On the Ability of Lightweight Checks to Detect Ambiguity in Requirements Documentation

Martin Wilmink[1] and Christoph Bockisch[2(✉)]

[1] Open Universiteit, Heerlen, The Netherlands
m.wilmink67@kpnmail.nl
[2] Philipps-Universität Marburg, Marburg, Germany
bockisch@mathematik.uni-marburg.de

Abstract. *Context & motivation*: The quality of requirements documentation, which is often written in natural language, directly influences the quality of subsequent software engineering tasks. Ambiguity is one of the main quality risks, but unfortunately natural language has a natural tendency towards ambiguity.

Question/problem: Precisely identifying ambiguity in specifications is virtually impossible fully automatically due the complexity and variability of natural language. Ignoring grammar and context in the analysis, on the other hand, makes an implementation and application feasible, but also reduces the accuracy. The question researched in this paper is whether such a lightweight check can still sufficiently accurately detect which requirements are formulated ambiguously or certainly.

Principal ideas/results: To investigate this research question, we have implemented a lightweight analysis tool based on a finite dictionary combining different results from the literature. The tool, called *tactile check*, adds annotations to phrases in requirements documents, which are weak respectively strong with regard to non-ambiguity. Within an embedded single case study, *tactile check* is applied to two real requirements documents (totaling 293 requirements) from KLM Engineering & Maintenance and the results (454 annotations in total) are assessed by three expert business analysts. In our study, the tool achieved a precision and recall of at least 77% respectively 59%. Annotations of weak phrases have prevalently been perceived as helpful for reducing ambiguity.

Contribution: In this paper, we establish that simple textual analyses with low overhead can detect ambiguity in requirements with significant accuracy. Our experts assessed the analysis' findings as helpful input to reducing the ambiguity. The tool and dictionary used in our study are provided for download to support repeatability of the study. Furthermore, we provide an extended dictionary for download that incorporates suggestions by our experts.

Keywords: Requirements engineering · Business requirements · Natural language · Ambiguity · Software quality · Context-insensitive analysis · *tactile check*

P. Grünbacher and A. Perini (Eds.): REFSQ 2017, LNCS 10153, pp. 327–343, 2017.
DOI: 10.1007/978-3-319-54045-0_23

1 Introduction

The requirements engineering process covers all elements, from business requirements elicitation to detailed baseline build definition. Requirement documentation forms one of the important artifacts of this process. Requirements describe the product services within its given boundaries [15,16]. They are initially specified at a very high level of abstraction and subsequently refined by adding technical details. In this paper, we focus on requirements documents at the first level of this process, which is called *business requirements*.

One key quality attribute of requirements is non-ambiguity, since ambiguity easily leads to misinterpretation and thus failing to satisfy the expectations of the business [2,10,11,13]. In practice, the business requirements are typically written in natural language. However, natural language is inherently ambiguous.

In the context of this study, we mean by *ambiguity* (or *level of ambiguousness*) of requirements whether (or to which degree) a requirement has the potential to be interpreted differently by different readers targeted by the requirements document. Thus, if, e.g., all members of the development team and the customer understand the requirement in the same way, we consider it to be *certain* even if the phrasing potentially also has multiple meanings. Referring to the Ambiguity Handbook [1], the kind of ambiguity considered in our study intersects with the categories *semantic ambiguity*, *pragmatic ambiguity* and *vagueness*.

Several studies [3,4,6,9,18] describe attributes, indicators and metrics for quality characteristics including non-ambiguity. The studies are typically accompanied by a tool to identify these indicators. Often, ambiguity is caused by the unintended usage of words or phrases that induce ambiguity in the text. The tools developed in the aforementioned studies, therefore, use finite dictionaries and complex techniques from natural language processing to determine the quality. However, the validation presented in these papers primarily takes place in an *academic context*. From our experience, one reason could be that existing tools are perceived as too heavy-weight by practitioners.

The first author of this paper is a functional application manager at KLM Engineering & Maintenance for various IT related projects, and is deeply involved in the definition of requirements. The study on which we report in this paper is thus carried out in an industrial context. In our study *we investigate if a simple and practical tool, only based on a finite dictionary, has the potential to improve the ambiguity-awareness* of the analyst writing the document and by doing so improving the overall quality of the specification document. Besides performing the study in an *industrial environment*, another contribution of our work is the combination of the concepts of two previously conducted studies:

- The NASA ARM tool [18] and its reconstruction [3] check for ambiguity in the form of *weak and strong phrases using a finite dictionary* approach. Additionally it checks the document structure, including cross references.
- The tool SMELL [6] detects various subjective and non-verifiable terms as ambiguity forms *without the objective of being 100% correct*. Advanced text analytics is used to recognize inflections of predefined words.

