Information and Software Technology 110 (2019) 3–16

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Detecting terminological ambiguity in user stories: Tool and experimentation

Fabiano Dalpiaz[a,*], Ivor van der Schalk[a], Sjaak Brinkkemper[a], Fatma Başak Aydemir[b], Garm Lucassen[c]

[a] *Utrecht University, the Netherlands*
[b] *Boğaziçi University, Turkey*
[c] *Secfi Inc., the Netherlands*

ARTICLE INFO

Keywords:
Natural language processing
Requirements engineering
Empirical software engineering
Ambiguity
User stories

ABSTRACT

Context. Defects such as ambiguity and incompleteness are pervasive in software requirements, often due to the limited time that practitioners devote to writing good requirements. *Objective.* We study whether a synergy between humans' analytic capabilities and natural language processing is an effective approach for quickly identifying near-synonyms, a possible source of terminological ambiguity. *Method.* We propose a tool-supported approach that blends information visualization with two natural language processing techniques: conceptual model extraction and semantic similarity. We evaluate the precision and recall of our approach compared to a pen-and-paper manual inspection session through a controlled quasi-experiment that involves 57 participants organized into 28 groups, each group working on one real-world requirements data set. *Results.* The experimental results indicate that manual inspection delivers higher recall (statistically significant with $p \leq 0.01$) and non-significantly higher precision. Based on qualitative observations, we analyze the quantitative results and suggest interpretations that explain the advantages and disadvantages of each approach. *Conclusions.* Our experiment confirms conventional wisdom in requirements engineering: identifying terminological ambiguities is time consuming, even when with tool support; and it is hard to determine whether a near-synonym may challenge the correct development of a software system. The results suggest that the most effective approach may be a combination of manual inspection with an improved version of our tool.

1. Introduction

The requirements engineering (RE) literature has extensively studied the identification and resolution of natural language (NL) requirements such as ambiguity, vagueness, and inconsistency [1–3]. The existence of numerous defects in requirements specifications is confirmed by empirical studies [3,4]. Although no strong evidence exists about the actual impact on project success [5], the field is active and there is an increasing number of tools for identifying errors, defects, and bad smells [6,7].

The identification of defects in NL requirements is not trivial. The existing automated tools make minimal reliance on human effort, but their performance is inhibited by the low maturity of NL processing (NLP) techniques. As pointed out by Cambria and White [8], current NLP technology is mostly in the syntactic "bag-of-words" curve, with some attempts to account for the semantics of the words ("bag-of-concepts"), but we are still far from the pragmatic curve in which the meaning can be exactly pinpointed depending on the context and use of the text. Current tools address this limitation by either focusing on simple tasks or making trade-offs between precision and recall [2,9,10].

Manual approaches, on the other hand, rely on the cognitive skills and analytic abilities of humans. For example, requirements inspections [11] have been proposed as a systematic approach for identifying linguistic defects by examining a specification against a set of heuristics. It goes without saying that human inspectors are an expensive resource; furthermore, the cognitive capabilities of humans do not scale well to large specifications and the effectiveness depends on the background of the inspector [12].

We advocate the synergistic use of NLP and human analysis in the context of user story requirements, a prominent notation in agile RE [13]. User stories employ a semi-structured template for expressing user requirements [14]: "As a student, I want to receive my grades via e-mail, so that I can quickly check them". Thanks to their structure, user stories can be effectively analyzed by automated NLP-powered tools; for example, our Visual Narrator tool [15] is able to automatically extract

[*] Corresponding author at: Department of Information and Computing Sciences, Buys Ballot Laboratory, Princetonplein 5, Utrecht 3584CC, the Netherlands.
E-mail address: f.dalpiaz@uu.nl (F. Dalpiaz).

https://doi.org/10.1016/j.infsof.2018.12.007
Received 1 August 2018; Received in revised form 22 December 2018; Accepted 23 December 2018
Available online 24 December 2018

concepts and relationships from user stories with high precision and recall. Unfortunately, the extracted models grow quickly and checking such models for defects is not much easier than analyzing the full text.

In our previous work [16], we have proposed an approach supported by the REVV tool that modularizes the models extracted from user story requirements by leveraging the notion of viewpoints [17]: the roles of the user stories (As a user …; As an administrator …). The visualization of REVV is inspired by Venn diagrams. REVV supports the identification of missing requirements and it highlights potential terminological ambiguities through an NLP algorithm that detects near-synonyms (e.g., car and vehicle).

Contribution. In this paper, we consolidate and extend our previous research [16] by conducting a controlled quasi-experiment with 57 participants. This quasi-experiment compares two approaches for tagging terminological ambiguity: a revised version of our tool called REVV-Light, and a pen-and-paper manual inspection of the user stories. The participants were organized into 28 groups, each group examining a real-world data set of more than 50 user stories. In addition to a quantitative comparison in terms of precision and recall, we present qualitative observations on the experiment and on the identification of ambiguities in RE.

Organization. Section 2 presents the research background for this paper. Section 3 introduces our algorithm for identifying near-synonyms in user story requirements. Section 4 describes our approach, supported by the REVV-Light tool, that combines the algorithm with information visualization. Section 5 defines the experiment in terms of scope, plan, and operation. Section 6 reports on quantitative and qualitative experimental results. Section 7 interprets the results and discusses our main findings. Section 8 contrasts our approach with related work, while Section 9 concludes the paper and presents future directions.

2. Background: from viewpoints to terminological ambiguity

Modern software systems are designed to accommodate the needs of multiple stakeholders, each of which has a somewhat different interest (stake) than those by the other stakeholders. For example, website administrators care about the creation and structuring of content, readers are mostly concerned with accessing existing content from a variety of devices, and content creators require efficient authoring tools. A *viewpoint* is a description of one stakeholder's perception of a system, and it consists of concepts and inter-relationships between them [18].

Viewpoints go hand in hand with inconsistencies and conflicts in stakeholders' requirements. Recognizing and reconciling these issues are key tasks in RE [19], and they amount to *i. intra-viewpoint checks*: assessing the consistency of the specification within one viewpoint, and *ii. inter-viewpoint checks*: verifying the consistency of the specification among different viewpoints [17].

Viewpoints may lead to ambiguity problems when the stakeholders employ different terminology and conceptual systems, i.e., ways of assigning meaning to a term [20]. Domain descriptions by different stakeholders lead to four types of relationships that depend on *i.* their chosen terminology: bank, car[1]; and *ii.* the distinctions (*denotations*) in the domain that the terms refer to: a financial institution, a ground alongside a body of water, a road vehicle [20]:

1. *Consensus*: same terminology, same distinction. Example: both experts use the term bank to refer to a financial institution.
2. *Correspondence*: different terminology, same distinction. Example: while one expert users the term car to refer to a road vehicle, another one uses the term automobile.
3. *Conflict*: same terminology, different distinction. Example: both experts use bank; one refers to a financial institution, the other refers to a ground.
4. *Contrast*: different terminology, different distinction. Example: one viewpoint examines road vehicles, the other focuses on financial institutions.

A requirement is ambiguous when it has multiple valid interpretations [21]. We argue that when a collection of requirements contains terms related by correspondence or conflict, there is a possible ambiguity in the employed terminology. Furthermore, the contrast relation may indicate missing requirements. Table 1 formalizes these concepts.

Illustration. Take the following user stories from the WebCompany data set [22].

- R_1. As a visitor, I am able to view the media gallery, so that I can see interesting photos about the event region.
- R_2. As an administrator, I am able to edit existing media elements of a particular gallery, so that I can update the content.
- R_3. As a user, I am able to add content to the selected profile.
- R_4. As a visitor, I am able to use the contact form, so that I can contact the administrator.

Consensus does not lead to any ambiguity. For example, the term administrator has the same denotation both in R_2 and R_4 and it refers to the individual who is managing the website and its users.

Ambiguity *may* occur with correspondence: distinct terms refer to the same denotation. The term media gallery in R_1 and the term gallery in R_2 do likely refer to the same denotation: a web gallery in which photographs are displayed. The problem is that most synonyms are in fact near-synonyms (*plesionyms*), as they refer to similar yet not identical denotations [23], thereby leaving the reader left to wonder if there is a difference between the two terms. This type of possible defect is the focus of this paper.

Ambiguity *may* occur also in the conflict state: the same term is used for different denotations. This phenomenon is called *homonymy*. In R_2, the term content refers specifically to a media element, while in R_3 the term content may refer to either text, descriptions, images, videos or audio fragments. We do not study homonymy as a possible source for ambiguity here.

The contrast state, instead, does not lead to ambiguity; on the other hand, it *may* indicate incompleteness, i.e., missing requirements. This happens when one viewpoint refers to a concept that does not appear in another viewpoint. R_4 includes contact form that the visitor uses to get in touch with the administrator. However, there is no other user story in our small collection that specifies how the administrator can respond to this action. We will briefly mention how our tool can be used for incompleteness in Section 4; some preliminary empirical results on its effectiveness can be found in our previous work [16].

3. NLP-powered Identification of near-synonymy

The ambiguity detection technique presented in this paper aims to detect terminological ambiguity—a defect in the category of lexical ambiguity [1]—between couples of terms for which it is unclear whether they represent the same denotation or distinct ones: this corresponds to detecting *near-synonyms*, as explained in Table 1.

To such extent, we propose an NLP-powered algorithm that integrates state-of-the-art semantic similarity techniques. This algorithm is used in Section 4 to set the background color of the terms in our requirements visualization technique, which organizes the concepts and relationships that are automatically extracted from a set of user stories.

Our NLP technique relies on algorithms that calculate the *semantic distance* between two terms: a numerical representation of the difference in meaning between two terms [24]. Current state-of-the-art NLP tools, such as Word2Vec, establish semantic similarity in the [0.0,1.0]

[1] In this paper, we emphasize terms in sansserif.

Table 1
Linking viewpoints' terminological and denotational relations [20] with possible ambiguity and incompleteness. Notation: t_1, t_2 are distinct terms, $[\![t]\!]^{V_1}$ is the denotation of term t according to the viewpoint V_1, and $\perp$ indicates absence of a denotation. For simplicity, we assume that a denotation refers to a single entity.

Relation [20]	Possible defect	Defect formalization	Example
Consensus	–	$[\![t_1]\!]^{V_1} = [\![t_1]\!]^{V_2}$	$[\![$bank$]\!]^{V_1}$ = financial institution $[\![$bank$]\!]^{V_2}$ = financial institution
Correspondence	Near-synonymy leading to ambiguity	$[\![t_1]\!]^{V_1} = [\![t_2]\!]^{V_2}$	$[\![$car$]\!]^{V_1}$ = road vehicle $[\![$automobile$]\!]^{V_2}$ = road vehicle
Conflict	Homonymy leading to ambiguity	$[\![t_1]\!]^{V_1} \neq [\![t_1]\!]^{V_2}$	$[\![$bank$]\!]^{V_1}$ = financial institution $[\![$bank$]\!]^{V_2}$ = land alongside river
Contrast	Incompleteness	$[\![t_1]\!]^{V_1} \neq \perp \wedge [\![t_1]\!]^{V_2} = \perp$	$[\![$bank$]\!]^{V_1}$ = financial institution $[\![$bank$]\!]^{V_2}$ = $\perp$

range via word statistics that compare the contexts in which a term is used [25]. The higher the similarity score, the higher the chance that the two terms have the same denotation.

