DETECTING VAGUE WORDS AND PHRASES IN REQUIREMENTS DOCUMENTS IN A MULTILINGUAL ENVIRONMENT

A Thesis

Submitted to the Graduate School
of the University of Notre Dame
in Partial Fulfillment of the Requirements
for the Degree of

Master of Science
in
Computer Science and Engineering

by

Breno Dantas Cruz

----------------------------------------
Collin McMillan, Director

Graduate Program in Computer Science and Engineering
Notre Dame, Indiana
June 2016
DETECTING VAGUE WORDS AND PHRASES IN REQUIREMENTS DOCUMENTS IN A MULTILINGUAL ENVIRONMENT

Abstract

by

Breno Dantas Cruz

Vagueness in software requirements documents can lead to several future maintenance problems, especially when the customer and development team do not share the same language. Currently, companies rely on human translators to maintain communication and limit vagueness by translating the requirement documents by hand, which is expensive and time consuming. Because of that an automatic solution is desirable to reduce this effort. In this thesis, I present two approaches to identifying vagueness in requirements documents in a multilingual environment. Subsequently, I conduct two studies of these approaches for calibration purposes. In the first study, six participants, two native Portuguese speakers and four native Spanish speakers, evaluated both approaches. Then, I conduct a field study to test the performance of the best approach in real-world environments at two different companies.
CONTENTS

FIGURES ............................................................. iv

TABLES ............................................................. v

ACKNOWLEDGMENTS .............................................. vi

CHAPTER 1: INTRODUCTION ....................................... 1

CHAPTER 2: RELATED WORK ...................................... 3

CHAPTER 3: APPROACH ............................................ 8
    3.1 Approach for English .................................. 9
    3.2 Variants for Target Languages ....................... 11
    3.3 Implementation ..................................... 14

CHAPTER 4: PILOT STUDY PROCEDURE ......................... 15
    4.1 Research Questions .................................. 15
    4.2 Methodology ........................................ 16
    4.3 Subject Materials .................................. 18
    4.4 Threats to Validity .................................. 18

CHAPTER 5: PILOT STUDY RESULTS ............................ 19
    5.1 Answers to RQ1 and RQ2 ............................ 19
       5.1.1 RQ1: Comparison of Variants V1 and V2 .... 19
       5.1.2 RQ2: Configuration Settings ................... 20
    5.2 Summary of the Pilot Study Results ................ 23

CHAPTER 6: FIELD STUDY EVALUATION ...................... 24
    6.1 Research Questions .................................. 24
    6.2 Methodology ........................................ 25
    6.3 Subject Requirements ............................... 28
FIGURES

3.1 Architecture of the Approach for English .......................... 9
3.2 Architecture of Variant 1 ............................................. 11
3.3 Architecture of Variant 2 ............................................. 13
7.1 Ratings for User Studies at BC and MC ............................ 35
TABLES

5.1 AVERAGE PERFORMANCE VALUES OF VARIANT V1 AND VARIANT V2 .................................................. 20
5.2 NUMBER OF CHANGES TO PORTUGUESE FILTERS ACCORDING TO TIER ............................................ 21
5.3 PERFORMANCE VALUES ACCORDING TO TIER FOR PORTUGUESE DOCUMENTS ........................................ 22
5.4 NUMBER OF CHANGES TO SPANISH FILTERS ACCORDING TO TIER ...................................................... 23
6.1 CROSS-VALIDATION DESIGN OF USER STUDY ................. 26
7.1 STATISTICAL SUMMARY OF THE RESULTS FOR RQ3, RQ4, RQ5 AND RQ6 ............................................. 36
ACKNOWLEDGMENTS

The author would like to thank the analysts at the anonymous multinational corporation for providing their time and effort to evaluate our approach. Likewise, I thank the analysts from the anonymous Brazilian company for participating. Author Breno Cruz is supported by the Brazilian Science Without Borders program process number 13003-13-5. Finally, this work was partially supported by the NSF CCF-1452959 grant. Any opinions, findings, and conclusions expressed herein are the author’s and do not necessarily reflect those of the sponsors.
Requirements documents in software development projects are almost always written in a natural language such as English [26]. Using a natural language makes the documents easy to comprehend [1] and share. However, usage of a natural language also brings in the fallacies of language such as ambiguities and vagueness. As Berry and Kamsties point out, such problems in requirements confuse stakeholders, leading to “diverging expectations and inadequate or undesirably diverging implementations” [5].

Vagueness is the phenomenon which makes a statement have multiple interpretations due to a lack of precision. For example, in the statement, “The system should respond as fast as possible,” the interpretation of “as fast as possible” can range from milliseconds to days. It is widely believed that such problems in requirements, including vagueness, should be pointed out to the authors of those documents [5, 17, 30]. Automated tool support that detects problematic words and phrases is desirable because human readers tend to resolve the issues unconsciously and unknowingly [24]. In the above example, a human reader may believe that a response time of a few seconds would suffice. If his or her believed meaning is different from that of the author’s, then an inconsistency may exist without either person being aware of it. Automated tools can help prevent these
situations by bringing them to the reader’s attention so that he or she may resolve the vagueness consciously, possibly with the assistance of the author.

Many tools and techniques have been developed for the automated identification of ambiguities and vagueness in software requirements. These include the use of a manually curated checklist [15, 30], usage of linguistic parsers [28], and supervised machine learning techniques [31]. However, each of these works supports only a single language. My work in this thesis tackles the problem of vagueness in a multi-lingual context. In their work, Jain et al [15] showed the applicability of identification of vagueness through a list developed by subject matter experts. In this thesis, I show how the manually curated list developed for English can be used to identify vagueness across languages. My work technically develops a mechanism to transfer the learning from English to a multi-lingual context, thereby saving significant amounts of human effort in developing a checklist for different languages.

I evaluate my multi-lingual tool through a series of empirical studies in two companies and in two languages. My results provide key evidence to serve as recommendations for designers of vagueness detection tools in requirements as well as evidence about the problems experienced with requirements with vagueness in two multilingual industrial settings. Please note that the companies wish to remain anonymous. One company is a large multinational corporation (which we denote MC), and one is a small Brazilian software company (which we denote BC).
CHAPTER 2

RELATED WORK

The usage of natural language for the documentation of software requirements brings the shortcomings of language into software engineering. There are various fallacies in natural language such as ambiguities (where a statement can have multiple distinct interpretations), vagueness (where a statement can have a continuum of interpretations), uncertainty (where a statement weakens its proposition of truth), and others. In computational linguistics and related literature, these phenomena are considered distinct and are studied separately. In the domain of software engineering, such phenomena are generally grouped under the class of ambiguity [28] and studied accordingly. I will begin by looking into the related work in software engineering and will subsequently focus on the different linguistic phenomena as studied in the computational linguistics literature.

