the **path finder** method. For optimization purposes, in this case we use a maximum of six hops; this helps us avoid finding too many irrelevant paths. Next, paths are scored using the **path scoring** method. However, we add a inclusion/exclusion mechanism based on thresholds (represented using lower case Greek letters) to detect when a category/path deviates too much from D. A p_i is excluded if $s_2 < \alpha \wedge s_3 < \beta$ for cat_n. After discarding, if c has at least one p_i, c is added to C with $l = in\text{-}domain$. The categories in the highest scored p_i are added to CAT with $l = in\text{-}domain$. All pairs of vertices in p_i are added to E.

3.3 Identification of Concepts from Related Domains

The third stage of the approach is to identify the concepts that are not a part of D, but still relevant enough to be considered from related domains. First, for each $c \mid c \in RES \wedge c \notin C$, all the p_i that connect c to *root* indirectly are discovered with the **related path finder** method. This method is similar to **path finder**, but it goes up (using super-categories) and down (using sub-categories) along the hierarchy of categories, which allows finding paths of the form: $p_i = (root \subseteq^c \ldots \subseteq^c cat_x^{sp} \supseteq^c \ldots \supseteq^c cat_n \supseteq^c c)$, which indirectly connect c to the *root* using a super-parent category cat_x^{sp}. In practice, this method finds the lowest common ancestors of c and *root* [8]. For optimization purposes, and due to the fact that using super-categories might result in two very unrelated domains to be connected (e.g., dbc:Statistics and dbc:Musicology are connected through the dbc:Academic_disciplines super-parent category), we use a maximum of eight hops.

Next, indirect p_i are scored using the **related path scoring** method with an exclusion threshold. This method is similar to **path scoring**, with only one change in s_2: it checks the intersection between the links from the category's resources and the links from the *SEED* resources. It has been shown that resources sharing similar links are related [53]. Using our exclusion mechanism, a p_i is too unrelated to D if $s_2 < \gamma \vee s_{cat_n} < \delta$ for cat_n. After discarding, if a concept has at least one indirect p_i remaining, c is added to C with $l = related\text{-}domain$. The categories and vertices in the best p_i are added to CAT with $l = related\text{-}domain$ and to E, respectively.

In rare cases, some concepts can be considered *in-domain* even though there is no direct path to *root*. For example, the *dbr:Sample_space* concept belongs to the probability domain, but it is a significant concept of statistics. For such cases, the **related path assessment** method checks five constraints to identify a resource as *in-domain* despite an indirect p_i. First, if the related p_i is connected through a sibling category of the *root*. Second, if the score of cat_n is high enough ($> \epsilon$). Third, if the percentage of shared links is high enough ($> \zeta$). Fourth, if the similarity between the resource and *SEED* is high enough ($> \eta$). Fifth, if the resource and the *root* link to each other. If at least four out of five constraints are met, the l associated to c is changed to *in-domain*.

3.4 Identification of Concepts from Foreign Domains

The fourth stage of the approach identifies the resources that are from domains that are not related to the subject of the textbook. This process is straightforward. Any remaining $c \mid c \in RES \wedge c \notin C$ is unrelated to D and is added to C with $l = out\text{-}of\text{-}domain$.

3.5 Identification of Core Domain Concepts

The final stage evaluates all $c \in C$ with $l = in\text{-}domain$ to see if any of them represent one of the most important concepts in D, which are the *core-domain* concepts. We assume that the most used resources through textbooks and DBpedia should indicate the most relevant concepts in D. We apply the **core concepts assessment** method to discover such concepts. First, a popularity score is assigned to each concept based on how many times it has been seen in the textbooks and the number of other concepts linking to it in the *DSG*. The resources with the popularity scores in the upper quartile are selected. Finally, only the concepts that are referenced widely from outside D are selected. We consider that core concepts are so relevant that they are also used in other domains. If most of the resources that reference (link) the concept belong to a different domain, then it is marked as a *core-domain* concept. To check if a resource r belongs to D or a different domain, a simplified version of the *related path scoring* method is used. Each of the direct categories of a resource is checked: if the combined score of the similarity between resources in the category and the percentage of shared links between the *seed resources* is below θ, the category is considered to be from a different domain. Finally, if most of the categories from r are from a different domain, then $r \notin D$.

3.6 Thresholds

After several experiments, the used thresholds were calibrated with the values shown in Table 1. The selected values were flexible but also robust enough to achieve good results in two very different domains: *statistics* and *ancient philosophy* (see Section 4). Section 4.4 includes a brief discussion on threshold calibration.

Table 1: Threshold values.