Semantic similarity via fingerprinting. Our approach invokes the Cortical.io tool that employs Semantic Folding Theory [26]. This tool employs a sparse 128 × 128 matrix that is constructed as follows:

1. Given a corpus of documents, each document is split into text snippets. Each snippet is circa 1–3 sentences long and represents a single topic;
2. The similarity between two snippets is determined in terms of how many similar words they include;
3. Each snippet is associated with one cell of a 128 × 128 matrix such that similar snippets are either in the same cell or in nearby cells.
4. The *semantic fingerprint* for one word in the corpus consists of the cells in the matrix in which the word appears frequently, given some threshold.

Cortical.io uses one of such matrices created from a large collection of websites, which can be utilized to calculate *semantic similarity* as follows:

- Between two words, based on how many cells their semantic fingerprints share. Words like dog and cat will have many shared cells, which refer to snippets that include words like fur, mammal, pet, etc.
- Between two paragraphs. For each paragraph, a semantic fingerprint is calculated by merging the fingerprints of each individual word, and by removing the cells with low frequency. The similarity between the paragraphs is calculated based on how many cells their fingerprints share.

Calculating the terminological ambiguity score. Algorithm 1 takes a set of user stories and generates an ambiguity score for all couples of terms that appear in the user stories.

Algorithm 1 Computing the near-synonymy ambiguity score of term pairs.

```
COMPUTEAMBIGSCORE(Set ⟨UserStory⟩ userStories)
1   Set ⟨Term⟩ usTerms = VISUALNARRATOR(userStories)
2   (Term,Term) termPairs = (t1,t2). t1,t2 ∈ usTerms ∧t1 ≠ t2
3   Set ⟨US⟩ ctxs = ∅
4   for each term ∈ usTerms
5   do ctxs.ADD(userStories.FINDSTORIESTHATCONTAIN(term))
6   for each (t1, t2) ∈ termPairs
7   do $sim_{t1,t2}$ = SEMANTICSIML(t1, t2)
8       int i = usTerms.INDEXOF(t1)
9       int j = usTerms.INDEXOF(t2)
10      (Set ⟨US⟩, Set ⟨US⟩) pairCtx = (ctxs[i]\ctxs[j], ctxs[j]\ctxs[i])
11      $simc_{t1,t2}$ = SEMANTICSIML(pairContext)
12      $ambig_{t1,t2} = \frac{2 \cdot sim_{t1,t2} + simc_{t1,t2}}{3}$
```

In line 1, the Visual Narrator tool [27] extracts atomic nouns (e.g., car, dog) and compound nouns (e.g., cable car, sledge dog) from the set *userStories*. In line 2, all combinations of term pairs are added to *termPairs*. In lines 3–5, the algorithm constructs the context of each term, i.e., the set of all user stories that contain that term.

The loop of lines 6–12 computes the ambiguity score for each pair of terms (*t*1, *t*2). The semantic similarity of the two terms is computed in line 7; in our implementation, we use the Cortical.io algorithm based on semantic folding and fingerprints, but other algorithms are possible as well. In lines 8–10, the algorithm builds the context of each term pair: all and only the user stories in which exactly one of the two terms occurs. We exclude the user stories in which both terms occur because we assume that the analyst who writes a user story purposefully chooses the employed terms, and therefore two distinct terms in the same story are unlikely to be in a correspondence relation.

In line 11, the similarity score for the contexts of each pair of terms is computed using paragraph similarity. Finally, in line 12, the ambiguity score of two terms is computed as a linear combination of term similarity and context similarity. We currently assign a weight of 2 to the former and a weight of 1 to the latter. As explained in our previous work [16], these weights have been defined through a correlation study with human taggers and one data set. The weights resulted in a *strong* and *significant positive correlation* between the scores of the algorithm and by the participants, $r = 0.806$, $p =< 0.001$.

Illustration. Take the following set of user stories: {us1 = "As a student, I want...", us2 = "As a student, I want to print my grades...", us3 = "As a professor, I want...", us4 = "As a student, I want to check my grades and contact professors...", us5 = "As a professor, I want to upload grades..."}. In line 1, Visual Narrator is executed and it extracts the terms student, professor, and grade, while line 2 computes all pairs: (student, professor), (student, grade), and (professor, grade).

Lines 3–5 construct the contexts for each term. For example, the context for student consists of all user stories in which the term appears: {us1, us2, us4}, i.e., "As a student, I want.... As a student, I want to print my grades.... As a student, I want to check my grades and contact professors...".

Lines 6–11 calculate the ambiguity score for each pair of terms. Assume the similarity score returned at line 7 when calling Cortical.io for the pair (student,professor) is 0.34. The pair of contexts for those terms is determined in line 10 as ({us1, us2}, {us3, us5}). In line 11, the semantic similarity algorithm is launched on the pair of contexts; assume this results in a context similarity of 0.66. Finally, in line 12, the ambiguity score is determined as $(2 \cdot 0.34 + 0.66)/3 = 0.44$.

4. Visualizing requirements ambiguity and incompleteness

Building on the framework of Table 1, we design a novel requirements visualization technique for analysts to explore multiple viewpoints and to help them pinpoint possible terminological ambiguity (near-synonyms) and incompleteness. Our approach combines the NLP techniques in Algorithm 1 with information visualization principles [28] that leverage human ability.

The visualization is inspired by our previous work on the automated extraction of conceptual models from user story requirements: the Visual Narrator tool [27]. However, such a visualization proved to lead to too large models when the data set size increases, thereby creating an obstacle for the analyst who needs to conduct a thorough analysis such as the detection of ambiguities.

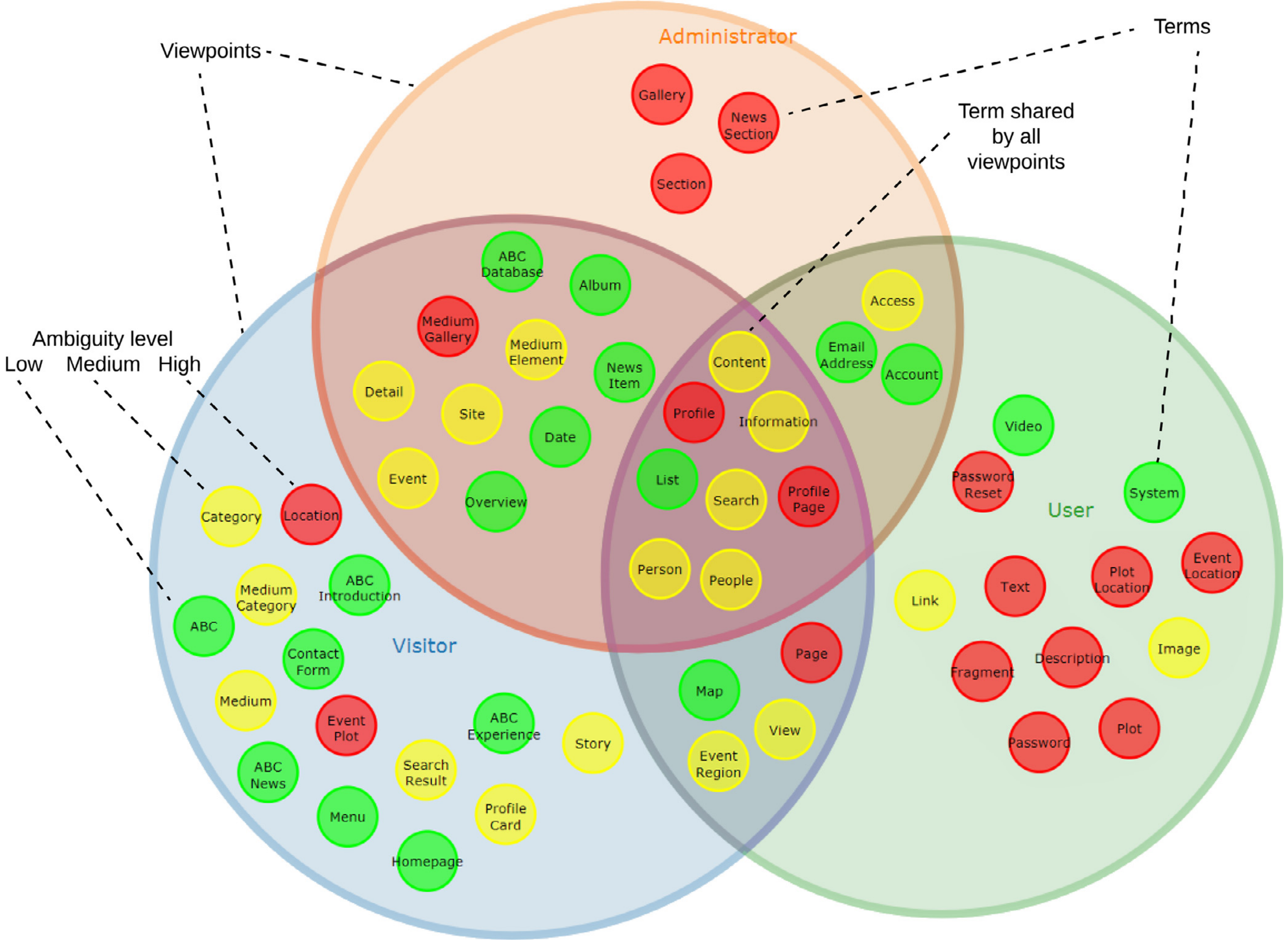


Fig. 1. Venn diagram visualization of three viewpoints and ambiguous terms.

To improve the situation, we visualize viewpoints via a Venn diagram, which is suitable for displaying overlapping elements [29]. Fig. 1 provides an example in which the terms used from three viewpoints (by the stakeholders *Administrator, User* and *Visitor*) are shown alongside their overlap.

Finding near-synonymy. The visualization highlights the possibly ambiguous terms by applying Algorithm 1. The background color of each term is set to represent the highest level of ambiguity that the term possesses with respect to another term. This high-level overview can be refined for more accurate results, as recommended by Shneiderman's *details-on-demand* principle [28].

Missing requirements and homonymy. Our approach helps an analyst explore the relationships between the terms used by multiple stakeholders. The Venn diagram in Fig. 2 illustrates the 7 areas (A–G) that originate from the analysis of 3 viewpoints.[2] There are interesting areas for the analyst to examine:

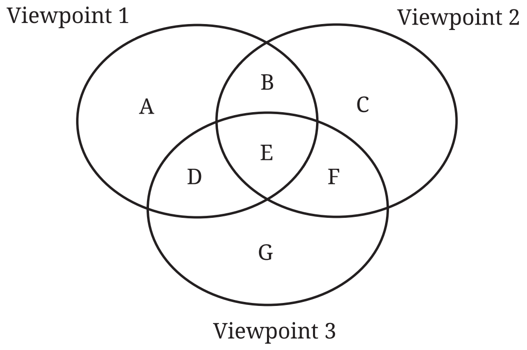


Fig. 2. The 7 areas (A–G) of our visualization applied to three viewpoints.