In the domain of software engineering, ambiguity is considered one of the key issues in the usage of natural language to document software requirements. Ambiguity is generally defined as the property leading to ‘multiple interpretations’ [3, 13]. It is also recognized that ambiguities can arise due to the linguistic characteristics of language (e.g. the phrase “as fast as possible” is ambiguous because is open to more than one interpretation and vague because is uncertain) as well as due to the analyst’s understanding of the domain (i.e. the knowledge of a domain can make an analyst interpret a requirement in multiple ways). An example of
a domain-specific ambiguity as presented by Kamsties et al. [16] is, “if a bank customer maintains a minimum balance in his or her account, there is no monthly service charge.” Here, knowledge of general practice in banking raises the ambiguity of whether the minimum balance is to be counted on a per day basis or is the average of the daily balance in a month. Ambiguities as in the latter example are referred to as ‘RE Context’ [16] or the ‘Software Engineering Context’ [3, 19]. It is also evident that the same understanding of the domain can actually help disambiguate a requirement. Kamsties et al. [16] provide the following example of ambiguous phrase, “generate a dial tone”: the characteristics of a dial tone may be precise to domain experts, yet ambiguous to others.

It is strongly believed that ambiguity in requirements should be tackled as early as possible in the creation of the documents. Ambiguous requirements can lead to building systems that miss stakeholder expectations [3, 31], involve costly rework [3, 10], or even invoke litigation [3]. Fixing errors later in the software development life cycle has been seen to increase project costs. In fact, Kiyavitskaya et al. [19] suggest such costs increase exponentially and Boehm et al. [6] report that the cost to fix a defect in the testing phase is 100 times more expensive than finding and fixing the defect in the requirements and design phases.

Recently, this linkage between ambiguity in requirements and overall project success was studied by De Bruijn et al. [8] and Philippo et al. [25]. De Bruijn et al. [8] point out that only one fault in the production system could be traced to an ambiguous requirement. Philippo et al. [25] state that a correlation between ambiguity in requirements and the budget overrun of a project could not be established. These studies show that the disambiguation of multiple interpretations gets done in the software development life cycle and does not directly surface at
the end of the project, either through the cost overruns or issues in production. However, these studies fail to measure the important conjecture that early identification of issues would improve project deliverables by either a quicker time to market or a lower cost. Yang et al. [31] and Kiyavitskaya et al. [19] report that the early identification of problems in requirements, like ambiguities, will be important toward the overall success of the project.

Ambiguity in natural language requirements have long been studied in software engineering. Most solutions use a set of look-up lists to identify words and phrases that are ambiguous. Such studies include the work of Berry et al. [3], Tjong et al. [28], and Gleich et al. [14]. Supervised machine learning based methods have been employed in works such as Yang et al. [31], Lucassen et al. [22], and Berry et al. [4].

In the work in computational linguistics, vagueness is said to be that property of words and phrases due to which you cannot assign a specific and precise meaning to a sentence [9, 11]. An example of a vague word is “tall”. It is not clear what constitutes a “tall” person. If a person of six feet in height is considered “tall”, then would a person of five feet and eleven inches not be considered “tall”? Such borderline cases characterize the vagueness phenomenon [18]. Vagueness is also said to be “inquiry resistant” [3, 18], i.e. no amount of empirical measurement of the height of a person would clarify whether the person is tall. In contrast to vagueness, ambiguity is when there are several but distinct and precise meaning to a sentence [3, 11].

The phenomenon of vagueness has an overlap with that of Hedging. Hedges generally refer to the linguistic devices that allow a speaker to be less committed to an expression (called Propositional Hedging) or to reduce the impact of the
expression (called Speech Act Hedging) [11, 31]. An example of Propositional Hedging would be, “He is around 6 feet in height.” An example of Speech Act Hedging would be, “Perhaps you should sit down?” Few forms of proportional hedging can lead to vagueness, as shown in the former example. However, not all forms of Propositional Hedging lead to vagueness. An example is “He may be 6 feet in height.” Here, “may” is considered to show the property of uncertainty and not vagueness. Thus vagueness has an overlap with Hedging, which in turn has an overlap with uncertainty. However, vagueness and uncertainty do not overlap. Some other examples of vague words and phrases include “easy”, “fast”, “sort of”, “e.t.c.”, “approximately” and “immediately.”

There have been several studies to automatically identify hedging and uncertainty. Some of the work includes Yang et al. [31], and Medlock et al. [23], which use supervised machine learning techniques. A semi-supervised learning approach has been made in Medlock et al. [23]. Most studies focus on English, though a few have been made in other languages, such as Hungarian as presented by Vincze [29].

However, each of the works in software engineering and computational linguistics focuses on a single language, largely English. To the best of our knowledge, our work is the first to study the multilingual context where the domain specific knowledge captured in English is transferred across languages for requirements documents. Our work also specifically tackles the phenomenon of vagueness which has been shown to be the most prominent category in software engineering.

Jain et al. [15] and Bucchiarone et al. [7] present the identification of problem phrases in software requirements using a list of words and phrases manually curated by business analysts. The list was specifically built for the requirements
typically seen in software engineering and consists of 189 entries. This list covers different phenomena including ambiguities, vagueness, and uncertainty. The majority of the entries (68%) map onto the phenomenon of vagueness. This tool has been used in several hundred projects and has seen good applicability. My experience with the usage of the tool has been that the phenomenon of vagueness is most commonly accepted by end users as a genuine problem in requirements (i.e. is a true-positive). A recent empirical study [8, 21] also showed that the phenomenon of vagueness is the most prominent category of the various problems in software requirements. I thus aim to study the identification of vagueness across languages.
CHAPTER 3

APPROACH

This chapter covers my approach to detecting the vague words and phrases in requirements documents. The input to my approach is a requirements document in natural language. The output is a markup of the requirements document that indicates the vague words and phrases. I first describe an approach for English documents. This English approach is based on a procedure in use at the multinational corporation (MC). Then, I describe two novel variants I created for Portuguese.