We adopt the approach of using a finite dictionary, ignoring context and grammar from the ARM tool and the incentive that 100% accuracy is unnecessary from the SMELL tool, into a lightweight analysis tool, which we call *tactile check*. We take over the dictionary of ARM as well as most inflections of words recognized by SMELL, but we omit context analyses like structural checks or text analytics. *Tactile check* is implemented as a macro for Microsoft Word and adds annotations to phrases in requirements documents, which are weak respectively strong with regard to non-ambiguity. It is very lightweight as it ignores context and grammar, which may reduce the precision of annotations and impact the usefulness in practice. We therefore investigate in our study the research question: *How do business analysts perceive the effectiveness of the* tactile check *in accurately detecting which requirements are formulated ambiguously or certainly?*

We apply *tactile check* in an embedded single case study to two actual requirements documents from KLM E & M with 199 respectively 94 requirements. A total of 454 phrase annotations are inspected by three expert requirements analysts from KLM E & M. For one of the documents, the experts affirmed a precision of at least 96% and a recall of at least 89%. For the other, they affirmed a precision of 77% and a recall of at least 59%. The weak annotations were predominantly perceived as helpful for reducing ambiguity, while annotations of strong phrases were considered not helpful. This paper is based on the first author's master thesis [17], where additional information can be found. We summarize the contributions of our work as follows:

- We establish that simple textual analyses with low overhead can accurately detect ambiguity in requirements.
- To make our study repeatable, we make the *tactile check* tool as well as the collected data available for download.[1]

2 Research Design

We split our general question into three sub-questions:

RQ-Weak. To what extent does the annotation of *weak* phrases with *tactile check* accurately detect ambiguous requirements?

RQ-Strong. To what extent does the annotation of *strong* phrases with *tactile check* accurately detect certain requirements?

RQ-Helpful. To what extent are the presented *tactile check* annotations perceived as helpful by business analysts to reduce the overall ambiguousness?

To answer these questions, we follow an approach [14] where the proposed *tactile check* tool is evaluated with the use of existing requirement documents as input data. The annotated phrases in the requirements document are the starting point for the assessment by three expert requirements analysts. This

[1] See the Tactile Check homepage: https://github.com/mwmk67/TactileCheck.

relates to an *interpretivism* research philosophy where data samples are limited but analyzed with in-depth knowledge. The expert assessment is based on a set of categories to be assigned to each annotated phrase. This follows an *abduction* research method where a new model is formed based on the collected data, which is a natural fit with the interpretivism philosophy.

In line with the research method and philosophy an *embedded single case study* is performed, where the case subject will be KLM Engineering & Maintenance and two different Business IT related projects as subject to analysis. From both projects the business requirements chapters are included for usage in this research. There is only one limitation on the document usage: No financial information may be disclosed.

- e-EGS (field loadable software solution to support Boeing 787). The requirements are written in natural language using the business stakeholder's vocabulary. The document revision state is "Approved".
- CMS-plus (logistics solution for aircraft maintenance execution and administration). The document is written in partially structured natural language as use cases/user stories. The document revision state is "Approved".

2.1 The *Tactile Check* Tool

First of all, we have developed a tool, called *tactile check*, to perform the lightweight, dictionary-based annotation of phrases in the requirements documents. This is to make the annotation process reliable and repeatable. To allow other researchers to re-assess our method with other requirements documents and other analysts, we make the tool and the dictionary available for download.[2]

Tactile check essentially combines the approaches of the NASA ARM tool [18] and the SMELL tool [6]. Both tools use dictionaries to identify weak and strong phrases, but they also both perform additional complex analyses, e.g., of the document structure or using text analytics. We limit our implementation to the dictionary-based analysis combining the dictionaries of ARM and SMELL. Instead of performing text analytics, we have extended the dictionary with inflections of its words.

Since requirements documents are written in MS Word at KLM E & M, we have developed the *tactile check* tool as a Visual Basic for Applications (VBA) macro. The dictionary is placed in a separate file to enable amendments to the dictionary without changing the VBA code. Annotations are represented by changing the phrase's font presentation, italic for weak, bold for strong phrases. The name of the quality indicator as defined by the loaded dictionary and a unique identification (sequence) number of the finding are added in the form of a comment. An example annotation of a weak as well as a strong phrase annotation is shown in Fig. 1.

Our tool it is provided as a Word macro and can thus be executed in the same environment as is used to write the requirements document. The results are

[2] See the Tactile Check homepage: https://github.com/mwmk67/TactileCheck.

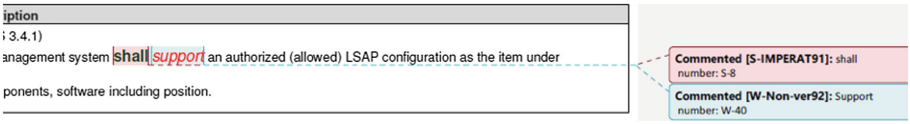

Fig. 1. Example of the annotation output.

also immediately displayed in the Word document. Annotations are provided as regular comment, a presentation format well known to requirements engineers. We therefore envision that requirements engineers can apply our tool very frequently, e.g., right after a new requirement has been specified or at the end of each day.

2.2 Data Collection

A detailed description of the data and data collection process for each research question can be found in annex 2 to the thesis of the first author [17]. In the following we give an overview of this process.