- All areas but E include the terms that are used either by one viewpoint (A, C, G), or by two viewpoints out of three (B, D, F). These are loci in which a missing requirement may be discovered: given a term that appears in one of such areas, the analyst should analyze if a requirement that refers to that term should be introduced for a viewpoint that is not covered by such area. In Fig. 1, for example, the term Plot appears only in the *User* viewpoint, but presumably also the *Administrator* may have some requirements about this content type.
- Areas E, B, D, F contain terms that are shared by at least two viewpoints. The instances of every term therein are in either a consensus relation (no problem) or a conflict (possible homonymy) relation. It is up to the analyst to determine which of these two relationships occurs, based on an analysis of the user stories that contain those terms.

[2] Using triangular shapes, it is possible to show six viewpoints on a 2D space [30].

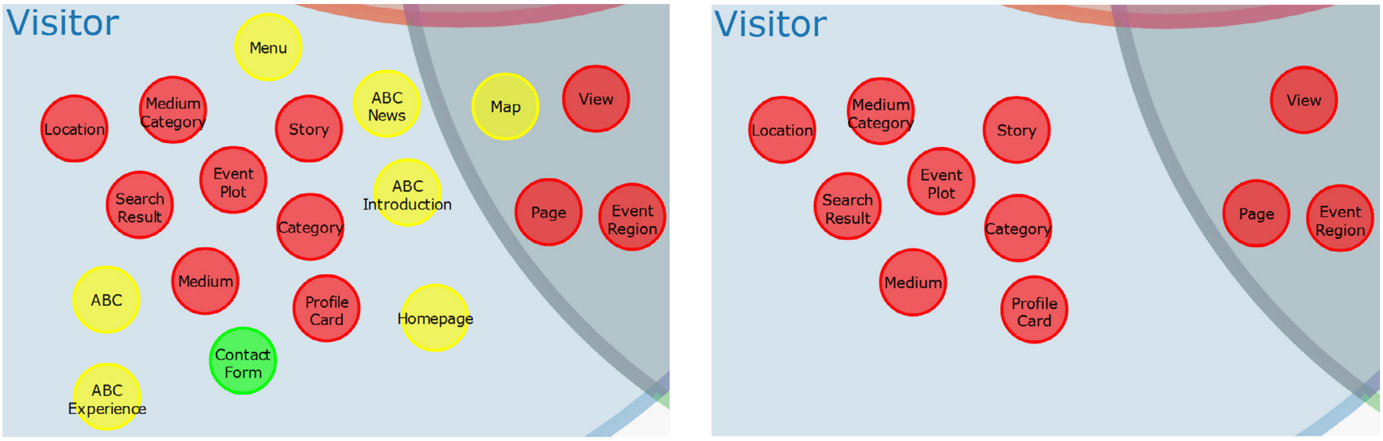


Fig. 3. REVV-Light's ambiguity filter: on the right-hand side, only terms that are part of a term pair with an ambiguity score above 0.4 are shown.

Filters. Our visualization comes with filters that can be applied to hide unwanted items from the display:

1. *Viewpoint filter* removes some viewpoints from the display, so that the analyst can focus on the remaining ones. This helps when more than three viewpoints exist, which is a common situation in practice as; see the number of roles in the data sets of Table 4 later in this article.
2. *Ambiguity filter* shows a list of the elements within a given ambiguity score range. As illustrated in Fig. 3, this list can be used to help examine the elements with high ambiguity score or to double check those with low-medium score.

Details-on-demand. These are features for retrieving additional details that are not visible through the main interface:

- *Association relationships* are the actions that a term refers to in the user stories. For example, in "As a user, I want to request a password reset", the association relationship of the term password reset is the verb request. When enabled, the association relationship is shown as a small icon next to the term. Each association relationship of a given term has a different color and is labelled with the first character of the verb. Further details can be inspected by clicking on the icon, which opens a small pop-up window. Fig. 4a shows the association relationships for some terms, and provides details for the verb request of term password reset.
- *Ambiguity inspection.* The ambiguity that a term shares with other terms can be inspected via a click. A boldface font is applied to the term label and the background is set to white, while the color of each other term is set to the ambiguity score shared with the selected term. Fig. 4b shows that the term profile page has high ambiguity with both profile and page.
- *User stories.* The user stories in which a term appears are shown in a pop-up window by double clicking on that term. The detailed term is given a black background, and other terms in those stories are given a blue background. Fig. 4c shows these details for the term access.

The REVV-Light Tool. The visualization we presented is implemented as a proof-of-concept Web 2.0 tool that embeds the algorithm for ambiguity detection of Section 3. REVV-Light is built on the Bootstrap framework, relies on the D3.js visualization library, and calls the REST API of cortical.io to compute semantic similarity. The tool is open source[3] and a demo deployment can be accessed online.[4] A screenshot of REVV-Light in action on the *CMS-Company* data set [22]—a content management system—is shown in Fig. 5.

The figure focuses on three viewpoints: *Editor* is clearly visible, *System Administrator* and *Marketeer* are only partially on the display, while *Developer, Decision Maker* and *Channel Manager* are deselected. The term Language Label is selected; Algorithm 1 indicates possible ambiguities with Language (high level of ambiguity) and Environment Language (medium level).

REVV-Light is a fork of REVV, the tool described and studied in our previous work [16]. The main differences are as follows:

- We included a faster user story pre-processing engine that checks the similarity between terms that appear in at least two different roles. While this reduces the number of suggested ambiguity instances, it significantly speeds up the pre-processing of the data sets; in our experience with REVV, this could take up to one hour for a data set with 100–150 terms;
- The experience with multiple data sets showed that an excessive number of term couples were marked as *highly ambiguous*. Therefore, we lowered the thresholds for the low, medium, and high ambiguity values. In REVV-Light, the low value is set to the [0,0.25) interval instead of [0,0.35), medium is set to [0.25,0.35) instead of [0.35,0.40), and high is set to [0.35,1] instead of [0.4,1];
- We removed some under-utilized functions from REVV such as the cluster view and the concept state filter. The change was made with the intention of simplifying the user experience.

5. Experiment: scoping, planning, and operation

We describe the design of a quasi-experiment that studies the relative effectiveness of the REVV-Light tool compared to the use of a manual inspection focused on ambiguity tagging. While the approach in Section 4 supports both ambiguity and incompleteness, the experiment investigates only terminological ambiguity. Some preliminary results on incompleteness can be found in our earlier work [16]. Our description follows the guidelines by Wohlin et al. [31].

5.1. Goal definition and context selection

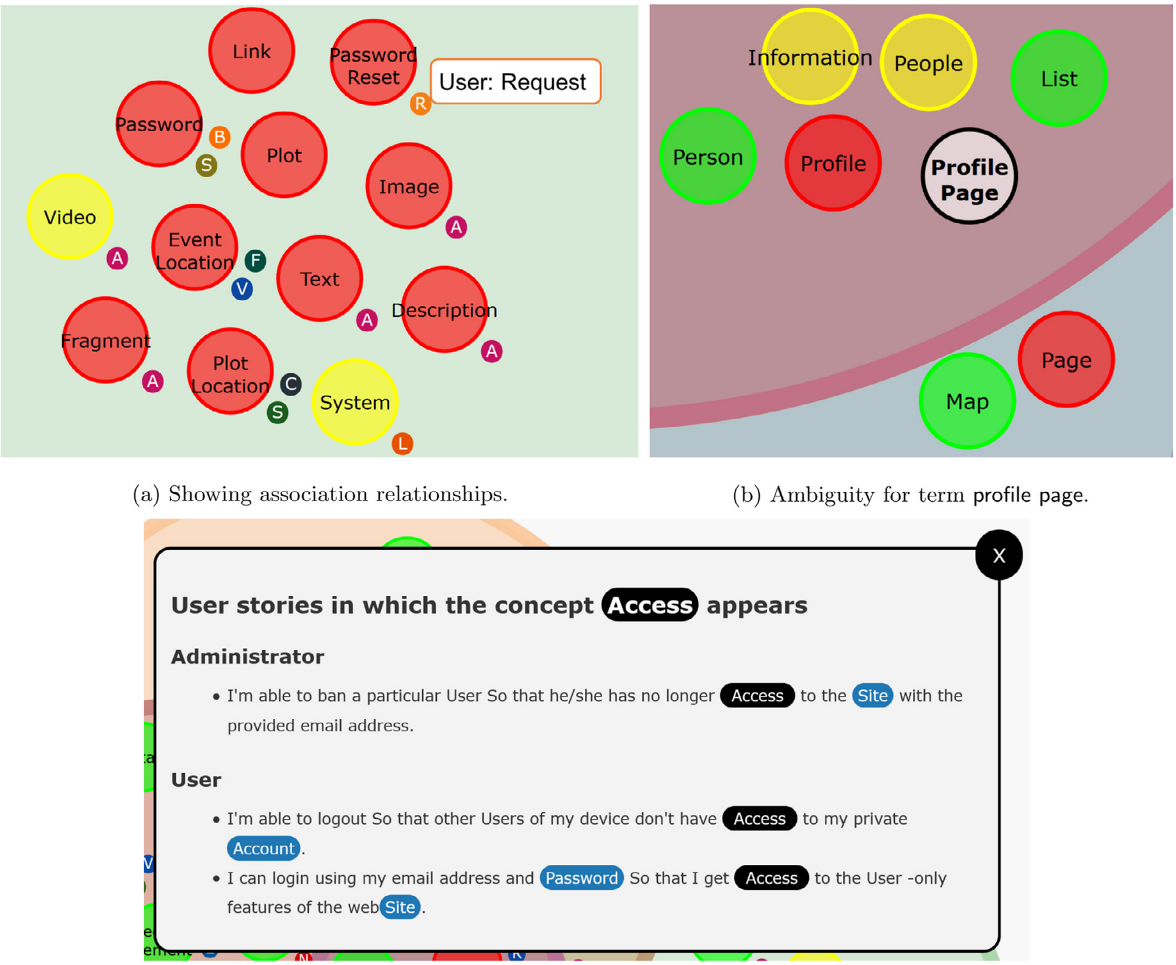
Table 2 presents the goal of our evaluation and the context selection. Note that, while the main quality focus is to analyze the effect of the treatments—pen-and-paper inspection vs. REVV-Light—on ambiguity detection precision and recall, we intend to collect a rich set of data that enables a qualitative interpretation of the quantitative results.

5.2. Planning

We detail how we planned for the experiment in line with the scope defined in the previous section and reported in Table 2.

[3] https://github.com/RELabUU/revv-light.