Threshold	α	β	γ	δ	ϵ	ζ	η	θ
Value	0.1	0.4	0.3	0.25	0.3	0.5	0.2	0.1

3.7 Example

Figure 3 presents the domain specificity graph of one *statistics* textbook [15]. In miniature, the whole graph is presented. The subgraph on the center is a zoom in and it contains the same five resources as in fig. 1, but the two graphs (named DOMAIN and DBPEDIA respectively) look completely different. In our DOMAIN graph, each resource is a concept with a clear relation to the domain of interest; there are no multiple paths from different domains to the resources, as in the DBPEDIA graph. The challenges described at the beginning of this section have been addressed. The *Median* concept had two possible paths and relations in the DBPEDIA graph, but in the DOMAIN graph it has been identified as a concept belonging

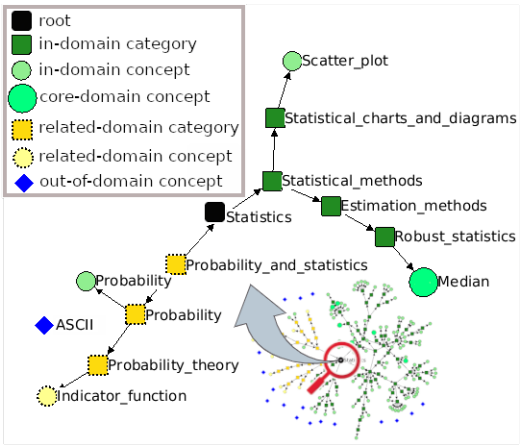

Figure 3: Example of a domain specificity graph.

to the domain, even as *core-domain*. There is a connecting path from *statistics* to *ASCII* in DBPEDIA, but in DOMAIN, it has been identified as a *out-of-domain* concept. The *Probability* and *Indicator_functions* have been considered relevant enough to be classified as *related-domain* concepts in the DOMAIN graph. Finally, the most relevant path from *statistics* to *Scatter_plot* has been identified in DOMAIN, in contrast, there were two possible paths in the DBPEDIA graph.

Our domain specificity graphs identify not only the type of domain specificity relationships, but also allows for explaining how the relations exist. For example, in the DOMAIN graph, *Indicator_function* is a related concept to *statistics* from the *probability* domain.

4 EVALUATION

Three evaluation experiments have been conducted using a local copy of the latest dump of DBpedia (version 2020.12.01). The ground truth models generated for the first two evaluations are made publicly available[10].

4.1 Evaluation One: MAIN DOMAIN

The goal of the first experiment was to examine how well our approach can identify concepts that belong to a domain. We used textbooks about *statistics*. *Statistics* is richly connected to many other domains (different sub-fields of mathematics and computer science). For this reason, creation of a comprehensive list of *related-domain* resources is not practically feasible. Therefore, for this evaluation, we were interested only in the classification accuracy of the *core-domain* and *in-domain* terms.

4.1.1 Data sets & Procedure. Ten introductory statistics textbooks were used [12, 15, 16, 18, 21, 24, 29, 38, 55, 59]. From each textbook, we extracted a knowledge model and enriched its index terms with their corresponding resources in DBpedia using the *dbc:Statistics* category as the main domain. Then, we applied the described approach for the domain specificity graph construction. The approach was executed over the combination of all selected textbooks to obtain their cumulative knowledge graph. To determine

[10] https://github.com/intextbooks/domain-specificity

the true domain specificity label of all concepts we created a ground truth using multiple sources. First, we identified the *core-domain* concepts by checking the intersection of six statistical glossaries ([20, 23, 28, 34, 52, 56]). A term was marked as *core-domain* if it appeared in at least two glossaries or in [34], which is a master list of core statistical terms created manually and rated by statistics instructors. After finding the corresponding DBpedia resources we obtained 186 core concepts. Then, we identified the *in-domain* concepts by merging three sources: 1) all DBpedia resources extracted from a list of statistical articles in Wikipedia[11], 2) a list of statistical DBpedia resources constructed using the ISI Multilingual Glossary of Statistical Terms (as described in [3]), and 3) all resources in DBpedia which have explicitly encoded in their URI that they belong to statistics (e.g., *dbr:Q-value_(statistics)*). After removing the concepts already marked as core, we got the final list of 2658 *in-domain* concepts. Any concept outside the ground truth was classified as *other-domain* (meaning it can be either *related-domain* or *out-of-domain*). We calculated the confusion matrix and used standard accuracy, precision, and recall as our evaluation metrics [44, chap. 9]. Additionally, we used a simple path-based baseline (BL) for comparison as used in previous works [33, 42]. BL only used the DBpedia categorization system: if a resource could be reached from the domain root using sub-categories within 2-hops, it was classified as *core-domain*; using sub-categories within 4-hops it was classified as *in-domain*; otherwise, it was considered being too far from the domain root and was classified as *other-domain*. Finally, we used the McNemar's test [49] (a non-parametric test for paired nominal data) to analyze statistical significance of differences in accuracy between our graph and the baseline.

4.1.2 Results. Table 2 shows the results of focusing on the boundary of the target domain (is a concept a part of the domain or not). The approach outperforms the baseline in terms of precision, recall and accuracy. The later difference is statistically significant as indicated by the McNemar's test ($\chi^2(1, N = 648) = 13.829$, $p < 0.001$).