Accuracy of Weak and Strong Phrase Annotation. In research questions RQ-Weak and RQ-Strong, we address the reliability of the *tactile check* annotation of weak and strong phrases. To investigate these questions, the experts assess all[3] phrases in the requirements documents and their annotations by taking the context of the phrase into account.

Following the binary classification diagram shown in Fig. 2, they determine a label for each annotated phrase. These labels (cf. Table 1) indicate to which degree the expert agrees that the identified phrase is formulated ambiguously (for weak phrase annotations) or non-ambiguously (for strong phrase annotations). The table is based on Femmer et al. [6], extended with labels for strong phrases. For each label we specify the type of result (true or false positive) and whether a such annotated phrase influences ambiguity (A-Y) or not (A-N).

For requirements that are not annotated with weak or strong phrases, we determine together with the experts, whether the requirement is ambiguous or certain, and, thus, should have contained an annotated phrase. These "missed phrases" form the false negatives. Since our expert analysts only have limited time available for this project, we can only investigate samples of the requirements to identify missed phrases.

Perceived Helpfulness. In research question RQ-Helpful, we address whether the annotations actually have the potential to improve the non-ambiguity of requirements. To determine the perceived helpfulness, the experts perform a

[3] The strong phrases "and" and "should" occur extremely frequently in similar sentence patterns. To save the limited time of our expert analysts, we asked them to only assess 15 occurrences of these two phrases. All assessments are almost identical.

Table 1. Weak (based on [6]) and strong labels and associated attributes. (true: true positive, false: false positive, A-Y: influencing ambiguity, A-N: not influencing ambiguity)

Label code		Description	Type	Infl. ambig.	
Weak	Strong			Weak	Strong
W-1	–	This finding revealed a potential problem	true	A-Y	–
W-2	–	This requirement needs a review	true	A-Y	–
W-3	–	There is some explicit knowledge, which should be written down	true	A-Y	–
W-4	–	There should be a reference at this point	true	A-Y	–
W-5	–	This is a major issue that must be addressed	true	A-Y	–
W-6	S-1	While this is not an issue here, it must be further explained and refined at a different point	true	A-N	A-Y
W-7	S-2	This could be problematic, but this part of the specification is not so important	true	A-N	A-N
W-8	S-3	This finding seems problematic, but is clear to a domain expert	true	A-N	A-Y
W-9	S-4	This is not a problem here	true	A-N	A-N
W-10	S-5	The *tactile check* did not work correct	false	A-N	A-N

critical review of each assessed phrase within its context and determine whether the phrase annotation is helpful to "trigger" the expert to further clarify or enhance the requirement.

Interviews. When experts complete the analysis of both requirement data sets, they are interviewed to elaborate on their experience using the *tactile check*. This interview is semi-structured and includes the following questions:

- What is the view on the chosen approach?
- What is the view on the usefulness of weak phrase annotations?
- What is the view on the usefulness of strong phrase annotations?
- Would the *tactile check* be useful as additional method to assist a business requirements author to reduce the overall ambiguity of business requirements?

2.3 Reliability and Internal Validity Aspects

To ensure that the research outcome is valid it is important to identify possible threats to the validity. Reliability relates to the ability to repeat the

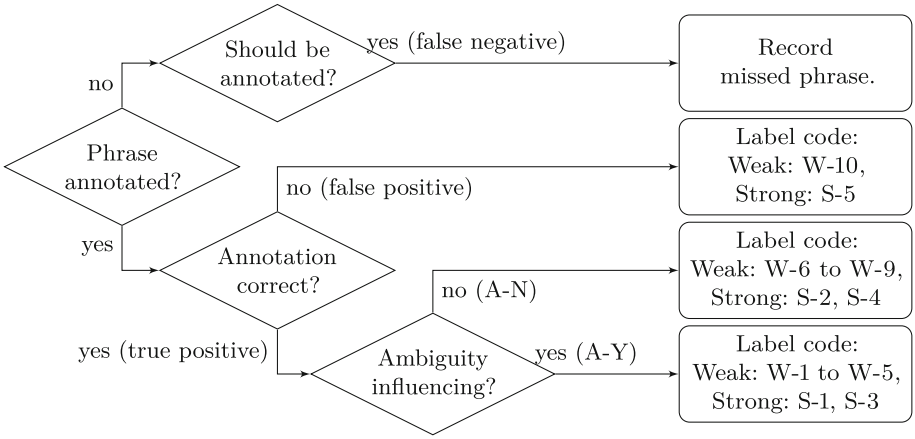

Fig. 2. Phrase classification diagram.

measurements and yield the same results. Saunders et al. [14] identify the threats to reliability and validity, which we discuss below.

Participant Error. Time pressure, distraction and knowledge influence our judgment and can lead to false assessment. To mitigate the influence of time pressure and distraction the experts are requested perform the classification and assessment in a time slot with a minimum of 2 h. Preferably the assessment is carried out at a distraction free office location.