In general, the approach works in three phases: 1) parse the requirements document to divide it into sentences, 2) tag the parts of speech of each word in the sentences, and 3) filter in the adjectives and adverbs that match a blacklist of known vague terms. At present, the creation of the English blacklist is proprietary to MC. Note that the research contribution of the approach in this thesis is the application of the list to a multilingual environment, not the creation of the English blacklist. After executed the tool can be used by a specialist to check for potential ambiguous terms with the client in order to make the necessary corrections.
3.1 Approach for English

The architecture of the approach for English is in Figure 3.1. First, a sentence parser reads a requirements document to separate the document into a list of sentences for each requirement (area 1). The parser finds breaks in the document that separate the requirements, e.g. subsection headers or rows in a table, and

\footnote{Specifically, Brazilian Portuguese.}
then each requirement is broken into a list of sentences. In my implementation I used a sentence detector from the openNLP library [2]. Next, I used the parts-of-speech tagger in OpenNLP to mark each word in each sentence with its most likely type (area 2). Note that I did not perform text preprocessing such as splitting or stop-word removal, as it might have impacted the performance of the POS tagger, which may have had a significant impact in other implementations if the requirements document included some language similar to source code (e.g., camel case word combinations).

Following this, I filtered out any words that were not adjectives and adverbs (area 3). My rationale is that the vagueness from modifiers is more difficult for readers to resolve than vagueness from nouns or verbs. Then, I applied another filter to remove terms that were not in a blacklist of known vague terms (area 4). The blacklist I used was created in a proprietary process. I used the proprietary blacklist because it was in use at MC, which is where I conducted my evaluation. However, freely-available solutions exist that could be used to implement my approach in another environment [12]. Another possible alteration to my approach for some environments may be to skip the filtering for modifiers if the stakeholders in that environment experience difficulty resolving vagueness of words other than modifiers.

The final step was to markup the requirements document so that the vague words and phrases were readily available to stakeholders (area 5). In my implementation, I converted the requirements document to a PDF, and then added highlighting colors to each word that was detected as vague. I also created an index of the phrases that contained these words. The index is navigable via a PDF viewer.
3.2 Variants for Target Languages

I designed two variants of the tool for a target language. Both variants are adaptations of the English approach, using a machine translator at different stages of
the process. In the first variant (V1), shown in Figure 3.2, the procedure is identical
to the English approach, except that I use a machine translator to translate the
English blacklist into a blacklist for the target language (Figure 3.2, area 1). In
cases where one word in English has multiple translations in the target language,
the blacklist will contain all translations in the dictionary. The advantage of V1 is
that the machine translator can be relatively unsophisticated, as only dictionary
translations for single words are necessary. A disadvantage is that it is likely to
overestimate the blacklist. I explore the degree of this overestimation in Chapter 4.
I chose to implement the variants for Portuguese and for Spanish due the request
from MC. However it is possible to configure the machine translator to read and
translate other languages, which I did not explore in this study.

The second variant (V2), shown in Figure 3.3 uses a machine translator to
translate the requirements document from the target language into English (Fig-
ure 3.3, area 1). Then, the procedure is identical to the procedure for the English
tool until the vague terms list in English is converted back to a list in the tar-
get language (Figure 3.3, area 2). I accomplish this conversion by maintaining
an “alignment” list during the translation of the target language document into
English. The alignment document records which word in the target language is
translated into which word or words in English for each sentence. In Figure 3.3,
area 2, I use the alignment to mark the words in the target language as vague
that were marked as vague for English. While V2 does necessitate a more complex
machine translator than V1 (e.g. one capable of alignment), an advantage is that
the translator is less likely to overestimate the number of vague words because
the translator will only pick one word from the blacklist. For example, in V2,
the English word “should” will be translated as either “deveria” or “devia” in
a sentence, not both. In V1, “should” would be placed in the blacklist as both “deveria” and “devia.” The larger number of translations in V1 could result in reduced precision.
3.3 Implementation

I implemented both variants for Portuguese. I also implemented V2 for Spanish, given my experience evaluating V1 and V2 for Portuguese (see Chapter 4). The machine translator I used in both cases was version 1.0 of Moses MT [20]. I trained Moses MT using the parallel corpus from Europarl (available at www.statmt.org/europarl/) for Portuguese-English and for Spanish-English.
CHAPTER 4

PILOT STUDY PROCEDURE

This chapter describes two pilot studies I conducted. The first involves both V1 and V2 for Portuguese. The second, which is informed by the first study, involves only V2 for Spanish. Note that these pilot studies are intended to calibrate and inform my research; a full evaluation of the final results is in Chapter 7.

4.1 Research Questions

Broadly speaking, my objective for the pilot studies was to gain knowledge about how the variants behave for different languages. Due to resource constraints, it was not feasible to conduct a full scale evaluation on every possible configuration of the approach. Therefore, I pose the following two Research Questions in a limited pilot study:

\( RQ_1 \) What is the performance in terms of quality of V1 as compared to V2?

\( RQ_2 \) What terms are considered vague in the English blacklist that are not considered vague in the Portuguese and Spanish translations of that blacklist?

The purpose of \( RQ_1 \) is to determine how to allocate resources for future work on the variants. Since resources were only available to continue development and evaluation of one of the variants, I pose \( RQ_1 \) to provide data to assist in the
decision. Variant $V_1$ is a less-expensive option overall because the translation of the blacklist is limited to a dictionary. To justify the cost of the complex machine translation tool required for $V_2$, it is important that there is evidence that $V_2$ has improved performance.

The rationale behind $RQ_2$ is that some words that are vague in English are not necessarily vague in all languages. For example, the word “skin” in English is ambiguous because it could mean “body tissue” or a “to remove”. In contrast, the word is not vague in Portuguese because it is translated as “pele”, which definitively means the “body tissue” and not “to remove”. This pilot study is a first step in identifying the degree of this problem and correcting it for later development and evaluation. Note that this pilot study is intended to guide my development. A thorough evaluation of this research question is conducted in Chapter 5.

4.2 Methodology

The methodology that I used to answer $RQ_1$ and $RQ_2$ is a user study with a small number of bilingual human evaluators. I split my user study into two sections: first, one section for Portuguese, and second, a section for Spanish. To limit resource expenditure, I conducted the Portuguese study prior to the Spanish study and adapted my procedure based on the information I learned from the Portuguese section.

I addressed $RQ_1$ in the first study in Portuguese. Generally speaking, my procedure had two steps: 1) the author (a native speaker of Portuguese) created a goldset for six Portuguese requirements documents, and then 2) I compared the output of each variant on these documents to the goldset. In brief, these results
showed that V2 performed more strongly than V1. Full precision and recall values are reported in Chapter 5.