[4] http://www.staff.science.uu.nl/dalpi001/revv-light/.



(a) Showing association relationships.

(b) Ambiguity for term profile page.

User stories in which the concept Access appears

Administrator

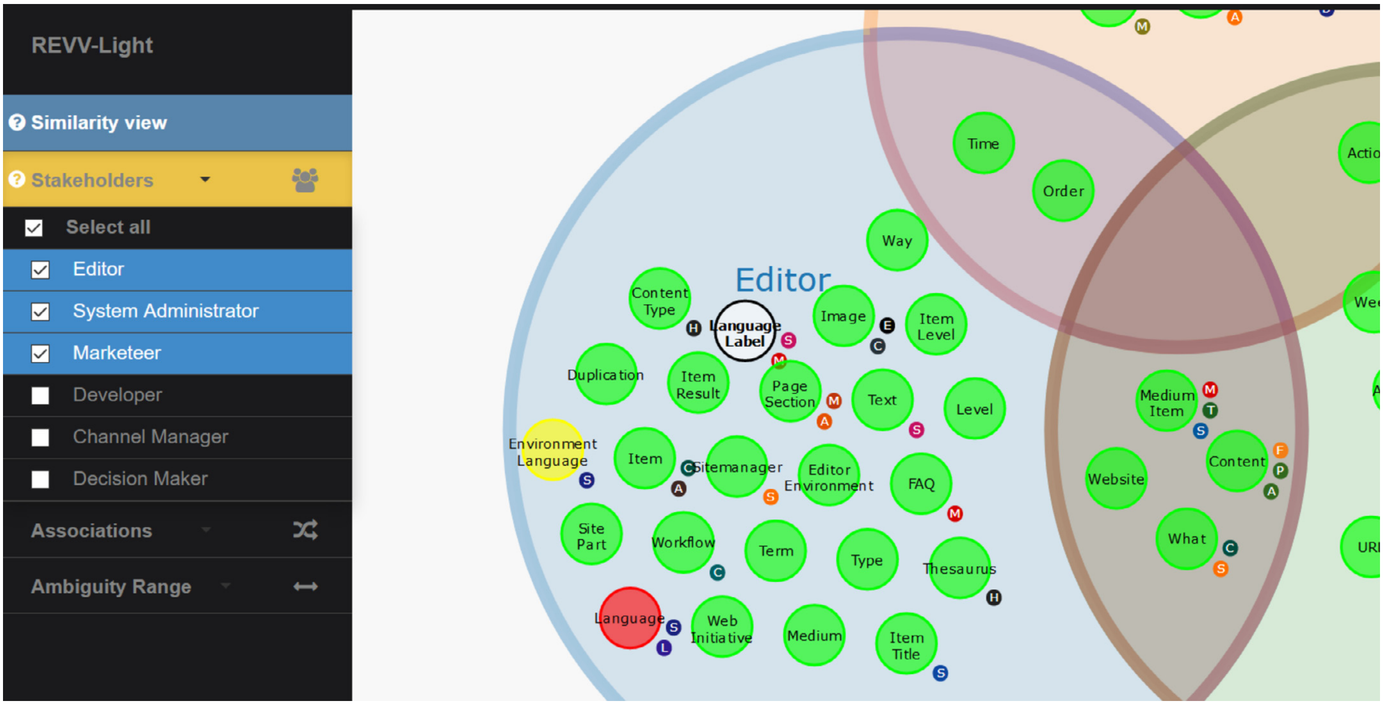
- I'm able to ban a particular User So that he/she has no longer Access to the Site with the provided email address.

User

- I'm able to logout So that other Users of my device don't have Access to my private Account.
- I can login using my email address and Password So that I get Access to the User -only features of the web Site.

(c) User stories including term access.

Fig. 4. Illustration of details-on-demand.



☑ Select all
☑ Editor
☑ System Administrator
☑ Marketeer
□ Developer
□ Channel Manager
□ Decision Maker

Fig. 5. The REVV-Light tool showing an excerpt of the CMS-Company data set.

Table 2
Goal definition for our quasi-experiment.

Object of study	We study two objects: *i.* The REVV-Light tool for identifying terminological ambiguity, and *ii.* a manual, pen-on-paper inspection of the requirements.
Purpose	Evaluate the relative effectiveness of REVV-Light compared to the pen-and-paper inspection.
Perspective	We take the point of view of RE researchers.
Quality focus	We study the *precision* and *recall* of each approach in detecting *terminological ambiguity*, i.e., the use of near-synonyms in a set of requirements.
Context	We involve 57 master's students in Information Science from Utrecht University that participate in the Requirements Engineering course. The students are organized into 27 groups of 2 members and 1 group of 3. We conduct a blocked subject-object study, for we have two objects and multiple subjects per object. Our study should be considered a quasi-experiment: while we make extensive use of randomization, the composition of the groups is decided by the participants.

5.2.1. Context selection

The experiment is conducted as part of the 2017–2018 edition of the master level course "Requirements Engineering" held at Utrecht University in the Netherlands. As such, the experiment is run off-line as opposed to being performed in the software industry. The tackled problem is a real one, that is, the ability of identifying possible terminological ambiguity in a set of requirements.

5.2.2. Hypothesis formulation

The hypotheses of this experiment stem from the results of our previous research [16]. There, we formulated four hypotheses to assess whether the REVV tool would exhibit higher precision and recall compared to a manual inspection in terms of identified ambiguities and missing requirements. The study, which was based on a single data set, led us to tentatively reject the hypotheses concerning precision, and to tentatively retain the hypotheses about recall.

These preliminary answers led us to increasing the number of data sets—to increase generality—and to focus only on terminological ambiguity in order to obtain more in-depth results, instead of conducting a broader but less thorough study of multiple defect types.

Hypotheses. In a time-constrained ambiguity detection session, couples of analysts who use the REVV-Light tool obtain a significantly higher X compared to couple of analysts using a pen-and-paper inspection, with *X* being as follows:

- *precision in finding terminological ambiguities* (H1);
- *recall in finding terminological ambiguities* (H2).

We use the information retrieval definition of precision and recall [32], for ambiguity detection—due to the size of the search space, i.e., all combinations of pairs of terms—can hardly be seen as a classification process in which failing to identify an ambiguity amounts to stating that two terms are not ambiguous. Thus, given a set of tagged couples of terms *Tagged* and a gold set of term couples *GoldSet*, precision and recall are defined as follows:

$$Prec = \frac{|GoldSet \cap Tagged|}{|Tagged|} \quad (1)$$

$$Rec = \frac{|GoldSet \cap Tagged|}{|GoldSet|} \quad (2)$$

Notes on the hypotheses. First, the time constraint for the ambiguity detection sessions is set to better resemble real-life settings, in which requirements analysts would generally devote short periods of time to detecting ambiguity in their requirements. Second, we study couples of analysts instead of individual ones to investigate whether their collaboration may lead to synergies. Third, we consider ambiguous verbs, ambiguous atomic nouns, and ambiguous compound nouns only; thus, we disregard adjectives, adverbs, and longer sentence chunks.

Table 3
Excerpt of the random assignment of the student groups SG01–SG28, who provided data sets DS01–DS28, respectively, to the experimental task roles T1–T4.

Data set ownership		Experimental task roles			
Data set	Data owner	T1	T2	T3	T4
DS01	SG01	SG09	SG10	SG18	SG06
DS02	SG02	SG03	SG17	SG15	SG23
...	...	...	...	...	...
DS27	SG27	SG08	SG11	SG16	SG19
DS28	SG28	SG02	SG19	SG06	SG26

5.2.3. Variables and subjects selection

The independent variable is the treatment used: manual inspection vs. REVV-Light. The dependent variables are precision and recall.

We selected subjects based on convenience: the subjects are the 57 students who participated in the 2017–2018 Requirements Engineering course in the period April 2018–June 2018. The subjects did self-organize into 28 groups (SG1–SG28): 27 groups of two students each, 1 group of three students.

5.2.4. Experiment design and instrumentation

The design of our experiment is illustrated by the BPMN diagram of Fig. 6. In the following, we describe each step, and we report on the instruments that are chosen to conduct the experiment.

Each student group played two main types of activities throughout the process:

- *Research*: after retrieving a data set and documenting it, they moderated the inconsistency resolution session, organized the time-constrained tagging (manual inspection and with REVV-Light) by observing the participants and by conducting a follow-up interview, and analyzed the results for their data set based on the four tagging sessions. This role is labelled as data owner – DO in Fig. 6.
- *Tagging* terminological ambiguities identified in four different data sets. First, each student group tagged two data sets without time constraints (roles T1 and T2). Then, they participated in the time-constrained sessions using manual inspection with one data set (role T3), and using the REVV-Light tool with another data set (role T4).

The data sets were assigned randomly by the lecturer with the constraint that each student group would analyze any one data set at most once when playing roles T1, T2, T3, and T4. Table 3 shows an excerpt of the assignment. The lecturer is the first author of this paper.

Data set search. Each of the groups was given 2 weeks to retrieve a real-world data set including at least 50 user stories, prepare a 1-page description of the context, and obtain additional materials such as test cases or user guides, if available. Preference should be given to publicly available or publishable data sets. As soon as a group had identified a data set, the lecturer of the course was contacted in order to get the data set approved and to avoid duplicates. An overview of the collected data

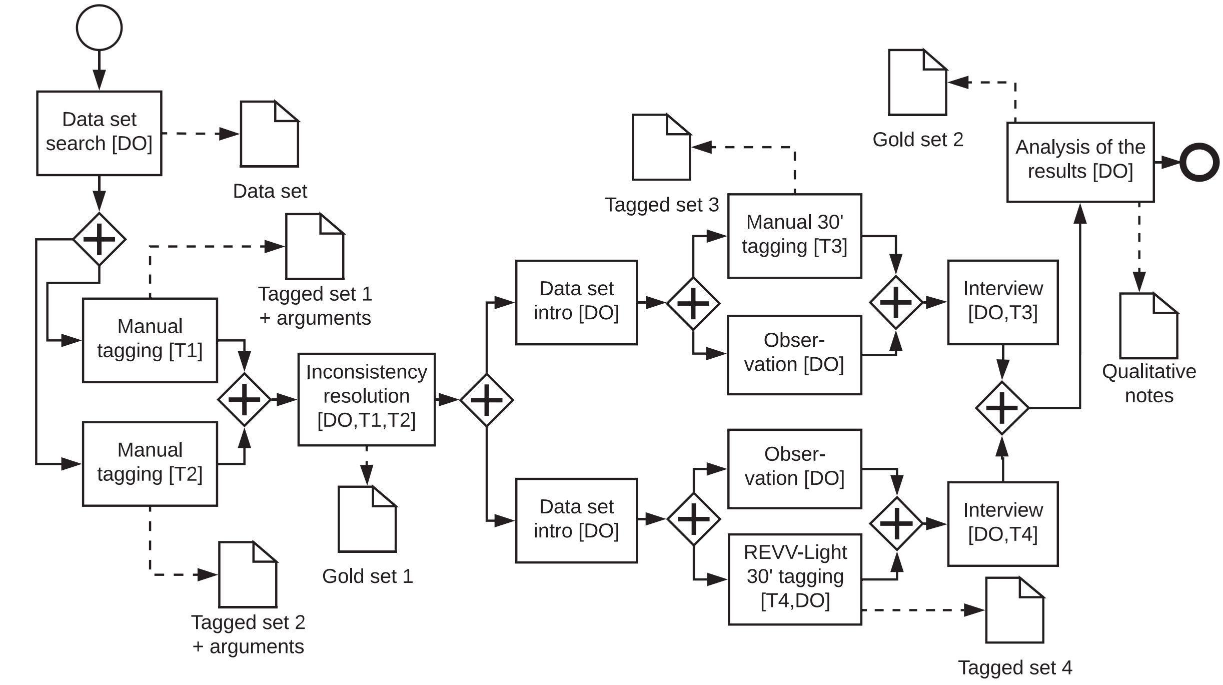


Fig. 6. BPMN diagram of the experimental protocol for one data set. The following labels denote the roles the student groups play: DO is the data owner, i.e., the group who retrieved the user stories; T1 and T2 are the groups who independently tagged the data set without time constraints; T3 is the group that did the manual, time-constrained tagging; T4 is the group that did the time-constrained tagging with REVV-Light. For readability, only output data flows are shown.