The "knowledge" element is elusive to quantification. As minimum requirement, the expert must work at least 5 years with KLM E & M and a minimum of 3 years as business analyst. Nevertheless, this will not eliminate the risk of different knowledge levels between the individual domain specialists. Therefore it is to be expected that variations in classification will occur. To counter this effect three domain experts will perform the assessment. In case that a participant provides an extreme different interpretation, an additional review and argumentation can possibly clarify the differences. Additionally, we use the Fleiss' Kappa measure to determine inter-rater reliability.

Participant Bias. The result of this research has no direct impact on the daily activities of the domain specialist. And, as assessment will be performed on an individual basis, no deliberation between the participants is expected. There is no foreseen incentive to develop *tactile check* as a supported method in their daily work routines. Therefore, it is not expected that the analysts will consciously steer the interpretation and classification to a perceived favorable outcome.

Each data set (e-EGS and CMS-plus) is analyzed by one expert who participated in the analyzed project. This holds the risk of bias, as one expert is validating work in which he was previously involved.

Researcher Error. Due to the nature of the first author's curriculum (part-time student), there is a risk that the time frame in which the research is performed becomes fragmented. To mitigate this risk, a research project plan including detailed and realistic time schedules is used to measure the progress and identify at an early stage deviations from the planning.

Researcher Bias. In this research, the researcher is part of the organization where the research is conducted (internal researcher, cf. [14]). While the advantages are for example easy access to data and resources, the disadvantage is familiarity with the organization. In this research setting it is not anticipated as influential, as the researcher is *not* part of the business analyst team that will perform the assessments. During the assessment it is envisioned that only minimal assistance from the researcher is required hence limiting the risk of influencing the assessor. When evaluating the acquired data using the *nominal* measuring level limits the complexity of the applicable calculations and risk of interpretation bias.

Construct Validity. For this research setup the following elements that influence the construct validity are identified:

Consistency of Input Data. To create a consistent annotation of phrases from the finite dictionary an automated tool is developed. The automated approach ensures that annotation of the weak and strong phrases as defined in the dictionary is consistent and repeatable.

Measurement Scale. The measurement level at which the data is classified is nominal. This suits the objective of classifying the different findings and counting totals for weak and strong categories. Measurement at the nominal level accommodates basic counting of elements.

Consistency of Data Collection. Each domain specialist performs the assessment based on his own level of experience and proficiency, i. e., while the results can be arithmetically correct, there is room for variance in the outcome. All data of each assessment is used to compare and evaluate the influence of this variance.

Triangulation. To be able to value the findings of the assessment and classification a semi-structured interview is conducted with each participating analyst. The results of the semi-structured interviews are to be compared with the results of assessment analyses.

External Validity. The used single case study research strategy limits the ability to generalize the outcome of this result. The result may be specific to the domain and the context of the data used. Based on the small and domain-specific data set and limited group of domain specialists who evaluate the annotated data the external validity is uncertain.

3 Data Analysis

Each data item contains the unique phrase identifier, an identifier of the analyst whose assessment is recorded, the classification label assigned by the analyst and whether the analyst perceives the annotation as helpful. The classification label is further split up into its characteristics, namely the type (true/false positive/negative) and whether it influences ambiguity (A-Y or A-N).

Table 2 shows the generic breakdown of the analyzed requirements and phrases. It can be seen that volume of the requirements and annotated phrases in the e-EGS data set is considerable bigger than in the CMS-plus data set.

Table 2. Overall count of requirements and phrases.

Description	e-EGS	CMS-plus
$\sum$ Requirements in document	199	94
$\sum$ Requirements annotated	188	40
$\sum$ Requirements with weak phrases annotated	55	10
$\sum$ Requirements with strong phrases annotated	187	37
$\sum$ Phrases annotated	367	87
$\sum$ weak phrases annotated	67	20
$\sum$ strong phrases annotated	300	67

For e-EGS, the sum of requirements with weak phrases and requirements with strong phrases is larger than the total number of requirements. The reason is that requirements can contain phrases with weak annotations and phrases with strong annotations at the same time. The data shows that the number of weak annotations is relatively low and much smaller than the number of strong annotations. This is not surprising, considering that both requirements documents already have finalized status.

Generally, a requirement contains multiple phrases and we have collected and analyzed the data per requirement as well as per phrase. Both approaches have yielded almost identical results with at most 2% variation. Therefore, we only discuss the results at the granularity of requirements in the following. The full data sets can be found in [17].