For $RQ_2$, I recruited two native Portuguese-speaking programmers (not otherwise affiliated with the authors) to evaluate the output of the V2 implementation for Portuguese. Likewise, I recruited four native Spanish-speaking programmers to evaluate the output of the Spanish V2. My procedure was to 1) build one survey containing the output of the V2 variant of the tool for each requirements document in each language, totaling six Portuguese surveys and six Spanish surveys, and then 2) to have human evaluators complete each survey.

A survey for one document showed each requirement followed by four radio buttons indicating four choices for every term that the tool highlighted as vague: Definitely Not Vague, Partially Not Vague, Partially Vague, and Definitely Vague. The choices were recorded as “scores” for each term: -1, -0.5, 0.5, and 1. I added the scores for each term for all human evaluators. For example, if two evaluators rated a term Definitely Vague, the term would receive a score of 2.

To answer $RQ_2$ from these results, I created three tiers of terms in the blacklist based on a three-tiered evaluation procedure used by Rodeghero et al. [27]. In one tier, I removed terms from the blacklist with scores less than or equal to -20. In a second tier, I removed terms with scores -10 or less. In a third tier, I removed terms with scores of 0 or less. The size and contents of these tiers allows us to answer $RQ_2$ (see Chapter 5) and improve the performance of my approach for the field studies (see Chapter 6).
4.3 Subject Materials

The key subject materials in my pilot study were the Portuguese and Spanish requirements documents. I identified a range of requirements documents that have been released in the public domain. The documents range in size from seven requirements up to forty requirements. Since these requirements are publicly available, I provide copies at my online appendix for reproducibility (URL is in Section 6.5).

4.4 Threats to Validity

As a pilot study, this section contains threats to validity not suitable for a thorough evaluation – I caution that my results were intended for calibration and exploration in a resource-restricted commercial environment. (For a thorough evaluation see my field study in Chapter 6). The three key threats to validity include the small number of human evaluators, the limited set of requirements documents and the goldsets created by the author. It is possible that results would vary with different evaluators or requirements.
CHAPTER 5

PILOT STUDY RESULTS

In this chapter, I present my answer to $RQ_1$ and $RQ_2$ as well as my data and rationale. These answers are the basis for the improvements of the field studies, which I present in Chapter 6.

5.1 Answers to RQ$_1$ and RQ$_2$

5.1.1 RQ$_1$: Comparison of Variants V1 and V2

For Portuguese, I found evidence that variant V2 is better than variant V1 due to higher values of precision and recall. The data supporting this finding is in Table 5.1. Table 5.1 shows that the precision and the recall of variant V2 is greater than the precision and the recall of variant V1. Variant V2 has term precision of 3.64% and term recall of 32.20%, while variant V1 has a lower term precision of 2.90% and a lower term recall of 11.93%. For snippet$^1$ detection, variant V2 is also better than variant V1. Variant V2 has a snippet precision of 34.01% and a snippet recall of 94.17%, while variant V1 has a snippet precision of 33.69% and a snippet recall of 90.56%. The superior quality of variant V2 means that it detects more vague terms than variant V1.

$^1$A snippet is a is a higher level of granularity when compared to a term. I considered a snippet as a sentence ending in a period that contains a vague term.
TABLE 5.1

AVERAGE PERFORMANCE VALUES OF VARIANT V1 AND VARIANT V2

<table>
<thead>
<tr>
<th>Variant</th>
<th>Term Precision</th>
<th>Term Recall</th>
<th>Snippet Precision</th>
<th>Snippet Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>2.90%</td>
<td>11.93%</td>
<td>33.69%</td>
<td>90.56%</td>
</tr>
<tr>
<td>V2</td>
<td>3.64%</td>
<td>32.20%</td>
<td>34.01%</td>
<td>94.17%</td>
</tr>
</tbody>
</table>

For Spanish, since none of the authors were native Spanish speakers, it was not possible to create a gold set for gathering precision and recall. Therefore, I decided to follow the same procedure described for the Portuguese pilot study. I collected the terms and comments from the questionnaires and applied them in a similar aggressive way to create a similar set of configuration files. These files were later used for the Spanish field studies, available in Chapter 6.

5.1.2 RQ$_2$: Configuration Settings

For Portuguese, I found evidence that adding or removing terms to the variant V2 blacklist and whitelist can improve precision and recall. Variant V2 has better precision than the original setting when using aggressive settings. Table 5.2 contains the number of modifications made to the blacklist and whitelist. The least aggressive filter is tier 1, which contains one change to the blacklist and two

---

2The whitelist is a document that contains terms that are not vague. The whitelist was also developed by the multinational company. The initial idea was to use the information gathered by the pilot study to improve the whitelist and to use it in future studies. The results from the whitelist do not impact the evaluation of my approaches.
TABLE 5.2

NUMBER OF CHANGES TO PORTUGUESE FILTERS
ACCORDING TO TIER

<table>
<thead>
<tr>
<th></th>
<th>Tier 1</th>
<th>Tier 2</th>
<th>Tier 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blacklist added</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Blacklist removed</td>
<td>1</td>
<td>7</td>
<td>24</td>
</tr>
<tr>
<td>Whitelist added</td>
<td>2</td>
<td>25</td>
<td>111</td>
</tr>
<tr>
<td>Whitelist removed</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

changes to the whitelist. Note that in the aggressive approach, tier 3, I added one hundred and eleven terms to the whitelist and removed twenty-four from the blacklist – more than twice the combined amount from the other two tiers. Table 5.3 shows the values of precision and recall for the different tiers. All tiers had better values of term precision and snippet precision when compared to the original setting. The highest term precision of 4.91% came from the aggressive approach. Even though Table 5.3 shows that tier 2 has greater values for snippet precision and snippet recall, I decided to use tier 3 because it incorporates comments and suggestions from the questionnaires and because it has the best term precision. These suggestions and comments were not added to lower tiers because, according to my rating system, they received values that best fit the tier 3 configuration.