Table 4
Overview of the 28 data sets (in total, 2067 user stories).

ID	Acronym	Description	Size	Roles	Terms	Public
DS01	ElectronicCorp	Performance reporting for a multinational electronics company	66	22	145	
DS02	FederalSpending	Online platform for delivering transparent information on US governmental spending	98	11	118	✓
DS03	Loudoun	Electronic land management system for the Loudoun County, Virginia	58	8	107	✓
DS04	Recycling	An online platform to support waste recycling	51	11	86	✓
DS05	Openspending	Website for create a transparent overview of governmental expenses	53	7	85	✓
DS06	BeerApp	App for managing machinery in a large beer brewery	62	3	85	
DS07	EnergyCorp	App for supporting the switch from one energy provider to another	60	21	118	
DS08	FrictionLess	Platform for obtaining insights from data	66	10	69	✓
DS09	Students	Business rules management for student financing	85	4	88	
DS10	ScrumAlliance	First version of the Scrum Alliance Website	97	15	115	✓
DS11	NSF	New version of the NSF website: redesign and content discovery	73	21	115	✓
DS12	CamperPlus	App for camp administrators and parents	55	4	56	✓
DS13	PlanningPoker	First version of the PlanningPoker.com website	53	6	53	✓
DS14	DataHub	Platform to find, share and publish data online	67	8	63	✓
DS15	Fleet	Reporting tool for fleet management	58	10	75	
DS16	MIS	Management information system for Duke University	68	13	132	✓
DS17	CASK	Simplified toolbox to enable fast and easy development with Hadoop	64	9	67	✓
DS18	NeuroHub	Research data management portal for the universities of Oxford, Reading and Southampton	102	10	119	✓
DS19	Alfred	Personal interactive assistant for independent living and active aging	138	6	126	✓
DS20	CyberSec	Cybersecurity and vulnerability management platform	56	8	80	
DS21	BADCamp	Conference registration and management platform	69	9	106	✓
DS22	RDA-DMP	Software for machine-actionable data management plans	83	27	115	✓
DS23	Archivesspace	Web-based archiving information system	57	12	72	✓
DS24	UniBath	Institutional data repository for the University of Bath	53	11	89	✓
DS25	Duraspace	Repository for different types of digital content	100	3	88	✓
DS26	RACDAM	Software for archivists	100	5	117	✓
DS27	CULRepo	Digital content management system for Cornell University	115	19	173	✓
DS28	Zooniverse	Citizen science platform that allows anyone to help in research tasks	60	3	82	✓

sets is presented in Table 4. Overall, the 28 data sets include 2067 user stories. We turned the 22 data sets that are not confidential into a public data set on Mendeley Data [33].

Manual tagging. When playing roles T1 and T2, each group had to manually tag the ambiguities found in two data sets assigned by the lecturer according to the scheme shown in Table 3. The student groups were given one week to perform such task, and possessed a copy of the auxiliary material collected by the data set owner. As stated in Section 5.2.2, the focus was on identifying ambiguous verbs as well as ambiguous nouns. Besides listing the identified ambiguities, each group provided arguments for marking a couple of terms as an ambiguity.

Inconsistency resolution. The lecturer sent the results of the first tagging to the data set owner. For example, as per Table 3, the reports of SG09 and SG10 (playing roles T1 and T2 for data set DS01) were sent back to the data owner SG01. Then, the data set owner organized a session, together with the groups playing T1 and T2 for that data set, intended to resolve inconsistencies in the tagging. The lecturer instructed the groups to follow a protocol in which the taggers would try to resolve discrepancies based on their own arguments, and the data set owner would intervene only when an agreement cannot be found. This activity results in the first version of the gold set.

Treatments comparison. The preliminary manual tagging activities described so far aim to incrementally construct a reliable gold set, and are followed by the controlled quasi-experiment itself. The process in Fig. 6 forks: the upper flow involves a group, playing role T3, that performs manual tagging; the lower flow involves a group, playing role T4, that is assisted by REVV-Light.

Some activities are common to both flows. First, the data set owner gives a short introduction of the data set, either orally or through a short written description. Then, the actual tagging session takes place, and lasts for 30 min with the data set owner acting as an observer. Finally, an interview takes place in which the taggers provide their opinion on the exercise, the sentiment toward the employed method, and the strategy used to conduct the tagging.

The main differences between the treatments concern the instrumentation:

- Manual inspection (T3): the group members are given two printed copies of the user story collection, two markers with different colors, two pens with different colors, and one notebook with a mouse.
- REVV-Light (T4): the group members are given an instance of REVV-Light pre-configured with the user story collection running on a notebook connected to a 22-inch screen, and a mouse.

The taggers are left free to decide how to collaborate to identify ambiguities, and all the students were requested to familiarize with the tool prior to the experiment. All tagging sessions are conducted in similar rooms reserved and prepared by the lecturer.

Analysis of the results. The data set owner takes as input the first gold standard, the tagged ambiguities by the groups playing T3 and T4, and consolidates the results into the final version of the gold standard that is used in this paper to calculate precision and recall. Moreover, the data set owner delivers also a report to the lecturer that includes qualitative notes taken from the observations and from the interviews.

5.2.5. Validity evaluation

We discuss the main threats to validity by explaining their possible effect as well as how we attempted to mitigate them.

Internal. One important threat concerns maturation, for the subjects had significantly more expertise with manual inspection than with the REVV-Light tool. Indeed, each group performed manual tagging twice prior to the actual experiment playing the roles T1 and T2. Furthermore, instrumentation threats exist because we did not fully control the setting in which the experiments were conducted; the data set owner group was free to decide how the groups playing roles T3 and T4 would report the identified ambiguities: orally, on a spreadsheet, pen on paper.

External. The setting in which manual inspection is compared against REVV-Light is not representative of real settings. Although we tried to emulate the lack of time for ambiguity detection by defining a short session, it is more likely that ambiguity is identified incrementally in practice.

Conclusion. Despite the many data sets (28), we have low statistical power for each individual data set, for only one team used a particular time-constrained approach on that specific data set. The reliability of measures is a difficult aspect: the notions of terminological ambiguity and near-synonyms were explained in the lectures and in the task assignment, yet different interpretations are very likely to exist. A threat of random heterogeneity exists: although all the students are master's students in information science, their skills and commitment vary. The random assignment is likely to mitigate the threat, but it does not remove it altogether.

Construct. The REVV-Light tool was not explained extensively and the experience with the tool was not tested prior to the experiment. Similarly, our definitions of terminological ambiguity depend on whether the participants find that two terms are possible synonyms, but this may depend on the domain, on the experience of the participant, and on her English language proficiency. This threat is partially mitigated by relying on groups of two students, but it still exists. Furthermore, all groups have used both approaches on different data sets: it is possible that their strategy for tagging cannot therefore be fully ascribed to a single treatment. Evaluation apprehension is a minor threat: the tagging activity was not graded, but this is still part of a course assignment, and this may have influenced the performance of some participants. Finally, some small changes were applied to make the data sets adequately processed by the tools; see Section 5.3 for details.

5.3. Operation

The experimental process of Fig. 6 was executed between April 23, 2018 and June 24, 2018. During that period, the 57 participants were taking part in the Requirements Engineering course. They were not made aware of being part of an experiment until they adopted the roles T3 and T4. Although the hypotheses were not revealed, they could be easily deduced from the fact that the lecturer is one of the authors of the REVV approach.

Every major step of the process was reported by the students as a graded assignment. The data set search lasted from April 23 to May 10, and resulted in a report on the data set, background information, additional documentation like test cases or a glossary, and an explanation of why the data set would be interesting. For the preliminary manual tagging (roles T1 and T2), the students were given one week: May 11 to May 18. Each group reported the tagging conducted on the data sets it received from the lecturer—thus, not its own data set. The inconsistency resolution took place from May 21 to June 4, and the students were responsible for arranging the session. The corresponding report described the first gold set and elaborated the rationale for the decisions. The manual inspection vs. REVV-Light experiment took place from June 5 to June 24. The rooms and the schedule were arranged by the lecturer and the 56 sessions took place from June 11 to June 18, with one exception on June 21. The report, written from the perspective of the data set owner, included the ambiguities found by the groups playing roles T3 and T4, the final gold set, observations on the experiments and notes about the conducted interviews, and a reflection on the entire project and ambiguity in RE.

Minor data set changes. Prior to the tagging sessions with the groups playing T3 and T4, some syntactic changes have been made to the original data sets to make the user stories automatically analyzable by the Visual Narrator tool. The students were instructed to avoid modifications that may alter the semantics of a user story, introduce or mitigate ambiguity. Some examples of the changes: *i.* spelling mistakes and typos were corrected (e.g., in DS01 and DS04); *ii.* the "I want to" indicator was introduced to replace "I want" (for example, in DS04, "I want the website to be easy to use" became "I want to have an easy-to-use website"; *iii.* the ends indicator was rephrased to "so that" followed by a noun or a pronoun (e.g., in DS07, "...to comply to legislation" becomes "...so that I comply to legislation”; *iv.* some user stories that included multiple functions were split such as, in DS16: “As a collection curator, I want to be able to set a date after which data will expire and be deleted or hidden then deleted. I would like the scheduled records deletion date to be displayed on the item and and component pages”.