When the task becomes more difficult and we try finding which concepts belong to the core of the domain, the gap between the baseline and the proposed approach increases overall, as seen in Table 3. The the baseline outperforms our approach only in terms of recall of *core-domain* concepts. However, on the other two recall values as well as the precision of obtained labels, our approach clearly performs better. It is important to mention that for many tasks, precision is a much more important metric as its increase leads to elimination of type I error. It is also worth noticing, that our approach makes considerably fewer serious mistakes (labeling *core-domain* concepts as *other-domain* and vice versa). When it comes to accuracy, again, the difference between the two methods was significant according to a McNemar's test ($\chi^2(1, N = 648) = 35.359$, $p < 0.001$).

4.2 Evaluation Two: MULTIPLE DOMAINS

We have analyzed the cross-domain robustness of the approach by applying it to the *ancient philosophy* domain. Additionally, we have checked the ability of the approach to distinguish between *in-domain*, *related-domain*, and *out-of-domain* concepts.

[11]https://en.wikipedia.org/wiki/List_of_statistics_articles

4.2.1 Data sets & Procedure. We used one textbook about ancient Greek and Roman philosophy [31]. Its knowledge model was extracted and enriched with corresponding DBpedia resources using the *dbc:Ancient_Greek_philosophy* category, after which . its domain specificity graph was generated. To determine the true domain specificity label of the concepts we created a ground truth manually. For each concept we assigned one of the three possible labels (*in-domain*, *related-domain*, and *out-of-domain*). We considered as *in-domain* the concepts directly associated with the 'topics' inferred from the chapters and covered across the textbook (e.g., Plato). *Related-domain* concepts corresponded to general philosophical notions and philosophers, people and places from the same era, and auxiliary philosophical terms (e.g., logic). General and broad concepts (e.g., art) were classified as *out-of-domain*. In case of doubt, we used the textbook itself, the Stanford Encyclopedia of Philosophy[12], and general web searches to clarify the relevance of a concept in the domain. A total of 426 unique concepts were classified. We used the same metrics as in the previous experiments. BL was applied as well, but in this case, any resource reached directly within 4-hops was classified as *in-domain*, within 4-hops using an indirect path as *related-domain*, or otherwise as *out-of-domain*. The McNemar's test was applied to verify statistical significance.

4.2.2 Results. Table 4 presents the results. They show that the accuracy of the approach remains high and stable in a different domain as well. Also, when identifying the possible domains of concepts, the accuracy of our approach is 20% higher than the baseline. Our method's combination of content and structural properties gets high precision and recall values across all possible labels compared to the method using only category-based paths (BL) . Some resources have both direct and indirect paths to the domain, and the use of a scoring function is the key to decide the proper relation to the main domain. Finally, there is a statistically significant difference between the accuracy of our model and the baseline according to a McNemar's test ($\chi^2(1, N = 426) = 62.959$, $p < 0.001$).

4.3 Evaluation Three: APPLICATION

The goal of this experiment was to show the added value of domain specificity labels. To that end, we applied our approach to model documents for a simple query-based retrieval task.

4.3.1 Data sets & Procedure. We employed Apache Lucene[13] to construct a web search system for textbooks. Our experimental model (*ind+*) used a combination of textual content, index terms, and *core-domain* and *in-domain* labels to model (sub)chapters and sections of textbooks, where index terms and labeled index terms received more weight. For each search query, a ranked list of documents was retrieved using a standard tf-idf scoring formula [48]. As baselines, we used two variations of the system: *tf-idf* uses only the content, and *ind* uses the content and the index terms (without domain specificity labels). To evaluate the added value of domain specificity information against the baselines, we followed a standard procedure [39, Chapter 8]. We selected two textbooks from Section 4.1 ([38] and [15]) as the target document collection. We composed a set of queries using 20 syllabi of university-level

[12]https://plato.stanford.edu/index.html
[13]https://lucene.apache.org/

Table 2: Results for domain boundary detection (statistics).

		actual relation					
	n = 648	in-domain	other-domain	$\sum$	Precision	Recall	Accuracy
GRAPH	in-domain	383	11	*394*	.972	.897	.915*
	other-domain	44	210	*254*	.827	.950	
	$\sum$	*427*	*221*				
BL	in-domain	362	13	*375*	.965	.848	.880
	other-domain	65	208	*273*	.762	.941	
	$\sum$	*427*	*221*				

*Statistical significance against BL.

Table 3: Results for core domain boundary detection (statistics).

		actual relation						
	n = 648	core-domain	in-domain	other-domain	$\sum$	Precision	Recall	Accuracy
GRAPH	core-domain	45	5	0	*50*	.900	.306	
	in-domain	94	239	11	*344*	.695	.854	.762*
	other-domain	8	36	210	*254*	.827	.950	
	$\sum$	*147*	*280*	*221*				
BL	core-domain	59	87	6	*152*	.388	.401	
	in-domain	70	146	7	*223*	.655	.521	.637
	other-domain	18	47	208	*273*	.762	.941	
	$\sum$	*147*	*280*	*221*				

*Statistical significance against BL.