For Spanish, I found evidence that the Spanish configuration files will have differences from the Portuguese files. According to my findings, while using Spanish, variant V2 requires changes to both blacklist and whitelist that are not present
TABLE 5.3

PERFORMANCE VALUES ACCORDING TO TIER FOR PORTUGUESE DOCUMENTS

<table>
<thead>
<tr>
<th>Setting</th>
<th>Term Precision</th>
<th>Term Recall</th>
<th>Snippet Precision</th>
<th>Snippet Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tier 1</td>
<td>4.23%</td>
<td>32.20%</td>
<td>34.52%</td>
<td>82.40%</td>
</tr>
<tr>
<td>Tier 2</td>
<td>4.85%</td>
<td>32.20%</td>
<td>35.70%</td>
<td>93.06%</td>
</tr>
<tr>
<td>Tier 3</td>
<td>4.91%</td>
<td>31.09%</td>
<td>35.12%</td>
<td>90.83%</td>
</tr>
<tr>
<td>Original</td>
<td>3.64%</td>
<td>32.20%</td>
<td>34.01%</td>
<td>94.17%</td>
</tr>
</tbody>
</table>

in the Portuguese blacklist and whitelist. A case where it is possible to see the difference between the Portuguese filter and the Spanish filter is in Table 5.4. The difference is the number of additions to the Spanish whitelist for aggressive configuration, tier 3, compared to the Portuguese, tier 3. The Spanish tier 3 whitelist received hundred and ninety-seven additions, whereas the Portuguese tier 3 whitelist received one hundred and eleven additions. Table 5.4 shows the number of alterations that each different Spanish tier received. Note that when using the aggressive filters, tier 3, it was necessary to make more changes to the original filters. Also, unlike for Portuguese filters, for Spanish filters it was necessary to add terms to the blacklist and remove terms from the whitelist.
### TABLE 5.4

<table>
<thead>
<tr>
<th></th>
<th>Tier 1</th>
<th>Tier 2</th>
<th>Tier 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blacklist added</td>
<td>0</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>Blacklist removed</td>
<td>5</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>Whitelist added</td>
<td>21</td>
<td>17</td>
<td>197</td>
</tr>
<tr>
<td>Whitelist removed</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

5.2 Summary of the Pilot Study Results

I derived two results from my pilot study. First, variant V2 has better precision and recall when compared to variant V1. I decided to use the aggressive set of filters, tier 3, as the configuration files of variant V2 because it contained feedback not present in other tiers and obtained the best overall term precision. Table 5.2 shows the number of changes that the Portuguese tiers received. Table 5.4 shows the changes to Spanish tiers. Table 5.3 shows the variant V2 values from precision and recall when processing Portuguese documents. My conclusion to the pilot study is that, when using variant V2, each language requires different alterations to its configuration files in order to reach desirable precision and recall.
CHAPTER 6

FIELD STUDY EVALUATION

This section describes the research questions and methodology of the three field studies of my approach. The field studies were used in conjunction with a qualitative study, available in Chapter 8, to answer the research question. One field study was conducted at BC (in Portuguese), and two others were conducted at MC (one in Spanish and one in Portuguese). During the pilot study (detailed in Chapters 4 and 5), I found that $V_2$ had a higher level of performance than $V_1$. In these field studies, I evaluate $V_2$ in industry.

6.1 Research Questions

My research objective is to evaluate $V_2$ in both Portuguese and Spanish, in an industrial setting. As a result of the pilot study, I found that even though $V_2$ outperformed $V_1$, there was evidence of further improvement after different adjustments to the blacklist in each language. Therefore, I pose the following research questions:

$RQ_3$ What is the degree of difference in performance of $V_2$ for the less aggressive (original) blacklist versus the more aggressive (tier 3) blacklist in Portuguese?
RQ$_4$ What is the degree of difference in performance of V2 for the less aggressive (original) blacklist versus the more aggressive (tier 3) blacklist in Spanish?

RQ$_5$ What is the degree of difference in performance of V2 for the less aggressive (original) blacklist versus the more aggressive (tier 3) blacklist at the MC?

RQ$_6$ What is the degree of difference in performance of V2 for the less aggressive (original) blacklist versus the more aggressive (tier 3) blacklist at the BC?

The rationale behind RQ$_3$ and RQ$_4$ is that since the approach is intended for use in a multilingual environment, the process for adapting V2 to different languages involves modifying the blacklists via the pilot study. While it is likely that the pilot study procedure leads to higher performance (given that it involves tuning to an expert-generated goldset), before recommending this procedure it is necessary to evaluate its impact. Thus, I pose RQ$_3$ and RQ$_4$.

The rationale behind RQ$_5$ and RQ$_6$ is that it is necessary to objectively determine how the performance of V2 varies according to different types of development teams. For example, developers of MC may consider a term vague, while developers from BC, a smaller company, may not perceive it as such. This difference in how terms are perceived by specialists will cause differences in performance. Therefore, I pose RQ$_5$ and RQ$_6$.

6.2 Methodology

I used a cross-validation user study design and a qualitative study (available in Chapter 8) to answer my research questions. An overview of this design can be found in Table 6.2. In this design, different groups of human experts evaluated the output of different tools when those tools were given different datasets as inputs.
I rotated the tools and datasets over three sessions. The purpose of this rotation was to avoid biases that were introduced by the human experts, such as fatigue. Also, “learning bias” is reduced since the experts see the tools in various orders.

In my experiment, I rotated two tools over three sessions with three groups. Typically, two tools would be evaluated over two sessions by two groups in a cross-validation design. Due to the extreme financial pressure involved in disrupting human experts in an industrial environment, it was necessary to collocate my experiment with a second unrelated experiment. For clarity and reproducibility, I report the full study design, but not the results of the unrelated study. The rows in Table 6.2 marked “null” may be ignored for this thesis. Since the rotation of tools and datasets is preserved, it is my view that collocating these experiments...

<table>
<thead>
<tr>
<th>Session</th>
<th>Group</th>
<th>Tool</th>
<th>Dataset</th>
<th>Survey</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>L1</td>
<td>P-A1</td>
<td>S1</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>L2</td>
<td>P-A2</td>
<td>S2</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>null</td>
<td>null</td>
<td>null</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
<td>null</td>
<td>null</td>
<td>null</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>L1</td>
<td>P-A3</td>
<td>S3</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>L2</td>
<td>P-A1</td>
<td>S4</td>
</tr>
<tr>
<td>3</td>
<td>A</td>
<td>L2</td>
<td>P-A3</td>
<td>S5</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>null</td>
<td>null</td>
<td>null</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>L1</td>
<td>P-A2</td>
<td>S6</td>
</tr>
</tbody>
</table>
has not reduced the quality of my results.

The tools in the study are two configurations of \( V^2 \). The first, \( L_1 \), is \( V^2 \) prior to any modifications from the pilot study (marked “original” during the analysis of \( RQ_2 \)). The second, \( L_2 \), is also \( V^2 \), but with the “tier 3” filter applied. The tier 3 filter was determined during and after the pilot study, as described in Chapter 5. Note that the filter is different for each language: I created the Portuguese filter for Portuguese and the Spanish filter for Spanish.