Aggregation of roles. A more significant modification regards the number of roles. When possible, the students were encouraged to merge some roles, especially when the user stories included 15+ roles and when too few user stories belonged to a specific role. Some roles were renamed due to a bug of the Visual Narrator, which does not support long role names; for example, “someone working on the NSF project” was renamed to “NSF employee” (DS11), while “collection curator housed in the Protected Data Network” became “collection curator” (DS16). Some user stories contained multiple roles, e.g., “As a user/administrator”; in those cases, sometimes only one role was kept (DS11), or they were rephrased, e.g., “Librarian/member of the library staff” became “Library staff member” (DS27).

6. Experimental results

We present the quantitative and qualitative results for the conducted experiment. The interpretation is left to the following section.

The qualitative tagging was done in NVIVO 12 Professional starting from the assignments that the students delivered after the experiment. The identified ambiguities were re-coded into an Excel spreadsheet [34] and the measurements of precision and recall were re-calculated by the authors of this paper using the Excel Data Analysis add-in, and relying on SPSS 24 for executing Levene’s test for the homogeneity assumption when comparing groups of different size.

6.1. Quantitative results

We extracted the ambiguous terms, their corresponding part of speech tag (noun, noun phrase, verb, verb phrase), and the final gold set from the reports of the groups. For each data set, we identify true positives (TP), false positives (FP), and false negatives (FN). True negatives are ignored given that we evaluate precision recall in information retrieval terms. Eqs. (1) and (2) are reformulated in terms of TP, FP, and FN by Eqs. (3) and (4):

$$Prec = \frac{|TP|}{|TP + FP|} \tag{3}$$

$$Rec = \frac{|TP|}{|TP + FN|} \tag{4}$$

We do not calculate the F-score because, for the terminological ambiguity identification task, we have no evidence that allows us to quantify the relative importance of precision and recall. As such, we cannot determine an appropriate value for the β variable to allow us to use a meaningful F_β [35].

Table 5 reports the macro-averages (AVG) and the standard deviation (SD) of these measurements for *i.* nouns, noun phrases, verbs, and verb phrases (All), *ii.* only for noun and noun phrases (N & NP), *iii.* only for verb and verb phrases (V & VP). The macro-averages are computed by independently calculating the values for each data set first, then taking averages of these values, hence treating each data set the same regardless of the number of identified ambiguities or the user stories in the data set.

Table 5
Overview of the macro-averages of the experiment, and results of the *t*-test.

Measure		Manual		REVV-Light		*t*-test	
		AVG	SD	AVG	SD	*t*(54)	*p*
All	Prec	0.60	0.24	0.51	0.24	1.54	0.13
	Rec	0.37	0.19	0.25	0.16	2.53	0.01**
N & NP	Prec	0.54	0.42	0.45	0.32	0.95	0.35
	Rec	0.27	0.26	0.26	0.23	0.15	0.88
V & VP	Prec	0.44	0.33	0.46	0.41	−0.25	0.80
	Rec	0.39	0.33	0.24	0.30	1.87	0.07

** $p < 0.05$

The results presented in Table 5 trivially reject our hypotheses stated in Section 5.2.2; the couple analysts who use REVV-Light tool outperforms the analysts using pen and paper (manual) only in one case in which they identify ambiguous terms that are verbs or verb phrases. It is clear from the data that REVV-Light does not yield to significantly higher results in precision and recall.

Next, we test the hypothesis that there is no difference in the average results for the manual and REVV-Light tool approaches using *t*-tests. The results show statistical significance with the manual inspection outperforming REVV-Light in the overall recall with $p < 0.05$.

6.2. Qualitative results

We organize the qualitative findings from the experiment according to three main aspects: *i.* the tagging strategy employed by the taggers (Section 6.2.1); *ii.* the main obstacles that were encountered with each treatment (Section 6.2.2); and *iii.* the sentiment toward the REVV-Light tool (Section 6.2.3).

6.2.1. Tagging strategy

Manual inspection. We could identify three main strategies that the participants employed in the manual inspection, i.e., when playing role T3:

- *Redundant tagging*: both group members worked individually, and discussed the ambiguities after their identification. This was the predominant strategy and we could observe three variants: *i.* discussion of the results at a pre-defined time instant such as 20 min after starting the tagging (12 groups); *ii.* discussing the results at certain intervals (2 groups); and *iii.* discussing every ambiguity as soon as it was identified (4 groups).
- *Splitting the data set*: each group member focused on non-overlapping tasks, either by splitting nouns and verbs (2 groups) or by analyzing the data set in orthogonal directions (from the top and the bottom, 2 groups).
- *Collaboration*: the group members examined together the same user stories (2 groups), i.e., did not split the task.

The strategy for three groups could not be clearly deduced from the observations in the student reports, although the notes seem to indicate redundant tagging. For one group, only one participant could attend the session.

REVV-Light. The observations concerning the tool treatment indicate different non-orthogonal ways for using REVV-Light when playing role T4:

- At least 13 groups used explicitly the ambiguity score to guide their inspection process, either looking at the colors of the circles (10 groups) or removing terms using the ambiguity filter (3 groups). At least three groups, on the other hand, deliberately chose to ignore that information.
- At least 8 groups made use of the associations to identify possible terminological ambiguities concerning verbs.
- At least 7 groups spent some time playing with the set of visible roles and trying to identifying an ideal combination that they deemed optimal for identifying ambiguities.
- Concerning the number of roles that were displayed concurrently, we can observe three strategies: working with a handful of roles at a time (4 groups), showing all stakeholders (3 groups), and progressively deselecting a role after all its terms were studied for ambiguity (4 groups).

- At least three groups explicitly examined in detail an ambiguity by reading carefully the associated user stories.
- At least three groups used also a printed copy of the user stories, first looking a the printouts, and using REVV-Light as an additional tool.

6.2.2. Obstacles

A number of difficulties were observed and reported concerning the conduction of the experiment with either treatment.

Search space size. The vastness of the search space—quadratic with the number of terms—was reported as a significant obstacle with both approaches. The lack of a holistic visualization of the data set was an issue for at least 7 groups doing the manual inspection (T3). At least 10 groups indicated that a digital search function (CTRL + F) would have been greatly beneficial. Also the groups using the tool (T4) experienced some issues; in particular, they would have liked to see a list of user stories (6 groups), and have user story identifiers (2 groups).

Domain knowledge. This was another key obstacle. 3 groups conducting manual inspection and 4 groups using REVV-Light stated that the lack of domain knowledge makes it hard to assess whether an ambiguity is genuine. The use of a dictionary, which was hardly possible due to time constraints, was mentioned as a limitation by a few groups: 2 groups playing T3 and 1 group playing T4.

Time pressure. The short duration of the experiment led inevitably to time pressure. 5 groups doing the manual inspection and 2 groups using REVV-Light groups mentioned this obstacle explicitly stating that some ambiguities may have been missed out or that the discussion on some tagged ambiguities could have been extended.

Performance issues of the tool. A recurring problem with the REVV-Light tool concerned its performance; the use of web animations and the heavy reliance on Javascript made the loading of the terms and their coloring slow (7 groups). The observations indicate that the speed depends on laptop, browser, and number of roles and terms. Furthermore, the introduction of hotkeys was found necessary by one group to overcome the necessity of clicking the reset button on the top-right of the screen in order to restore the visualization to the default ambiguity coloring.

Tool bugs. REVV-Light is a proof-of-concept tool and, unsurprisingly, some bugs were identified as an obstacle by the participants. A recurring issue, explicitly reported by three groups is that the tool assigns colors to *potential* ambiguities, but the visualization is perceived by the users as *real* ambiguities. This is not a bug per se—Algorithm 1 is a heuristic—but creates unrealistic expectations in the users. An easier-to-fix bug is that some terms are shown outside the role containers (2 groups). Additional bugs reported by a single group are the difficulty of handling abbreviations, the omission of some terms when many roles are visualized, the inability to properly highlight some compound nouns in the user stories, and the effects of stemming on the comprehensibility of some terms (e.g., data becomes datum).

6.2.3. Sentiment

The interviewed participants reported a range of sentiment types toward REVV-Light as an instrument to identify terminological ambiguity.

Positive. Two groups found the tool useful when the requirements data set is large, and one group highlighted that the tool's main benefit is that it enables the analyst to not read all the user stories. The members of one interviewed group found the tool intuitive and declared their intention to use it. Three groups expressed a generic positive impression about the tool. Finally, one group appreciated the ability of the tool to extract and pinpoint the verbs, which can be quickly scanned to identify ambiguities.

Mixed. Some participants expressed mixed feelings about the tool because of two causes: *i.* the tool is an interesting concept but it is not sufficient to detect all ambiguities efficiently: it should rather be seen as complementary to manual tagging (5 groups); *ii.* REVV-Light is prototypical but it would become the preferred option had it higher performance and precision (3 groups).

Negative. Several participants stated mostly negative feelings about REVV-Light, and either challenged the effectiveness of the approach itself, or pointed out technological issues, as shown in Section 6.2.2. The major *conceptual* criticism, pointed out by two groups, is that the tool creates a tunnel vision that pushes people to focus on the colors and words without considering the context in which they occur. A generic preference for manual tagging was mentioned six times. The major *technological* concern relates to the limited precision of the NLP algorithms (5 groups), often due to the low recall. Only two groups mentioned explicitly that the low performance leads to low satisfaction, but Section 6.2.2 shows that this aspect was a problem. Finally, one group found the tool too difficult to use.

7. Interpretation and discussion

The quantitative results clearly reject our hypotheses H1 and H2: the REVV-Light tool does not lead to significantly higher precision and recall than a manual inspection in identifying terminological ambiguity. Nevertheless, the qualitative results from Section 6.2 can be used to interpret the raw numbers. We organize our analysis of the results into four main categories.

Ambiguity tagging is time consuming. Time pressure was explicitly mentioned as a challenge by a few groups who participated in the experiment, as shown in Section 6.2.2. Furthermore, only a few taggers thought their tagging was complete, and most reports conclude that the final gold standard is incomplete. As a comparison, the participants playing roles T1 and T2 spent roughly 2–3 h on each data set. These insights confirm conventional wisdom in the RE practice; ambiguity detection is not a common activity due to the high cost and the uncertain return on investment. A possible solution is to use interactive tools that identify defects on-the-fly during requirements authoring [36].

Recognizing true ambiguities in RE. It is difficult to execute a reliable investigation of the effectiveness of an approach for ambiguity tagging in RE. First, the search space is vast, for one would have to compare each possible couple of terms in the data set. Second, domain knowledge is essential to pinpoint true ambiguities, but assuming that all team members have perfect knowledge is an unlikely-to-hold assumption. Third, while the notion of ambiguity is well defined in linguistics, one would actually want to identify only ambiguities that have an impact on the RE process or in later software development phases. These difficulties are clearly evidenced by the ambiguities tagged by the participants: out of over 1032 ambiguities, only 2 are shared by the groups playing T1, T2, T3, and T4 for the same data set. The challenge is also confirmed when computing Fleiss' Kappa, which indicates a *poor* agreement between the four raters ($k = -0.205$, $p \leq 0.001$).