Table 4: Results for multi-domain boundary detection (ancient philosophy).

		actual relation						
	n = 426	in-domain	related-domain	out-of-domain	$\sum$	Precision	Recall	Accuracy
GRAPH	in-domain	129	8	0	*137*	.942	.977	
	related-domain	3	148	16	*167*	.886	.937	.932*
	out-of-domain	0	2	120	*122*	.984	.882	
	$\sum$	*132*	*158*	*136*				
BL	in-domain	127	8	2	*137*	.927	.962	
	related-domain	1	51	4	*56*	.911	.323	.723
	out-of-domain	4	99	130	*233*	.558	.956	
	$\sum$	*132*	*158*	*136*				

*Statistical significance against BL.

statistics courses[14] to represent typical information needs of students: 100 queries were selected from statistics syllabi and 40 from statistics-related syllabi. Additionally, we selected ten queries from the Table of Content from an additional textbook [45]. Three experts in statistics were recruited to generate a set of relevance assessments for query-document pairs. For each query, each rater indicated the chapters, sub-chapters, and sections from the two textbooks that were relevant using a three-point scale: (1) partially relevant, (2) relevant, and (3) highly relevant. Finally, we applied the experts' assessments as the ground truth to evaluate the results retrieved by the queries using average *normalized discounted cumulative gain* (NDCG) at 1, 3, and 5. NDCG@1 measures the effectiveness of retrieving the most relevant document, while @3 and @5 measure it for three and five most relevant documents, respectively. Inter-reliability between raters was calculated using pooled Fleiss' kappa across all queries [14, 43]. The resulting kappa of 0.36 is considered a fair agreement. Given that all raters were experts and made their relevance assessments fully independently from each other, we used a smoothed factor[15] to compute the final relevance for each query-document pair.

4.3.2 Results. NDCG mean values and standard deviations are presented in Table 5. Results show that the *ind+* model using domain specificity of index terms supports more accurate retrieval of relevant documents. Pairwise t-tests confirm that the difference between *ind+* and *tf-idf* is significant across all three metrics, and between *ind+* and *ind* for NDCG @3 and @5 (see Table 5).

The three systems perform similarly when a query corresponds to a single concept that appears textually in concise (sub)chapters (e.g., "nonparametric statistics"). The use of index terms in both *ind* and *ind+* helps to find synonyms. For example, the "graph" query is matched to concepts like "chart," "bubble plot," and "scatter diagram." Finally, the domain specificity data is useful as it encodes information about the relevance of a term in a domain. The *ind+* model is better than *ind* when the queries correspond to concepts

[14] e.g., https://www.stat.berkeley.edu/~mgoldman/sylsm09.pdf

[15] Calculated using $1 + ((1 - support) \cdot (\log(support) \cdot support))$, which was inspired by the *expected information gain* formula used in decision trees

with identified domain specificity (e.g., "type I and type II errors" which is a *core-domain* concept).

ind+ is a simple model that could be further imrpoved using the full potential of domain specificity and the *DSG*. For example, queries like "histograms and other graphs" could be matched to their corresponding main concept in the *DSG* (*db:Histogram*) to use the information from its parent category (*dbc:Statistical_charts_and_diagrams*) to identify related concepts (*dbc:Pie_chart*, *dbc:Box_plot*, ...). Creating a more powerful model is part of our future work.

Table 5: Retrieval of documents.

	NDCG@1			NDCG@3			NDCG@5		
	tf-idf	*ind*	***ind+***	*tf-idf*	*ind*	***ind+***	*tf-idf*	*ind*	***ind+***
M	.303	.410	**.420**	.327	.427	**.452**	.348	.459	**.474**
SD	.386	.411	.412	.349	.355	.353	.355	.358	.356
t	-3.24	.639	-	-4.74	2.5	-	-4.95	2.04	-
df	140	140	-	140	140	-	140	140	-
Sig	< .01	.524	-	< .0001	< .05	-	< .0001	< .05	-

Table 6: Coverage of *in-domain+* concepts (statistics).

Set size	1	2	3	4	5	6	7	8	9	10
M	97	157	202	239	270	297	321	343	362	383

4.4 General Discussion

Our approach is sensitive to the incompleteness / inconsistencies in DBpedia. Since classification of resources into categories is done mainly by Wikipedia authors, it is a subjective process and some categories can be missing or inappropriate. For example, the categories of *dbr:Mutual_exclusivity* do not include *probability*, therefore it is not classified as a *related-domain* concept of *statistics*). Some other resources have categories that reflect too high relevance to the domain. In our *ancient philosophy* evaluation, *dbr:Alcibiades* is marked as a *pupil of Socrates* in DBpedia, and therefore it is classified as an *in-domain* concept, while, in fact, he was a not a philosopher.