The datasets that I used were derived from proprietary requirements documents from both MC and BC. Legal agreements prevent us from releasing these requirements, so analysts at each company only evaluated requirements from their own organization. I divided the requirements at each company into three equally-sized datasets. In Table 6.2, I refer to these as P-A1, P-A2, and P-A3 for Portuguese-Multinational Corporation.

I conducted three experiments with three sessions each. Table 6.2 shows only one experiment: at MC in Portuguese. A second experiment was done at MC in Spanish and a third experiment was done at a BC in Portuguese. The second and third experiment had a design identical to the first experiment (Table 6.2), except that I used different datasets. I conducted a total of nine sessions across all three experiments.

During each session, the human analysts read a series of software requirements from the datasets. I highlighted the terms in the requirements that the tool identified as vague. The analysts then rated each term on a Likert scale from 1 to 4, where 1 was Very Vague, 2 was Vague, 3 was Somewhat Vague, and 4 was Not Vague. In Spanish these translated to Muy ambiguo, Ambiguo, Parcialmente ambiguo, and Sin ambigüedades. In Portuguese, the options were Muito Ambíguo,
I used the following procedure to analyze the experiments’ results. First, I grouped the terms according their vagueness classification. Each classification received a specific weight. For instance, Very Vague receives the weight 1.5, Vague received 1.0, Partially Vague 0.5, and Not Vague received the negative weight -1.0. I chose these values because the distinction between how vague or how ambiguous is a term is directly correlate to the specialist’s interpretation. Thus, those are not strong assertions and need to be balanced. I classify the Not Vague value to be just as important as the Very Vague, because I need this information to, if the case requires it, remove undesirable terms from the blacklist. To calculate the vagueness value of a term, the weights are multiplied by the number of word occurrences while adding the weights to create a total. The total is divided by the number of that term’s occurrences throughout all surveys in the experiment. I mapped the occurrences of each term to a matrix that contains survey identifiers. The matrix contains value of 1 if the term is present in the survey and 0 otherwise. This matrix is then multiplied by the terms’ normalized values. The resulting matrix is used for the Wilcoxon nonparametric tests.

6.3 Subject Requirements

The summaries in the study correspond to software requirements from two different sources for the Portuguese field study and from one source for the Spanish field study. The two Portuguese field studies received two datasets of different sizes. Both datasets contained requirements written in Brazilian Portuguese. For the Portuguese MC-based field study, I used 450 requirements, half functional requirements and half non-functional. Those requirements were identified by a
number and were described in a paragraph. The BC-based field study received a smaller dataset with high-level requirements. All high-level requirements were described in paragraphs without identification. The Spanish MC-based field study received a very detailed dataset of functional and non-functional requirements. The dataset contain around 375 requirements, all identified by number and described by a paragraph.

6.4 Participants

I had a total of six participants in my field study. The Portuguese portion of the field study had four participants, two software engineers from the BC and two from MC. The two professionals from the BC were all bi-lingual, being native Portuguese speakers (Brazilian dialect) and proficient English speakers. The two software engineers from the BC had programming and requirements elicitation experience. The two professionals from MC had programming and requirements elicitation experience. Both MC professionals spoke at least two languages; one was a native Portuguese speaker and the other was fluent in Portuguese. The two professionals were advanced English speakers. The two participants from the MC who have taken the Spanish field study were all Spanish native speakers. Both Spanish speaking professionals from the MC had more than two years of experience with programming and with requirements elicitation. They were both native speakers from Latin America and were capable of speaking more than two languages.
6.5 Reproducibility

For the purposes of reproducibility and independent study, I have made all non-proprietary data available via an online appendix:

http://www3.nd.edu/~bdantasc/projects/adetection/

6.6 Threats to Validity

As with any study, my evaluation also carries threats to validity. I identified the following sources of validity threats. First, my evaluation was conducted by human experts who may be influenced by factors such as stress, fatigue, or variations in software development experience. I attempted to mitigate these threats through my cross-validation study design, which altered the order in which the participants viewed the results from V2. I also recruited participants from different areas of expertise and different levels of experience. Another possible source of a threat to validity is the set of requirements I selected. I chose sets of requirements from the respective companies. For the MC, I chose two groups of requirements provided by the same MC, one for Portuguese and another for Spanish. For the BC, I was provided with a recent document used for software development by the company. The last threat that I identified is the accuracy of the OpenNLP POS tagger. My approach evaluates adjectives and adverbs, and those terms are identified by the tagger. If the tagger is not accurate, vague terms can be overlooked by the tool. I confirmed my results with accepted statistical testing procedures; still, I cannot guarantee that a different group of participants would not have produced a different result.
CHAPTER 7

FIELD STUDIES RESULTS

In this chapter, I present my preliminary answers to \( RQ_3, RQ_4, RQ_5, \) and \( RQ_6, \) as well my data and rationale based upon the results gathered from the field studies. The full answer to these research questions in is Chapter 6, where I preset the results of my qualitative study.

7.1 Preliminary Answers to Questions RQ3, RQ4, RQ5 and RQ6

7.1.1 RQ3: V2 Portuguese Performance Level

For Portuguese, I found that the variant \( V2 \) had higher performance with the original setting in the MC field study. In the BC field study, the tool had a higher performance with the tier 3 filter. For the MC field study, from a total of 649 samples, where a sample is a term deemed as vague by the tool (the same term can be detected several times), Table 7 shows that the average vagueness value of -0.194 for the original setting, was higher than the value of -0.366 from tier 3, which means that on average the results from the original setting are closer to correctly detecting vague terms.

For the BC field study, from a total of 214 samples, Table 7 shows that the average vagueness value of -1.453 for the original setting, was lower than the value of -1.171 from tier 3, which means that on average the tier 3’ setting is
closer to correctly detecting vague terms. Figure 7.1 shows the amount of term occurrences found by the tool for Portuguese in both MC and BC studies. For the MC, Figure 7.1 shows the increase of Not Vague terms detected by V2 when using the tier 3 filter. For the BC, Figure 7.1 shows a decrease in Not Vague and Very Vague terms detected and an increase of Somewhat Vague terms detected.

7.1.2 RQ4: V2 Spanish Performance Level

For Spanish, I found that the variant V2 had higher performance with the tier 3 setting. From a total of 315 samples, Table 7 shows that the average vagueness value is -2.224 for the tier 3 and -2.383 for the original, which means that on average the results from the tier 3 setting are closer to correctly detecting vague terms. Figure 7.1 shows the difference in performance of V2 while using the tier 3 and original filter. According to Figure 7.1 there is a decrease in the detection of Not Vague terms detected and an increase of Somewhat Vague and Vague terms detected when comparing the original filter and the tier 3 filter.