Experience matters. The student reports concerning the execution of the experiment evidenced that some participants had a different level of experience with manual tagging and with REVV-Light. As already commented in the validity evaluation in Section 5.2.5, the taggers gained experience with manual inspection when playing roles T1 and T2. On the other hand, the participants had limited experience with the REVV-Light tool. This observation was confirmed by the reports: at least 8 groups exhibited little to no experience, and were unaware of basic functionality such as the possibility to visualize the user stories in which a term occurs.

Table 6
T-test for the equality of means between low experienced groups and the others.

Measure		Low Exp.		Others		*t*-test	
		AVG	SD	AVG	SD	*t*(26)	*p*
All	Prec	0.45	0.22	0.53	0.25	−0.85	0.41
	Recall	0.21	0.12	0.27	0.18	−0.96	0.35
N & NP	Prec	0.37	0.27	0.49	0.34	−0.97	0.34
	Recall	0.20	0.19	0.30	0.25	−1.07	0.29
V & VP	Prec	0.24	0.37	0.59	0.39	−2.34	0.03**
	Recall	0.22	0.36	0.25	0.27	−0.22	0.82

** $p \leq 0.05$

Thus, we decided to conduct an additional *t*-test that compares the 8 groups with low demonstrated experience with REVV-Light with the other 20 groups. Due to the uneven size of the groups, we first ran Levene's test to assess the normality of variances [37]; normality was confirmed, and we could therefore execute the *t*-test assuming equal variance. The results are shown in Table 6, and they indicate that the teams with higher experience consistently obtained higher precision and recall in all cases: overall, for nouns and noun phrases, and for verbs and verb phrases. However, statistical significance is obtained ($p \leq 0.05$) only for the precision of verb and verb phrases.

The difference concerning the V & VP class prompted us to consider another observation made in Section 6.2.1: at least 8 groups explicitly made use of the associations to identify possible ambiguities concerning verbs. We tested for equality of means the groups who used associations and the others—we could, again, assume equal variance after running Levene's test—and this led to the results shown in Table 7. The *t*-test for A vs. O shows a significant difference in precision and recall for the class V & VP. This seems to indicate that using the association filters significantly improves the performance of analysts who use REVV-Light compared to those who use the tool but do not use such function. We further tested how the groups using the association filter would compare to the groups doing manual inspection. The *t*-test for A vs. M highlights a statistically significant difference only in the precision.

Synergies between the treatments. In absolute terms, none of the treatments obtained excellent results; in particular, when we consider recall, the manual inspection achieved an average of 0.37, while REVV-Light obtained an average of 0.25. The precision results are a bit higher: 0.60 for the manual inspection, and 0.51 for REVV-Light. The qualitative observations denote, however, fundamental differences: while REVV-Light creates a tunnel view that hides the context in which the terms occur (but creates an overview of the terms!), the manual inspection suffers from the opposite problem, as the tagger is confronted with the entire search space and has no overview of the data set. These properties suggest that an improved tool for tagging terminological ambiguity should combine the strengths of both approaches. The participants suggested, for example, that REVV-Light could be improved with a visualization of all the user stories on a side of the screen, or that the terms suggested by Algorithm 1 could be visualized directly on the list of user stories. This is a research direction that we intend to follow, with the overall aim to conduct research that has a positive impact on the RE practice.

8. Related work

Ambiguity in RE. Several studies on ambiguity in RE have been conducted over the past twenty years. The seminal contribution of Berry and Kamsties [1] provides an authoritative overview of the main categories of ambiguity and their relevant for RE, including lexical—investigated in this paper—, syntactic or structural, semantic, and pragmatic. Their work has the merit of bringing theories from linguistics to the RE field. Since then, researchers have proposed numerous approaches to cope with different types of ambiguity.

Tjong et al. [38] built an ambiguity-detection tool called SREE: the Systemized Requirements Engineering Environment. SREE aims to achieve 100% recall in the identification of weak terms based on a dictionary of such terms. We focus also on lexical ambiguity; however, we investigate a different kind for we focus on near-synonyms, and our approach does not make use of a dictionary.

Several authors focused on syntactic or semantic ambiguity. Willis and colleagues [39] introduced the notion of nocuous ambiguity as opposed to harmless ambiguity, and propose an automated approach for identifying coordination ambiguities. Yang and colleagues studied anaphoric ambiguity: the use of pronouns such as it, them and their [40]. They built a classifier that identifies instances of anaphoric ambiguity and tries to predict whether the referenced noun is unclear, i.e., if the anaphora is nocuous.

Kiyavitskaya et al. [41] conduct a meta-study that results in a set of requirements for an effective ambiguity detection tool. They propose a two-step approach that combines two tools: the first tool is used to identify potentially ambiguous sentences in a requirements specification, while the second tool would show what is potentially ambiguous about each of the sentences identified by the first tool. To the best of our knowledge, there are no full implementations of such a concept. Our tool has a less ambitious aim.

Ferrari et al. [42] studied the notion of pragmatic ambiguity that depends on the background of the reader. They present a method that, given a graph model of the domain, provides the different interpretations of a requirement according to such graph model, and compares the interpretations. The extraction of a domain knowledge graph from domain documents could be used to enrich our ambiguity score algorithm, which uses a domain-independent corpus.

InfoVis for RE. The systematic literature review by Abad et al. [43] classifies existing approaches in requirements engineering visualization along the RE activities they support, the involved stakeholders, and the focus on the problem or solution domain. The review organizes the existing papers into the following categories: requirements evolution, requirements communication, requirements inspection, requirements planning, and non-functional requirements. According to Abad's framework, our work supports the *requirements verification* activity, it focuses on the *problem domain* by analyzing the stakeholders' needs, and it is intended for *decision makers*.

Among the existing visualization approaches, a similar approach to ours is taken by Savio et al. [44], who propose a 3D pyramidal visualization in which each face of the pyramid represents one stakeholder, and the pyramid is sliced along the z-axis to denote different levels of refinement of the requirements. However, their approach does not focus on terminological ambiguity.

Reddivari et al. [45]'s RecVisu + tool organizes requirements graphically into clusters based on their similarity, it includes an algorithm for automated cluster label generation, and it supports manipulating the requirements during their elaboration. Besides the different purpose, it is interesting to observe that our work takes an orthogonal approach: the atomic elements in REVV-Light are the terms instead of the requirements, and the analyst can then inspect the corresponding requirements by requesting details, as shown in Fig. 4c.

Other researchers propose different uses of information visualization in RE. Duarte et al. [46] discuss how to use multiple visualization techniques—including motion charts, treemaps, tag clouds, and fusion charts—to involve stakeholders during requirements elicitation. Agarwal and colleagues visualize the results of theme-based release planning in terms of clustering techniques [47]. Wnuk et al. [48] tackle the problem of visualizing large-scale requirements using feature survival charts (FSC +), and apply the technique to a large company with thousands of features.

In our previous work [49], we proposed a cluster-based visualization of the terms extracted from user story requirements. Differently, REVV-Light does not aggregate the terms via clustering, but rather

Table 7
T-test for the equality of means between the groups that used REVV-Light with and without paying attention to the associations (A vs O), and between the groups using REVV-Light paying attention to the associations and the groups doing manual inspection (A vs M).

Measure		REVV-Light				Manual		*t*-tests			
		Assoc		Others				A vs O		A vs M	
		AVG	SD	AVG	SD	AVG	SD	*t*(26)	*p*	*t*(26)	*p*
All	Prec	0.57	0.29	0.48	0.22	0.60	0.24	0.81	0.42	0.39	0.70
	Rec	0.28	0.15	0.23	0.17	0.37	0.19	0.70	0.49	1.16	0.25
N & NP	Prec	0.50	0.36	0.43	0.30	0.54	0.42	0.55	0.59	0.25	0.80
	Rec	0.20	0.11	0.28	0.26	0.27	0.26	0.83	0.42	0.71	0.49
V & VP	Prec	0.81	0.26	0.32	0.38	0.33	0.33	3.30	0.003[a]	2.96	0.006[a]
	Rec	0.46	0.32	0.15	0.24	0.39	0.33	2.84	0.009[a]	0.51	0.61

[a] $p < 0.01$

organizes them according to viewpoints, with ambiguity detection algorithms support the identification of possible defects. Finally, we proposed also the Interactive Narrator [50] as a default visualization interface for the Visual Narrator's output. That tool is inspired by Shneiderman's information visualization guidelines [28], e.g., overview-first, details-on-demand and filtering; however, REVV-Light adds explicit support for ambiguity identification and a role-centered organization of the extracted terms.

9. Conclusions and future work

This paper presented an extensive account on REVV-Light, an open source Web 2.0 tool that combines information visualization and natural language processing in order to help requirements analysts pinpoint terminological ambiguity that stems from the occurrence of near-synonyms in user story requirements.

In addition to describing the concept of the tool-based approach, we reported on an experiment in which 57 students organized into 28 groups assessed the precision and recall of REVV-Light versus a manual inspection based on pen and paper. The results reject the hypotheses that the current version of REVV-Light outperforms the manual inspection in terms of precision and recall.

More generally, the results show how difficult it is with either approach to obtain high precision and recall. As discussed in Section 7, *i.* low recall can be ascribed to the size of the search space, which is quadratic with the number of terms that occur in the user stories; and *ii.* low precision is probably due to the difficulty in establishing if two terms are near-synonyms, and whether their near-synonymy may lead to different interpretations of the requirements.

The qualitative observations gathered during the experiments provided rich insights that enable a better interpretation of the quantitative results. Our findings confirm that tagging ambiguities is a time-consuming activity that can be justified only by an adequate return on investment; the latter depends on the impact of the ambiguities in the following software development phases, which is hard to predict. Moreover, we could assess how experience in ambiguity tagging is a determinant factor in obtaining high precision and recall. This was visible in the experiment: the groups leveraged their prior experience with manual inspection and were able to use more efficiently the thirty minutes at hand.

The different pros and cons of the two tested approaches lead us to the *hypothesis* that a synergy between both approaches may be beneficial, by combining the ability to navigate through the context of ambiguity (the user stories themselves) with the overview that a visualization technique can provide, e.g., that inspired by Venn diagrams.

A major research direction concerns the design, development, and experimentation of such concept that combines REVV-Light and manual inspection. To obtain better results concerning the suggested ambiguities, we shall consider going beyond domain-independent corpora and using domain-specific information, in line with existing proposals from the literature [42]. We hypothesize, thus, that the use of domain knowledge—either embedded an automated tool or possessed by manual taggers—may lead to significantly higher precision.

The visualization technique needs to be improved to avoid the tunnel vision that was mentioned by the experiment participants in Section 6.2.3. Although we made it explicit to the participants that REVV-Light suggests *potential* ambiguities, the main effects were that *i.* some participants were induced to accept those suggestions as genuine ambiguities, and *ii.* other participants did not consider any terms that were not suggested by the tool.

The experimentation made it obvious that even proof-of-concept tools require a sufficient level of maturity; bugs reduce the potential of the tool and create negative sentiment in the users. Bugs and low usability of the tool led to frustration situations, which are likely to have hindered the performance of the participants. This issue is likely to affect practitioners in real projects too.