We also noticed that for the *ancient philosophy* domain, the abstracts of the resources tend to be more general than for *statistics*, and higher thresholds in the scoring functions would have been beneficial. One possible solution for this situation is to automatically adjust the thresholds based on the seed resources' path scores.

Finally, as we used more textbooks in the same domain, the coverage of concepts in the domain increases. We calculated the number of *in-domain+* concepts discovered when increasing the number of textbooks. For the *statistics* domain, we took ten textbooks and experimented with all possible sets (permutations) when using from 1 to 10 textbooks. Table 6 contains the average numbers of correctly discovered *in-domain+* concepts for each set.

5 RELATED WORK

Textbook Indices. Index sections are a source of document and domain-specific terms. Surprisingly, textbook indices have not yet received much attention in the literature. NLP tools and heuristic reasoning were applied to extract terminology using the index section of a single book [35]. A security textbook was used as a source of terms to develop a cybersecurity ontology [58]. Besides using index terms, terminology extraction methods can automatically identify and extract core vocabulary of a specialized domain in un- and semi-structured corpora [19, 37].

Domain Specificity. Automatic approaches have been proposed for domain specificity classification: using sample terms to query documents and estimate the domain specificity according to the distribution of the domain of those documents [32], training of classifiers to assign to a term one category from a predefined set of classes [47], and using terminological services to create domain vectors and assign categories to vector representations of documents [25].

Domain Discovery. In-domain terms can be discovered by generating a list with relevant terms in a specific domain. Domain-relevant terms can be extracted from documents using TFIDF techniques [62]. Wikipedia has been widely used to extract domain-specific terms/thesauri [41], and sets of relevant categories to a domain of interest [42, 57]. Finally, DBpedia has been used to extract domain-specific hierarchical subgraphs of categories and resources [33].

Semantic Relatedness. For domain specificity, the notion of semantic relatedness is important to identify if a resource (or set of resources) is part of the target domain. Several features have been used to compute the relatedness between elements: the Wikipedia's network of inter-article links [60], the proximity of terms in DBpedia [36], and the similarity of the properties of two resources [46]. At the domain level, a combination of features (graph-based, text-based, and web-based) has been used to rank DBpedia resources based on their relatedness to one specific domain [17].

6 CONCLUSION AND FUTURE WORK

We explored how textbook indices can be used to extract high-quality knowledge graphs in narrow domains. Specifically, we presented an approach to automatically label terms in relation to the main subject of a textbook (i.e. domain specificity). We extended the traditional binary classification of specificity into four labels that better reflect the degree of how much a term belongs to the target subject: *core-domain* for the most important domain concepts, *in-domain* for regular concepts in the main domain, *related-domain* for neighboring domains, and *out-of-domain* for concepts unrelated to the domain, but important for pedagogical reasons. Ultimately, this approach allows to address one of the biggest problems of textbook indices as sources of domain knowledge, namely presence of a large portion of entries that are either weakly related or unrelated to the target domain. Our evaluations experiments have demonstrated that the approach is capable to distinguish with high accuracy between the concepts relevant and non-relevant to the domain. Additionally, the accuracy of identifying the most important *core-domain* concepts also remains considerably high. Moreover, the approach has been successfully tested across two different domains (*statistics* and *ancient philosophy*). Finally, we showed that the domain specificity information can be helpful in the context of information retrieval tasks.

Our next step is to further experiment with the potential of domain specificity information when using our knowledge graphs in combination with powerful language models. One possible direction is to explore informed word embeddings [54, 64], where we could use the index terms and different weights according to their domain specificity to produce embeddings reflecting a domain of interest.