7.1.3 RQ5: MC Performance Difference of Original Versus Tier 3 Filter

For the MC, I found statistical significance that there is a difference in performance when using the tool for Spanish or for Portuguese. The original setting had higher performance than the tier 3 for Portuguese documents, while tier 3 had higher performance than original for Spanish documents. The statistical information that supports this claim is displayed in Table 7. Table 7 shows that my tier 3 setting had worse performance when compared to the original setting for Portuguese. The mean for the original setting was -0.194, while the mean for tier 3 was -0.366. The negative mean means that, on average, most of the output
was considered as not vague by the experts. The variance of original was 2.877, and the variance tier 3 was 4.028. This means that terms detected by original were considered, on average, more ambiguous than terms detected by the tier 3 setting for Portuguese documents in MC. I used the Wilcoxon test, and it gave me the values of 50285.000, expected value of 37539.000, variance 4685475.000, and p-value <1e-4. On the other hand, for Spanish, I found statistical significance that tier 3 had higher performance than the original setting in the MC Spanish experiment. The statistical information that supports this finding is displayed in Table 7. Table 7 shows that my tier 3 setting had a superior performance when compared to the original setting. The mean for the original setting was -2.383, while the mean for tier 3 was -2.224. The variance of original was 112.042, and the variance of tier 3 setting was 113.911. This means that terms detected by tier 3 were considered, on average, more ambiguous than terms detected by the original setting for Spanish documents. I used the Wilcoxon test, and it gave me the values of 4060.000, expected value of 6847.500, variance 350597.625, and p-value <1e-4.

7.1.4 RQ6: BC Performance Difference of Original Versus Tier 3 Filter

For BC, I found a result different from the result I found in MC. I found statistical significance that the tier 3 setting had higher performance than the original for Portuguese documents. The statistical information that supports this claim is displayed in Table 7. Table 7 shows that my original setting had worse performance when compared to the tier 3 setting. The mean for the tier 3 setting was -1.171, while the mean for original was -1.453. The negative mean means that, on average, most of the output was considered as not vague by the experts.
The variance of original was 27.136, and the variance of tier 3 setting was 4.554. This means that terms detected by tier 3 were, on average, more ambiguous than terms detected by the original setting for Portuguese documents in the MC. I used the nonparametric Wilcoxon test and it gave me the values of 7589.000, expected value of 6930.500, variance 376339.250, and p-value 0.027.

7.2 Summary of the Field Study Results

I derived several preliminary results from my field study. First, depending on the working environment, there will be differences in performance of the tool. As was pointed out, for Portuguese the tier 3 filter setting had higher performance for the BC documents, which are from a small software development company. For MC documents, which are from a big multinational corporation, the original setting had higher performance. Second, depending on the number of samples, there will be a difference in the performance: for both the Portuguese BC and the Spanish MC, where the number of samples was less than half of the samples from the Portuguese MC, the tool had better performance with the aggressive tier 3 filter. Lastly, depending on the language, there will be differences in the vagueness values, independent from the filters settings. For instance, for both Portuguese studies the vagueness values were higher than the values from the Spanish study. My conclusion from the field study is that, according to the preliminary data, variant V2 for both original and tier 3 filters can be used for vagueness detection. The tier 3 filter can be used for smaller documents, while the less aggressive original filter can be used for larger documents. The choice of which to select will vary according to the target language and the size of the document.
Figure 7.1: Ratings for User Studies at BC and MC
TABLE 7.1: STATISTICAL SUMMARY OF THE RESULTS FOR RQ₃, RQ₄, RQ₅ AND RQ₆

<table>
<thead>
<tr>
<th>RQ</th>
<th>Samples</th>
<th>Experiment</th>
<th>Tool Version</th>
<th>SD</th>
<th>µ</th>
<th>Vari.</th>
<th>U</th>
<th>U_{expt}</th>
<th>U_{vari}</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ₃</td>
<td>649</td>
<td>MC Portuguese</td>
<td>Original Tier 3</td>
<td>1.696</td>
<td>-0.194</td>
<td>2.877</td>
<td>50285.000</td>
<td>37539.000</td>
<td>4685475.000</td>
<td>&lt;1e-4</td>
</tr>
<tr>
<td></td>
<td>214</td>
<td>BC Portuguese</td>
<td>Original Tier 3</td>
<td>5.209</td>
<td>-1.453</td>
<td>27.136</td>
<td>7589.000</td>
<td>6930.500</td>
<td>376339.250</td>
<td>0.283</td>
</tr>
<tr>
<td>RQ₄</td>
<td>315</td>
<td>MC Spanish</td>
<td>Original Tier 3</td>
<td>10.585</td>
<td>-2.383</td>
<td>112.042</td>
<td>4060.000</td>
<td>6847.500</td>
<td>350597.625</td>
<td>&lt;1e-4</td>
</tr>
<tr>
<td>RQ₅</td>
<td>649</td>
<td>MC Portuguese</td>
<td>Original Tier 3</td>
<td>1.696</td>
<td>-0.194</td>
<td>2.877</td>
<td>50285.000</td>
<td>37539.000</td>
<td>4685475.000</td>
<td>&lt;1e-4</td>
</tr>
<tr>
<td></td>
<td>315</td>
<td>MC Spanish</td>
<td>Original Tier 3</td>
<td>10.585</td>
<td>-2.383</td>
<td>112.042</td>
<td>4060.000</td>
<td>6847.500</td>
<td>350597.625</td>
<td>&lt;1e-4</td>
</tr>
<tr>
<td>RQ₆</td>
<td>214</td>
<td>BC Portuguese</td>
<td>Original Tier 3</td>
<td>5.209</td>
<td>-1.453</td>
<td>27.136</td>
<td>7589.000</td>
<td>6930.500</td>
<td>376339.250</td>
<td>0.283</td>
</tr>
</tbody>
</table>

*aWilcoxon test values are $U$, $U_{expt}$, and $U_{vari}$ and $p$. A “Sample” is an output from V2.*
CHAPTER 8

QUALITATIVE STUDY RESULTS

In this chapter, I explore the opinions and feedback for my approach. I use those responses in conjunction with the data gathered during the field study, available in Chapter 7, to fully answer RQ3, RQ4, RQ5, and RQ6. Open-ended questions for feedback and comments were available throughout the pilot study. In the open-ended questions, I asked participants to put words that were present in the output from the tool. In addition, several participants used the open-ended questions to provide optional comments and feedback about the tool’s output.