To discuss the research questions RQ-Weak and RQ-Strong, we analyze the correctness of the annotations as seen by the expert analysts. The questions revolve around the accuracy of the weak and strong annotations. Important components of accuracy are the *precision* and *recall*, this is the percentage of correctly annotated requirements and the percentage of missing requirements annotations, respectively. Both measures can be combined, equally weighted, using the *balanced F-score* (F_1*-score*). To calculate these measures, we determine the values of *true* and *false positives* as well as *false negatives* (cf. Sect. 2.2). The formulas of for precision, recall and the F_1-score are given below. Other measures

such as *miss rate* or *specificity* can also be calculated from the data presented in this paper. This can answer additional questions such as the likelihood of missing ambiguous requirements, which are however not the objective of the study presented here.

$$precision = \frac{true\ positives}{true\ positives + false\ positives} \tag{1}$$

$$recall = \frac{true\ positives}{true\ positives + false\ negatives} \tag{2}$$

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{3}$$

The data analysis presented in this section is based on the classification of items (i.e., the annotations provided by *tactile check*) by three different experts to increase the reliability of the classifications. To assess this reliability, we use Fleiss' Kappa measure [7] to determine the agreement between our raters. Generally, a positive κ value means that there is agreement between raters beyond what would be expected by chance; a value of 1 means complete agreement. We are limited to this instrument for assessing the inter-rater agreement, since we use a *nominal scale* for the classification.

Some annotations have not been rated by all experts due to time constraints. For the inter-rater reliability test we only considered those annotations rated by all three experts (this are 20 annotations for the CMS-plus data set and 43 for e-EGS). In both cases the confidence level was set to 95%, and in both cases we have a very low p-value ($9.687 \cdot 10^{-13}$, respectively $3.379 \cdot 10^{-5}$); this means that the statistical significance of our results is very high. Finally, the κ values of 0.586 (CMS-plus) and 0.202 (e-EGS) show that there is agreement between our experts, in the case of CMS-plus even largely so.

In the remainder of this section, we first discuss the data analysis from the perspective of our three research questions and finally combine the results obtained for the accuracy-related research questions (RQ-Weak and RQ-String) with the results of the perceived helpfulness (RQ-Helpful).

3.1 Accuracy of Weak Phrase Annotations

Our first research question, RQ-Weak, was: *To what extent does the annotation of* weak *phrases accurately detect ambiguous requirements?* To answer this question, we first need to establish whether the precision and recall values indicate that our method is usable with respect to weak phrase annotations. For each document and for each analyst Table 3 shows the number of annotations that were identified as true and false positives, and the number of false negatives found by the analyst for weak phrase annotations.

For the evaluated weak phrases in the e-EGS data set, the values for precision and recall are close to 90% or above. This indicates that most results are considered relevant and that most relevant results are shown. The F_1-score of 92% and above confirms a good accuracy for detecting weak phrases.

Table 3. Collected data from e-EGS and CMS-plus with regard to *weak* phrases.

	Analyst #1		Analyst #2		Analyst #3	
	e-EGS	CMS-plus	e-EGS	CMS-plus	e-EGS	CMS-plus
$\sum$ true positives	55	10	55	10	38	10
$\sum$ false positives	2	3	1	3	1	3
$\sum$ false negatives	7	6	7	7	4	0
Precision	96%	77%	98%	77%	97%	77%
Recall	89%	63%	89%	59%	90%	100%
F_1-score	92%	69%	93%	67%	94%	87%

For the CMS-plus data set, the values are considerably lower than those gathered from the e-EGS data set. But at least the value for precision is still relatively high with 77%, meaning that only one out of four annotations is wrong. For analysts #1 and #2, the recall drops to 59%, indicating that according to them almost half the weak phrases are left out. Analyst #3 did not identify any false negatives, leading to a recall of 100%. This result should be considered an outlier. The accuracy calculated by the F_1-score is therefore between 67% and 69%, which can still be considered good.

The low recall for CMS-plus is partially caused by phrases deemed weak by the assessors that are not explicitly listed in the dictionary and therefore not annotated in the text. Investigation shows that synonyms of these phrases are in the dictionary.[4] Analyst #3, however, classifies these additional phrases as W-10 ("not a problem") and thus not as "missed", hence his 100% score.

When discussing the lower recall values for the CMS-plus data set, the analysts indicate that despite the lower score, the values are sufficient to use the *tactile check* to annotate weak phrases. The overall experience by the experts is that annotating the weak phrases is consistent and precise enough to be valuable.

3.2 Accuracy of Strong Phrase Annotations

The second research question we want to investigate is RQ-Strong: *To what extent does the annotation of* strong *phrases accurately detect certain requirements?* We investigate this analogously to Sect. 3.1. Table 4 shows the number of annotations that were identified as true and false positives, and the number of false negatives.

For the evaluated strong phrases in both the e-EGS and CMS-plus data sets, the values for precision and recall are close to 90% or above. score of 88% and above shows that the accuracy with regard to detecting strong phrases is very good.

[4] An extended dictionary containing the additional phrases can be found at https://github.com/mwmk67/TactileCheck.

Table 4. Collected data from e-EGS and CMS-plus with regard to *strong* phrases.