REFERENCES

[1] 2015. *Best Practices for Indexing* (1 ed.). American Society for Indexing (ASI).
[2] 2017. *The Chicago Manual of Style* (17 ed.). The University of Chicago Press.
[3] Isaac Alpizar-Chacon and Sergey Sosnovsky. 2019. Expanding the Web of Knowledge: one Textbook at a Time. In *Proceedings of the 30th on Hypertext and Social Media* (Hof, Germany) *(HT '19)*. ACM, New York, NY, USA.
[4] Isaac Alpizar-Chacon and Sergey Sosnovsky. 2020. Order out of Chaos: Construction of Knowledge Models from PDF Textbooks. In *Proceedings of the 20th ACM Symposium on Document Engineering* (San Jose, CA, USA) *(DocEng 2020)*. ACM, New York, NY, USA.
[5] Isaac Alpizar-Chacon and Sergey Sosnovsky. 2021. Knowledge models from PDF textbooks. *New Review of Hypermedia and Multimedia* 0, 0 (2021), 1–49. https://doi.org/10.1080/13614568.2021.1889692
[6] Kurt Ament. 2001. *Indexing: a nuts-and-bolts guide for technical writers*. William Andrew.
[7] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*. Springer, 722–735.
[8] Michael A Bender, Martin Farach-Colton, Giridhar Pemmasani, Steven Skiena, and Pavel Sumazin. 2005. Lowest common ancestors in trees and directed acyclic graphs. *Journal of Algorithms* 57, 2 (2005), 75–94.
[9] Pat F. Booth. 2013. *Indexing The Manual of Good Practice*. De Gruyter.
[10] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. *Introduction to Algorithms, Third Edition* (3rd ed.). The MIT Press.
[11] András Csomai and Rada Mihalcea. 2007. Investigations in Unsupervised Back-of-the-Book Indexing.. In *FLAIRS Conference*. 211–216.
[12] Peter Dalgaard. 2011. *Introductory statistics with R*. Lightning Source UK Ltd.
[13] Laurens De Vocht, Sam Coppens, Ruben Verborgh, Miel Vander Sande, Erik Mannens, and Rik Van de Walle. 2013. Discovering meaningful connections between resources in the web of data. In *LDOW*.
[14] Han De Vries, Marc N Elliott, David E Kanouse, and Stephanie S Teleki. 2008. Using pooled kappa to summarize interrater agreement across many items. *Field methods* 20, 3 (2008), 272–282.
[15] Frederik M. Dekking, Cor Kraaikamp, Hendrik P. Lopuhaä, and Ludolf E. Meester. 2005. *A modern introduction to probability and statistics: understanding why and how*. Springer.
[16] Jay L. Devore and Kenneth N. Berk. 2012. *Modern Mathematical Statistics with Applications*. Springer.
[17] Tommaso Di Noia, Vito Claudio Ostuni, Jessica Rosati, Paolo Tomeo, Eugenio Di Sciascio, Roberto Mirizzi, and Claudio Bartolini. 2016. Building a relatedness graph from linked open data: A case study in the it domain. *Expert Systems with Applications* 44 (2016), 354–366.
[18] David M. Diez, Mine Cetinkaya-Rundel, and Christopher D. Barr. 2019. *OpenIntro statistics* (fourth ed.). openintro.org.
[19] Anurag Dwarakanath, Roshni R Ramnani, and Shubhashis Sengupta. 2013. Automatic extraction of glossary terms from natural language requirements. In *2013 21st IEEE International Requirements Engineering Conference (RE)*. IEEE, 314–319.
[20] Valerie J. Easton and John H. McColl. 1997. Statistics Glossary from STEPS. http://www.stats.gla.ac.uk/steps/glossary/index.html. [Online; accessed 12-2020].
[21] Michael Havbro. Faber. 2012. *Statisticcs and probability theory: in pursuit of engineering decision support*. Springer Verlag.
[22] Michael Färber, Basil Ell, Carsten Menne, and Achim Rettinger. 2015. A comparative survey of dbpedia, freebase, opencyc, wikidata, and yago. *Semantic Web Journal* 1, 1 (2015), 1–5.
[23] Mark G Filler and James A DiGabriele. 2012. *A Quantitative Approach to Commercial Damages: Applying Statistics to the Measurement of Lost Profits*. John Wiley & Sons.
[24] Michael O. Finkelstein. 2009. *Basic concepts of probability and statistics in the law*. Springer.
[25] Robert Gaizauskas, Emma Barker, Monica Lestari Paramita, and Ahmet Aker. 2014. *Assigning Terms to Domains by Document Classification*. Technical Report. 11–21 pages. https://demo.taas-project.eu/domains.
[26] Eelco Herder, Sergey Sosnovsky, and Vania Dimitrova. 2017. Adaptive intelligent learning environments. In *Technology Enhanced Learning*. Springer, 109–114.
[27] Birger Hjørland. 2008. What is knowledge organization (KO)? *KO Knowledge Organization* 35, 2-3 (2008), 86–101.
[28] International Statistical Institute. 2011. ISI Multilingual Glossary of Statistical Terms. http://isi.cbs.nl/glossary/. [Online; accessed 12-2020].
[29] Hans-Michael Kaltenbach. 2012. *A concise guide to statistics*. Springer.
[30] William E. Kasdorf. 2003. *The Columbia guide to digital publishing*. Columbia University Press.
[31] Anthony Kenny. 2004. *Ancient Philosophy: A New History of Western Philosophy, Volume 1*. OUP Oxford.
[32] Mitsuhiro Kida, Masatsugu Tonoike, Takehito Utsuro, and Satoshi Sato. 2007. Domain classification of technical terms using the web. *Systems and Computers in Japan* 38, 14 (2007), 11–19.