8.1 Answers to Questions RQ3, RQ4, RQ5 and RQ6

8.1.1 RQ3: V2 Portuguese Performance Level

For Portuguese, most of the open-ended answers point out vague terms that were not detected by the tool – for instance, the comment, “Word ‘devem’ was not marked.” There were open-ended answers where the participant stated that in the context of that phrase the word could be considered as vague – for example, the comment, “In this context the word ‘pertencia’ can also be considered ambiguous.” These answers show that there are vague words, but there are also words that, depending on the context, can be perceived as vague.
In the field study there are differences for the results of BC and MC. For BC the results show an increase in performance of tier 3, which is related to the removal of not vague terms from the blacklist. While for MC the results show a decrease in performance of the tool when using the tier 3. The decrease in performance in the Portuguese MC study is explained by how the analysts interpret the requirements. For the Portuguese MC study both outputs from the tool were equal. This means that the removed words in tier 3 were already not tested by original filter and the difference in performance was cause by how the specialists analyzed the requirements, as explained by the Portuguese open-ended answers. The data displaying the performance for Portuguese documents is available in Chapter 7, Subsection 7.1.1, which shows that the original filter provided more precise results when compared to the more aggressive filter for MC and that the tier 3 performed better at the BC study.

8.1.2 RQ4: V2 Spanish Performance Level

For Spanish, the open-ended questions show that the participants were more concerned with the meaning and grammar of a phrase. One of the participants stated that, “Just words are not vague” and another stated, “Words by themselves are not vague, but the way the paragraph was written is really hard to read.” These statements show that Spanish speakers analyze the vagueness taking into account grammar and the way that the phrase was constructed. The removal of terms from tier 3 caused the decrease in the detection of false positives, which lead to an increase in performance. The data displaying the increase in performance for Spanish documents is available in Chapter 7, Subsection 7.1.2, which shows that the use of the more aggressive blacklist tier 3 blacklist caused an increase in
performance of the tool.

8.1.3 RQ$_3$: MC Performance Difference of Original Versus Tier 3 Filter

The open-ended questions can explain the increase in the performance of the tool when using the more aggressive tier 3 filter for the MC study for Spanish and a decrease in performance for Portuguese terms.

As previously explained in question RQ$_3$, the Portuguese-speaking participants were more likely to focus on the meaning of words alone and the context that the words are being used. For the Portuguese MC study the output of the tool for the original and tier 3 filter were equal, the difference in performance between the tier 3 filter and the original was related to how the participants analyzed the context of the phrases. Fact that can explain why the tool had a reduced performance when using the tier 3 for the Portuguese MC study. For Spanish results show what would be expected. The removed words from the tier 3 caused the tool to detect less false positives causing an increase in performance.

8.1.4 RQ$_6$: BC Performance Difference of Original Versus Tier 3 Filter

When analyzing the BC results for the performance of the tool, I identified a different behavior to the one found in MC for the Portuguese study. In BC the removal of not vague terms in tier 3 caused the decrease in the detection of false positives by BC. The decrease in the detection of not vague terms improved the performance of the tool when using the tier 3 filter.
8.2 Qualitative Results Discussion

I received two types of answers to open-ended questions. One type of answer would focus mainly on specific words that were missing or were wrongly pointed out by the tool. The other type of answer would focus on the context of a phrase, stating that the words alone do not generate ambiguity. The answers can be linked to the native language of the respondents. The majority of comments from Spanish speakers were regarding grammar and phrase structure from the analyzing documents. Spanish speakers also tended to provide a couple of synonyms to the detected words. Portuguese answers more strictly followed the command from the open-ended question, which was, “(Optional) if you see one or more words that are not present in the output, please inform.” Portuguese speakers did not provide additional feedback or comments, but did provide the additional vague words not detected by the tool.

The following are some examples of comments from Portuguese-speaking participants:

- “The word ‘devia’ is missing”
- “Missing word ‘e’”
- “In this context the word ‘pertencia’ can also be considered ambiguous.”
- “Word ‘devem’ was not marked”
- “What does ‘RL’s’ stands for?”

The following are examples of translated comments from Spanish-speaking participants:

- “Do you mean {⋯} or {⋯}?”
“Words by themselves are not vague, but the way the paragraph was written is really hard to read.”

“Just words are not vague”.

“‘tanto secundaria como primaria’ should be ‘tanto primaria como secundaria’”

“paquetera”

From the comments, I infer that there is a difference in how different language speakers perceive vagueness while reading a text. The Portuguese participants of the study were more concerned with the meaning of words and how they would create the sense of vagueness in the context of the phrase. Spanish-speaking participants of the study were more concerned with grammar and the overall language structure than vague words. Spanish respondents gave several comments indicating that they were more concerned with how words were used, providing synonyms and concern about the overall structure.
CHAPTER 9

CONCLUSION

In this thesis, I have presented a novel approach for automatically detecting vagueness in software requirements documents. I have modified a proprietary algorithm provided by the multinational corporation to work in conjunction with a machine translation tool in order to output the vague words. I conducted a set of pilot studies to calibrate the tool for improving its performance in terms of precision and recall for different languages. Our pilot studies aimed to improve the capability of the tool to detect vague terms in Spanish and in Portuguese. In two cross-validation studies, I compared the summaries from our approach to determine what the performance of the tool would be for different languages and in different types of work environments. I found that our tier 3 filter had a better performance for detecting Spanish vague terms, while the original filter had a better performance for detecting Portuguese vague terms.

I have a few recommendations for future work. First, I recommend using variant V2 because I found evidence that variant V2 is capable of finding vague terms for both Portuguese and Spanish. Still, a second recommendation is to invest resources to conduct a pilot study of the filter configuration, since I did not find a result consistent for both languages. Note that the difference for each language is consistent within that language, as I found in the statistical tests; thus, one filter configuration may be valid for one language, but not for another. If used
for Spanish, I recommend using the tier 3 filter configuration, since I found that tier 3 outperformed the original filter for Spanish. On the other hand, for Portuguese I recommend using the original filter. If variant V2 is used for any other language, an investment into a pilot study is worthwhile, similar to the procedure I describe in Chapter 4. A third recommendation is to expand the existing approach to analyze the general structure of a phrase, in order to detect cases where ambiguity is created due to the way a phrase is written. A last recommendation for future work is the creation of blacklists for specific languages without using the translated blacklist, in order to compare the performance between a list created by native speakers with a list translated from English.
BIBLIOGRAPHY

1. C. Arora, M. Sabetzadeh, L. Briand, and F. Zimmer.


22. G. Lucassen, F. Dalpiaz, J. M. E. van der Werf, and S. Brinkkemper. Forg-