	Analyst #1		Analyst #2		Analyst #3	
	e-EGS	CMS-plus	e-EGS	CMS-plus	e-EGS	CMS-plus
$\sum$ true positives	59	19	59	16	25	22
$\sum$ false positives	4	0	4	0	3	0
$\sum$ false negatives	2	4	5	0	4	1
Precision	94%	100%	94%	100%	89%	100%
Recall	97%	83%	92%	100%	86%	96%
F_1-score	95%	90%	93%	100%	88%	98%

3.3 Helpfulness of Annotations

The last research question is RQ-Helpful: *To what extent are the presented* tactile check *annotations perceived as helpful by business analysts in order to reduce the overall ambiguousness?*

For each label, the experts are asked to rate whether they consider it helpful or not with regard to reducing the ambiguousness level of the requirements document. Figures 3 and 4 show the distribution of the answers of each expert per requirements document.

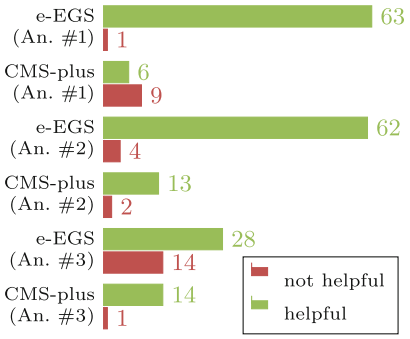

Fig. 3. Weak phrases helpful or not.

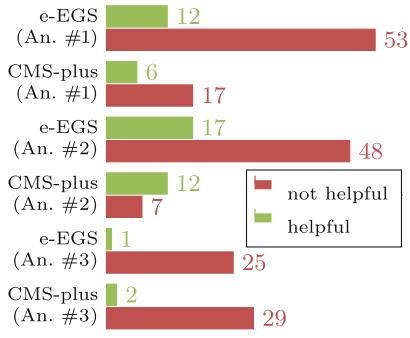

Fig. 4. Strong phrases helpful or not.

For the weak phrases, only analyst #1 ranks the annotations predominantly not helpful for the CMS-plus document. He explains that although the phrase "all" (which occurs multiple times) is indicated as weak, in his opinion it does not influence the overall ambiguity of the given requirements. In all other cases, the weak annotations are mostly rated helpful.

The strong annotations are predominantly rated not helpful more often than helpful with only one exception: The annotations in the CMS-plus document as rated by analyst #2. When asked to elaborate on this exceptional ranking, this

is related to the combination of the used strong phrases in combination with the adjacent weak phrases and that the requirement could be further improved by also rephrasing the strong phrase.

3.4 Effectiveness of a Lightweight Tactile Check

To answer the main research question, the results from our research sub-question are combined and further analyzed. The classification label assigned to each annotated requirement by the experts encodes whether the annotation really indicates an impact on ambiguousness (A-Y) or not (A-N). Furthermore, the experts specify for each annotation whether they perceive it as helpful (Y) or not (N) to improve the requirement. The ratings for the requirements can be plotted in a four-quadrant matrix as shown in Fig. 5.

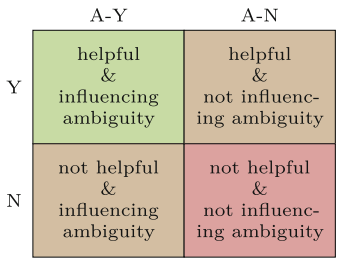

Fig. 5. Four-quadrant evaluation matrix.

For the different data sets we determine the number of annotations, which fall in the different categories of each quadrant. Figure 6 shows this matrix for each data set whereby the number of weak annotations falling in the different categories are written in the respective quadrant. The quadrant with the largest count is highlighted.

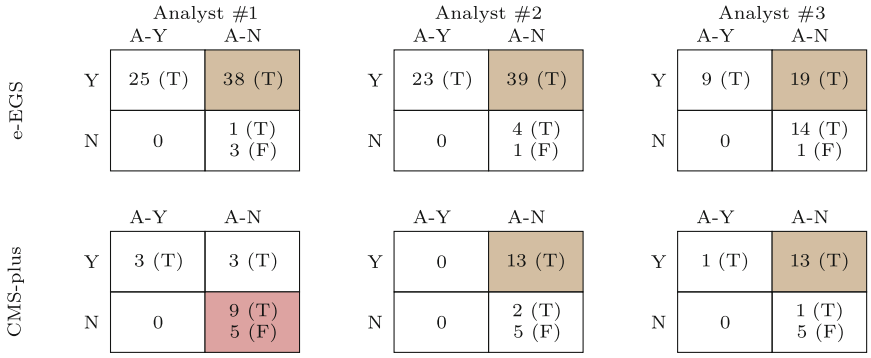

Fig. 6. Four quadrant evaluation for *weak* phrases. (T: true positive, F: false positive)

The matrices show that annotations are predominantly ranked as helpful, although not (seriously) influencing the ambiguity. Nevertheless, for the e-EGS case, still a significant number of weak annotations are regarded as helpful *and* influencing ambiguity. Only analyst #1 classifies most weak phrases of CMS-plus as not helpful.