Future studies should extend the notion of nocuous ambiguity [39] toward those cases of ambiguity that are likely to have an impact on the following stages of the development process. While the research community will inevitably deliver new techniques and tools thanks to the increasingly lower barriers to access advanced NLP tooling, it is essential to obtain evidence that fighting ambiguity is necessary and leads to demonstrable benefits.

Acknowledgement

We are indebted to the 57 students who participated in the experiment. We thank the companies who made available their user stories. We dedicate this paper to our co-author Ivor van der Schalk, who developed the REVV tool; we will remember you by continuing the research that you have enabled.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.infsof.2018.12.007.

References

[1] D.M. Berry, E. Kamsties, M.M. Krieger, From Contract Drafting to Software Specification: Linguistic Sources of Ambiguity, Technical Report, School of Computer Science, University of Waterloo, Canada, 2001.
[2] M. Bano, Addressing the challenges of requirements ambiguity: a review of empirical literature, in: Proc. of the International Workshop on Empirical Requirements Engineering, 2015, pp. 21–24.
[3] B. Rosadini, A. Ferrari, G. Gori, A. Fantechi, S. Gnesi, I. Trotta, S. Bacherini, Using NLP to detect requirements defects: an industrial experience in the railway domain, in: Proc. of the International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ), 2017, pp. 344–360.
[4] F. de Bruijn, H.L. Dekkers, Ambiguity in natural language software requirements: a case study, in: International Working Conference on Requirements Engineering: Foundation for Software Quality, Springer, 2010, pp. 233–247.

[5] E.J. Philippo, W. Heijstek, B. Kruiswijk, M.R. Chaudron, D.M. Berry, Requirement ambiguity not as important as expected results of an empirical evaluation, in: International Working Conference on Requirements Engineering: Foundation for Software Quality, Springer, 2013, pp. 65–79.

[6] H. Femmer, D.M. Fernández, S. Wagner, S. Eder, Rapid quality assurance with requirements smells, J. Syst. Softw. 123 (2017) 190–213.

[7] G. Génova, J.M. Fuentes, J. Llorens, O. Hurtado, V. Moreno, A framework to measure and improve the quality of textual requirements, Requir. Eng. 18 (1) (2013) 25–41.

[8] E. Cambria, B. White, Jumping NLP curves: a review of natural language processing research, IEEE Comput. Intell. Mag. 9 (2) (2014) 48–57.

[9] D. Berry, R. Gacitua, P. Sawyer, S. Tjong, The case for dumb requirements engineering tools, in: Proc. of the International Working Conference on Requirements Engineering: Foundation for Software Quality, in: LNCS, vol. 7195, 2012, pp. 211–217.

[10] G. Lucassen, F. Dalpiaz, J.M.E.M. van der Werf, S. Brinkkemper, Improving agile requirements: the quality user story framework and tool, Requir. Eng. 21 (3) (2016) 383–403.

[11] E. Kamsties, D.M. Berry, B. Paech, Detecting ambiguities in requirements documents using inspections, in: Proceedings of the first workshop on inspection in software engineering (WISE01), Citeseer, 2001, pp. 68–80.

[12] Ö. Albayrak, J.C. Carver, Investigation of individual factors impacting the effectiveness of requirements inspections: a replicated experiment, Empir. Softw. Eng. 19 (1) (2014) 241–266.

[13] G. Lucassen, F. Dalpiaz, J.M.E.M. van der Werf, S. Brinkkemper, The use and effectiveness of user stories in practice, in: Proceedings of the International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ), in: LNCS, vol. 9619, Springer, 2016, pp. 205–222.

[14] M. Cohn, User Stories Applied: For Agile Software Development, Addison Wesley Professional, Redwood City, CA, USA, 2004.

[15] G. Lucassen, M. Robeer, F. Dalpiaz, J.M.E. van der Werf, S. Brinkkemper, Extracting conceptual models from user stories with visual narrator, Requir. Eng. 22 (3) (2017) 339–358.

[16] F. Dalpiaz, I. van der Schalk, G. Lucassen, Pinpointing ambiguity and incompleteness in requirements engineering via information visualization and NLP, in: Proc. of the International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ), 2018.

[17] A. Finkelstein, J. Kramer, B. Nuseibeh, L. Finkelstein, M. Goedicke, Viewpoints: a framework for integrating multiple perspectives in system development, Int. J. Softw. Eng. Knowl. Eng. 2 (1) (1992) 31–57.

[18] G.P. Mullery, CORE - a method for controlled requirement specification, Proc. Int. Conf. Softw.Eng. (1979) 126–135.

[19] I. Sommerville, P. Sawyer, Viewpoints: principles, problems and a practical approach to requirements engineering, Ann. Softw. Eng. 3 (1) (1997) 101–130.

[20] M.L. Shaw, B.R. Gaines, Comparing conceptual structures: consensus, conflict, correspondence and contrast, Knowl. Acquisit. 1 (4) (1989) 341–363.

[21] K. Pohl, Requirements Engineering: Fundamentals, Principles, and Techniques, Springer, 2010.

[22] M. Robeer, G. Lucassen, J.-M. Van der Werf, F. Dalpiaz, S. Brinkkemper, Automated extraction of conceptual models from user stories via NLP, in: Proc. of the International Requirements Engineering Conference, 2016.

[23] C. DiMarco, G. Hirst, M. Stede, The semantic and stylistic differentiation of synonyms and near-synonyms, in: Proc. of the AAAI Spring Symposium, 1993, pp. 114–121.

[24] L.J. Rips, E.J. Shoben, E.E. Smith, Semantic distance and the verification of semantic relations, J. Verbal Learn. Verbal Behav. 12 (1) (1973) 1–20, doi:10.1016/S0022-5371(73)80056-8.

[25] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Proc. of the Neural Information Processing Systems Conference, 2013, pp. 3111–3119.

[26] F. De Sousa Webber, Semantic folding theory and its application in semantic fingerprinting, arXiv:https://arxiv.org/abs/1511.08855 (2015).

[27] G. Lucassen, M. Robeer, F. Dalpiaz, J.M.E.M. van der Werf, S. Brinkkemper, Extracting conceptual models from user stories with visual narrator, Requir. Eng. 22 (3) (2017) 339–358, doi:10.1007/s00766-017-0270-1.

[28] B. Shneiderman, The eyes have it: a task by data type taxonomy for information visualizations, in: Proc. of the IEEE Symposium on Visual Languages (VL), 1996, pp. 336–343.

[29] L. Micallef, Visualizing Set Relations and Cardinalities Using Venn and Euler Diagrams, Ph.D. thesis, University of Kent, 2013.

[30] J.J. Carroll, Drawing Venn Triangles, HP Labs Technical Report 73, 2000.

[31] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell, A. Wesslén, Experimentation in Software Engineering, Springer Science & Business Media, 2012.

[32] J.W. Perry, A. Kent, M.M. Berry, Machine literature searching X. Machine language; factors underlying its design and development, Am. Doc. 6 (4) (1955) 242–254.

[33] F. Dalpiaz, Requirements Data Sets (User Stories), 2018. Mendeley Data, v1, http://dx.doi.org/10.17632/7zbk8zsd8y.1.

[34] F. Dalpiaz, F.B. Aydemir, Tagged Near-Synonyms in Requirements Specifications, 2018. Mendeley Data, v1, http://dx.doi.org/10.17632/yjnp5chzbv.1.

[35] D.M. Berry, Evaluation of tools for hairy requirements and software engineering tasks, in: Proc. of the IEEE International Requirements Engineering Conference Workshops (REW), 2017, pp. 284–291.

[36] H. Femmer, Requirements quality defect detection with the qualicen requirements scout, in: Proc. of the 1st Workshop on Natural Language Processing for Requirements Engineering (NLP4RE), 2018.

[37] H. Levene, Robust tests for equality of variances, in: Contributions to Probability and Statistics. Essays in Honor of Harold Hotelling, 1961, pp. 279–292.

[38] S.F. Tjong, D.M. Berry, The design of SREE: a prototype potential ambiguity finder for requirements specifications and lessons learned, in: Proc. of the International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ), 7830, 2013, pp. 80–95.

[39] A. Willis, F. Chantree, A. De Roeck, Automatic identification of nocuous ambiguity, Res. Lang. Comput. 6 (3) (2008) 355–374.

[40] H. Yang, A. De Roeck, V. Gervasi, A. Willis, B. Nuseibeh, Analysing anaphoric ambiguity in natural language requirements, Requir. Eng. 16 (3) (2011) 163.

[41] N. Kiyavitskaya, N. Zeni, L. Mich, D.M. Berry, Requirements for tools for ambiguity identification and measurement in natural language requirements specifications, Requir. Eng. 13 (3) (2008) 207–239.

[42] A. Ferrari, G. Lipari, S. Gnesi, G.O. Spagnolo, Pragmatic ambiguity detection in natural language requirements, in: Proc. of the International Workshop on Artificial Intelligence for Requirements Engineering (AIRE), IEEE, 2014, pp. 1–8.

[43] Z.S.H. Abad, G. Ruhe, M. Noaeen, Requirements engineering visualization: a systematic literature review, in: Proc. of the International Requirements Engineering Conference, 2016.

[44] D. Savio, P. Anitha, A. Patil, O. Creighton, Visualizing requirements in distributed system development, in: Proc. of the Workshop on Requirements Engineering for Systems, Services and Systems-of-Systems, IEEE, 2012, pp. 14–19.

[45] S. Reddivari, S. Rad, T. Bhowmik, N. Cain, N. Niu, Visual requirements analytics: a framework and case study, Requir. Eng. 19 (3) (2014) 257–279.

[46] D. Duarte, C. Farinha, M.M. da Silva, A.R. da Silva, Collaborative requirements elicitation with visualization techniques, in: Proc. of the IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), IEEE, 2012, pp. 343–348.

[47] N. Agarwal, R. Karimpour, G. Ruhe, Theme-based product release planning: an analytical approach, in: Proc. of the Hawaii International Conference on System Sciences (HICSS), IEEE, 2014, pp. 4739–4748.

[48] K. Wnuk, T. Gorschek, D. Callele, E.-A. Karlsson, E. Åhlin, B. Regnell, Supporting scope tracking and visualization for very large-scale requirements engineering-utilizing FSC+, decision patterns, and atomic decision visualizations, IEEE Trans. Softw. Eng. 42 (1) (2016) 47–74.

[49] G. Lucassen, F. Dalpiaz, J.M.E. van der Werf, S. Brinkkemper, Visualizing user story requirements at multiple granularity levels via semantic relatedness, in: Proc. of the International Conference on Conceptual Modelling (ER), 2016, pp. 463–478.

[50] G.-J. Slob, F. Dalpiaz, S. Brinkkemper, G. Lucassen, The interactive narrator tool: effective requirements exploration and discussion through visualization, Posters and Demos Track of REFSQ, 2018.