[33] Sarasi Lalithsena, Sujan Perera, Pavan Kapanipathi, and Amit Sheth. 2017. Domain-specific hierarchical subgraph extraction: A recommendation use case. In *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 666–675.
[34] R. Eric Landrum. 2005. Core terms in undergraduate statistics. *Teaching of Psychology* (2005).
[35] Mikel Larrañaga, Urko Rueda, Jon A. Elorriaga, et al. 2004. Acquisition of the domain structure from document indexes using heuristic reasoning. In *International Conference on Intelligent Tutoring Systems*. Springer, 175–186.
[36] José Paulo Leal, Vânia Rodrigues, and Ricardo Queirós. 2012. Computing semantic relatedness using dbpedia. In *Symposium on Languages, Applications and Technologies, 1st*. Schloss Dagstuhl, 133–147.
[37] Lucelene Lopes, Renata Vieira, Maria José Finatto, and Daniel Martins. 2010. Extracting compound terms from domain corpora. *Journal of the Brazilian Computer Society* 16, 4 (2010), 247–259.
[38] Birger Madsen. 2011. *Statistics for Non-Statisticians*. Springer.
[39] Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. 2018. . Cambridge University Press.
[40] David Martín-Moncunill, Miguel-Ángel Sicilia-Urban, Elena García-Barriocanal, and Salvador Sánchez-Alonso. 2015. Evaluating the degree of domain specificity of terms in large terminologies. *Online Information Review* (2015).
[41] David Milne, Olena Medelyan, and Ian H Witten. 2006. Mining domain-specific thesauri from wikipedia: A case study. In *2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06)*. IEEE, 442–448.
[42] Daniil Mirylenka, Andrea Passerini, and Luciano Serafini. 2015. Bootstrapping domain ontologies from wikipedia: A uniform approach. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.
[43] Thomas R Nichols, Paola M Wisner, Gary Cripe, and Lakshmi Gulabchand. 2010. Putting the kappa statistic to use. *The Quality Assurance Journal* 13, 3-4 (2010), 57–61.
[44] David L. Olson and Dursun Delen. 2008. *Advanced data mining techniques*. Springer Science & Business Media.
[45] Brett W. Pelham. 2013. *Intermediate statistics: A conceptual course*. SAGE.
[46] Guangyuan Piao, Safina showkat Ara, and John G Breslin. 2015. Computing the semantic similarity of resources in dbpedia for recommendation purposes. In *Joint International Semantic Technology Conference*. Springer, 185–200.
[47] Leonardo Rigutini, Ernesto Di Iorio, Marco Ernandes, and Marco Maggini. 2006. Automatic term categorization by extracting knowledge from the Web. *Frontiers in Artificial Intelligence and Applications* 141 (2006), 531–535.
[48] Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management* 24, 5 (1988), 513–523.
[49] Steven L Salzberg. 1997. On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data mining and knowledge discovery* 1, 3 (1997), 317–328.
[50] Vadim Savenkov, Qaiser Mehmood, Jürgen Umbrich, and Axel Polleres. 2017. Counting to k or how SPARQL1. 1 property paths can be extended to top-k path queries. In *Proceedings of the 13th international conference on semantic systems*. 97–103.
[51] Andy Seaborne and Steven Harris. 2013. *SPARQL 1.1 Query Language*. W3C Recommendation. W3C. https://www.w3.org/TR/2013/REC-sparql11-query-20130321/.
[52] Philip B. Stark. 2019. Glossary of Statistical Terms from SticiGui. https://www.stat.berkeley.edu/~stark/SticiGui/Text/gloss.htm. [Online; accessed 12-2020].
[53] Khushboo Maulikmihir Thaker, Peter Brusilovsky, and Daqing He. 2018. Concept enhanced content representation for linking educational resources. In *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. IEEE, 413–420.
[54] Thy Thy Tran, Makoto Miwa, and Sophia Ananiadou. 2020. Syntactically-informed word representations from graph neural network. *Neurocomputing* 413 (2020), 431–443.
[55] Jan Ubøe. 2017. *Introductory statistics for business and economics: theory, exercises and solutions*. Springer International Publishing AG.
[56] University of St Andrews. 2016. Statistics Glossary. https://web.archive.org/web/20161113233144/http://www.st-andrews.ac.uk/psychology/current/statisticsglossary/#d.en.78939. [Online; accessed 12-2020].
[57] Jorge Vivaldi and Horacio Rodríguez. 2010. Finding domain terms using wikipedia. In *Proceedings of the 7th International Conference on Language Resources and Evaluation, LREC 2010*. 386–393. http://www.dlsi.ua.es/{~}atoral/
[58] Arwa Wali, Soon Ae Chun, and James Geller. 2013. A bootstrapping approach for developing a cyber-security ontology using textbook index terms. In *2013 International Conference on Availability, Reliability and Security*. IEEE, 569–576.
[59] Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers, and Keying Ye. 2012. *Probability & statistics for engineers & scientists*. Prentice Hall.
[60] Ian H Witten and David N Milne. 2008. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. (2008).
[61] Zhaohui Wu, Zhenhui Li, Prasenjit Mitra, and C Lee Giles. 2013. Can back-of-the-book indexes be automatically created?. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. 1745–1750.