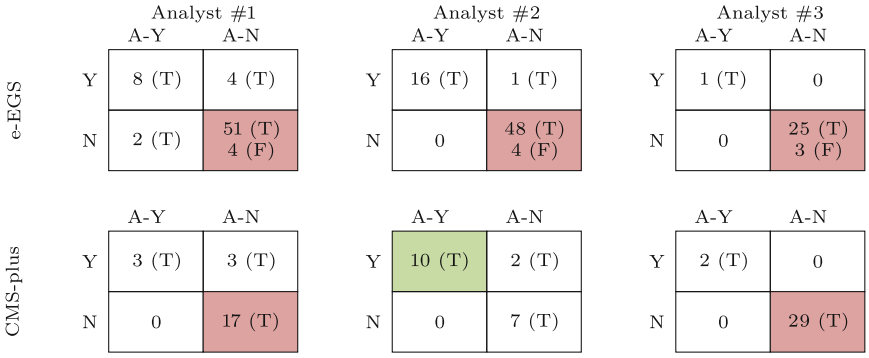

Fig. 7. Four quadrant evaluation for *strong* phrases. (T: true positive, F: false positive)

Figure 7 shows the same analysis for strong annotations. The strong phrases are predominantly ranked as not helpful and not influencing ambiguousness—in the e-EGS case even unanimously.

3.5 Discussion

To further elaborate on the perceived effectiveness of the lightweight ambiguity analysis of requirements, the analysts are asked for their expert opinion on several questions in a semi-structured interview. All three analysts are skeptical about the usefulness of annotating strong phrases, i. e., of marking requirements that are already perceived as good. In general, all experts express that the chosen approach is useful and applicable in practice, although it should be limited to annotating weak phrases.

Already during the assessments of the weak phrases, the analysts were indicating that the annotation made them "rethink" the formulation of single requirements. Even if a requirement has not been seen as severely ambiguous, options to clarify and simplify the requirement often become apparent. The annotations give an additional opportunity to reflect on the written requirements and can even provide the incentive to discuss this further with the stakeholders.

An important remark made by two analysts is that using the proposed *tactile check* would be more beneficial during the initial phase of compilation of the requirements. A critical note is that a consequence of applying the approach could be that requirements engineers (involuntarily) adapt to avoiding the usage of known weak phrases defined in the finite dictionary. Another consequence may be that requirements engineers are lead to not reviewing requirements without weak annotations, although these may still be ambiguous.

4 Related Work

There are several studies concerned with automatic quality assessment for requirements documents, of which we only present a few here. Gleich et al. [8]

detection ambiguity based on Part-Of-Speech tagging (POS), which is a technique from computational linguistics to identify for each word in a sentence, which syntactic role it plays. Patterns are defined that match sentences of tagged words to recognize lexical, syntactic, semantic, pragmatic or vagueness ambiguity. When a pattern matches, this also gives a short explanation in how far the sentence may be ambiguous.

The tool SMELLS introduced by Femmer et al. [5,6] further refines and employs techniques from Natural Language Processing (NLP): POS tagging, morphological analysis, finite dictionaries and lemmatization to identify possible ambiguous language usage. The tool is validated in an academic and industrial setting. Like in our study the specialists indicate that lightweight feedback early in the requirements specification cycle is seen as very beneficial.

These works and others [3,4,6,9,18] commonly describe some quality indicators together with analyses (sometimes supported by a tool) to determine the quality of requirements. In contrast to the approach evaluated in this study, these works apply relatively heavy-weight analyses using natural language processing or structural analyses of the whole document. In our study, we showed in an industrial context that a significantly simpler approach, i.e., purely dictionary-based, can also be employed to good effect.

5 Conclusions and Future Work

In this study, we analyzed whether a lightweight *tactile check* to detect weak and strong phrases with respect to non-ambiguity can effectively improve the quality of requirements documents. To ensure a consistent annotation of such phrases, we have developed a tool performing the checks as macro for MS Word together with a finite dictionary. To make our study repeatable, we provide both the tool and the dictionary, as well as an extended dictionary for download.[5] Within the study, we have applied this check to two actual requirements documents from KLM Engineering & Maintenance. Analyzing a total of 293 requirements with the *tactile check* resulted in 454 annotated phrases, which were assessed by three business analysts from KLM E & M. The analysts generally perceived the approach as effective in practice. The gathered data shows, in line with the qualitative assessment of the experts, that annotations for ambiguous requirements could be identified with a high precision and recall of 92% respectively 87% on average. For the annotated weak phrases, 58% were valued as helpful and 28% as helpful and positively influencing the overall ambiguity level.

The analysts confirmed that annotating the *weak* phrases is beneficial in reducing the ambiguousness level in the written business requirements. During the interviews, the analysts indicated that using the *tactile check* method to identify the weak phrases early during the initial phase of the requirements specification would be most beneficial. This would provide an additional incentive to discuss the annotated requirements with the stakeholders to clarify the requirements.

[5] See https://github.com/mwmk67/TactileCheck.

For the *strong* phrases, the analyzed data show that the *tactile check* method is not perceived as beneficial in reducing ambiguity influences of requirements written in natural language. The reason is that the attention is drawn to requirements, which are already identified as non-ambiguous. Thus, it is expected that no action needs to be taken on these requirements and therefore the annotation does not lead to a reduction in the ambiguousness level. Also the quantitative analysis has shown that there is no perceived benefit to annotating strong phrases. A better alternative could be to annotate requirements that lack strong phrases.

The difference in build-up of the two requirements documents used in this case study was visible in the phrases annotated and in the assessment. The e-EGS requirements are stated in plain natural language, and showed more variety in the annotations than the CMS-plus requirements that are written using a use case/user story structure. This difference in writing styles was not taken into account when the documents were selected. The CMS-plus document did not show a high number of strong phrases that are part of the strong dictionary and the annotated weak phrases showed little variation. We believe that this shows that the *tactile check* method is less effective on semi-structured documents.

During the interviews the experts also mentioned potential risks of the approach. A consequence of applying the approach could be that requirements engineers (involuntarily) adapt to avoiding the usage of known weak phrases defined in the finite dictionary. Another consequence may be that requirements engineers are lead to not reviewing requirements without weak annotations, although these may still be ambiguous.

As future work, we would like to adapt our approach according to the conclusions above and to repeat the study. In particular, it should be assessed whether the identified risks actually materialize. And the hypothesis should be investigated that the lightweight *tactile check* approach is even more advantageous for documents in an early stage of the requirements engineering process.

The ISO standard 29148 [12] includes natural language criteria for requirements specifications including weak and strong phrases. We did not consider the standard in our study, but will collate the phrases from there and our finite dictionary.

References

1. Berry, D.M., Kamsties, E., Krieger, M.M.: From contract drafting to software specification: linguistic sources of ambiguity. In: A Handbook (2003). http://cs.uwaterloo.ca/~dberry/handbook/ambiguityHandbook.pdf. Accessed 31 Dec 2016
2. Boehm, B., Basili, V.R.: Software Defect Reduction Top 10 List. Computer **34**(1), 135–137 (2001)
3. Carlson, N., Laplante, P.: The NASA automated requirements measurement tool: a reconstruction. Innov. Syst. Softw. Eng. **10**(2), 77–91 (2014). http://dx.doi.org/10.1007/s11334-013-0225-8

4. Davis, A., Overmyer, S., Jordan, K., Caruso, J., Dandashi, F., Dinh, A., Kincaid, G., Ledeboer, G., Reynolds, P., Sitaram, P., Ta, A., Theofanos, M.: Identifying and measuring quality in a software requirements specification. In: Proceedings First International Software Metrics Symposium, pp. 141–152. IEEE (1993)
5. Femmer, H., Fernández, D.M., Wagner, S., Eder, S.: Rapid quality assurance with requirements smells. J. Syst. Softw. **123**, 190–213 (2017)
6. Femmer, H., Fernández, D.M., Juergens, E., Klose, M., Zimmer, I., Zimmer, J.: Rapid requirements checks with requirements smells: two case studies. In: Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering, pp. 10–19 (2014). http://doi.acm.org/10.1145/2593812.2593817
7. Fleiss, J., et al.: Measuring nominal scale agreement among many raters. Psychol. Bull. **76**(5), 378–382 (1971)
8. Gleich, B., Creighton, O., Kof, L.: Ambiguity detection: towards a tool explaining ambiguity sources. In: Wieringa, R., Persson, A. (eds.) REFSQ 2010. LNCS, vol. 6182, pp. 218–232. Springer, Heidelberg (2010). doi:10.1007/978-3-642-14192-8_20
9. Gnesi, S., Lami, G., Trentanni, G., Fabbrini, F., Fusani, M.: An automatic tool for the analysis of natural language requirements. Int. J. Comput. Syst. Sci. Eng. **20**(1), 53–62 (2005)
10. Hairul, M., Nasir, N., Sahibuddin, S.: Critical success factors for software projects : a comparative study. Sci. Res. essays **6**, 2174–2186 (2011)
11. Hofmann, H.F., Lehner, F.: Requirements engineering as a success factor in software projects. IEEE Softw. **18**(4), 58–66 (2001)
12. IEEE: ISO/IEC/IEEE 29148: 2011 Systems and software engineering - Life cycle processes - Requirements engineering, pp. 1–94. ISO (2011)
13. Kamata, M.I., Tamai, T.: How does requirements quality relate to project success or failure?. In: 15th IEEE International Requirements Engineering Conference (RE 2007), pp. 69–78. IEEE, October 2007
14. Saunders, M., Lewis, P., Thornhill, A.: Research Methodes for Business Students, 6th edn. Pearson Benelux, London (2012)
15. Sommerville, I.: Software Engineering, 10th edn. Pearson Education Limmited, Harlow (2016)
16. Wiegers, K., Beatty, J.: Software Requirements, 3rd edn. Microsoft Corporation, Redmont (2013)
17. Wilmink, M.: Requirements ambiguousness pitfalls. Master's thesis, Open Universiteit Nederland (2016)
18. Wilson, W.M., Rosenberg, L.H., Hyatt, L.E.: Automated analysis of requirement specifications. In: Proceedings of ICSE, pp. 161–171. ACM Press, New York, USA, May 1997