[62] Feiyu Xu, Daniela Kurz, Jakub Piskorski, and Sven Schmeier. 2002. A Domain Adaptive Approach to Automatic Acquisition of Domain Relevant Terms and their Relations with Bootstrapping.. In *LREC*.
[63] Yu Yuan, Jie Gao, and Yue Zhang. 2017. Supervised learning for robust term extraction. In *2017 International Conference on Asian Language Processing (IALP)*. IEEE, 302–305.
[64] Junchi Zhang, Mengchi Liu, and Yue Zhang. 2020. Topic-informed neural approach for biomedical event extraction. *Artificial intelligence in medicine* 103 (2020), 101783.
[65] Yong Zhang, Fen Chen, Wufeng Zhang, Haoyang Zuo, and Fangyuan Yu. 2020. Keywords Extraction Based on Word2Vec and TextRank. In *Proceedings of the 2020 The 3rd International Conference on Big Data and Education*. 37–42.

A ALGORITHMS

Algorithm 1: Path scoring

Input: a P_c
for $p_i \in P_c$ **do**
  for $m \leftarrow 1$ *to* n **do** // n categories in p_i
    $R_m \leftarrow \texttt{getResourcesFromCategory}(cat_m \in p_i)$
    $s_1 \leftarrow \texttt{cosSim}(\texttt{getAbstr}(R_m), \texttt{getAbstr}(SEED))$
    // captures the sim between the category and D (represented by $SEED$)
    $s_2 \leftarrow \texttt{countDirectLink}(R_m, root)/|R_m|$
    // captures the % of resources that directly mention D
    $s_3 \leftarrow s_{cat_{m-1}} * \texttt{cosSim}(\texttt{getAbstr}(R_m), \texttt{getAbstr}(R_{m-1}))$
    // captures the sim between the current and the previous category in the path
    $s_{cat_m} \leftarrow \frac{s_1+s_2+s_3}{3}$
  end
end

Algorithm 2: Related path assessment

Input: a p_i
$cstr_1 \leftarrow$ **if** p_i *of the form:* $root \subseteq^c cat_1 \supseteq^c cat_2 \supseteq^c \ldots \supseteq^c c$ **then** 1 // c is connected through a sibling category of $root$
$cstr_2 \leftarrow$ **if** $s_{cat_n} > \epsilon$ **then** 1 // the score of the path is high
$cstr_3 \leftarrow$ **if** s_2 *of* $cat_n > \zeta$ **then** 1 // the % of shared links between c and D is high
$cstr_4 \leftarrow$ **if** $cosineSim(\texttt{getAbstr}(c), \texttt{getAbstr}(SEED)) > \eta$ **then** 1 // the similarity between c and D is high
$cstr_5 \leftarrow$ **if** $\texttt{countDirectLink}(c, root) \vee \texttt{countDirectLink}(root, c)$ **then** 1 // c and $root$ link to each ther
$cstr_t \leftarrow \sum_{n=1}^{5} cstr_n$

Algorithm 3: Core concepts assessment

Input: a p_i
$core \leftarrow \emptyset$
for $c \in C \wedge l_c = in\text{-}domain$ **do**
  $pop_1 \leftarrow \frac{\sum_{t=1}^{\#textbooks} 1 \text{ if } c \in textbook_t}{\#textbooks}$
  $pop_2 \leftarrow \frac{|\{e : e \in E \wedge e = \{x, c\}\}|}{\max\{|\{e : e \in E \wedge e = \{x, c'\}\}| : \forall c' \in C\}}$
  $pop_c \leftarrow \frac{pop_1+pop_2}{2}$
end
for $c \in C \wedge pop_c \in$ *upper quartile* **do**
  $general \leftarrow 0, specific \leftarrow 0$
  $LINKS \leftarrow$ *resources that link to* c
  for $res \in LINKS$ **do**
    for $cat \mid cat \supseteq^c res$ **do**
      $R_{cat} \leftarrow \texttt{getResourcesFromCategory}(cat)$
      $abstractsR_{cat} \leftarrow \texttt{getAbstr}(R_{cat})$
      $abstractsSEED \leftarrow \texttt{getAbstr}(SEED)$
      $sim \leftarrow \texttt{cosSim}(abstractsR_{cat}, abstractsSEED)$
      $links \leftarrow \frac{|\texttt{getLinks}(R_{cat}) \cap \texttt{getLinks}(SEED)|}{|\texttt{getLinks}(R_{cat})|}$
      $score \leftarrow \frac{sim+link}{2}$
      if $score > \theta$ **then**
        $specific \leftarrow specific + 1$
      else
        $general \leftarrow general + 1$
      end
    end
  end
  if $general > specific$ **then**
    $coreTerms \leftarrow coreTerms \cup \{c\}$
  end
end