Detecting Vague Words & Phrases in Requirements Documents in a Multilingual Environment

Breno D. Cruz*, Bargav Jayaraman†, Anurag Dwarakanath†, and Collin McMillan*
*Department of Computer Science and Engineering
University of Notre Dame, Notre Dame, IN, USA
Email: {bdantasc, cmc}@nd.edu
†Accenture Technology Labs India
Bangalore, Karnataka, India
Email: {bargav.jayaraman, anurag.dwarakanath}@accenture.com

Abstract—**Vagueness in software requirements documents can lead to several 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. In this paper, we describe two approaches that automatically identify vagueness in requirements documents in a multilingual environment. We perform two studies for calibration purposes under strict industrial limitations, and describe the tool that we ultimately deploy. In the first study, six participants, two native Portuguese speakers and four native Spanish speakers, evaluated both approaches. Then, we conducted a field study to test the performance of the best approach in real-world environments at two companies. We describe several lessons learned for research and industrial deployment.**

I. Introduction

Requirements documents in software development projects are almost always written in a natural language such as English [27]. Using natural language makes the documents easy to comprehend [1] and share. However, usage of 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" [6].

Vagueness is the phenomenon which makes a statement have multiple interpretations due to a lack of precision. Examples include "The system should respond *as fast as possible*". Here the interpretation of "as fast as" can range from a few milliseconds to a few days. It is widely believed that such problems in requirements including vagueness should be pointed out to the authors of the documents [6], [17], [31]. Automated tool support that detects problematic words and phrases is desirable because human readers tend to resolve the issues unconsciously and unknowingly [25]. 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 [31], [16], usage of linguistic parsers [29] and supervised machine learning techniques [32]. These tools point out ambiguities and vagueness in a single natural language.

Research Problem & Industrial Context The context of our work is a large multinational corporation which has an internally-developed tool that detects vagueness in requirements documents *in English only*. The English-only tool is essentially a blacklist that highlights words that are in a manually curated list of words that are frequently vague. While relatively simple conceptually, the blacklist was extremely expensive to create and is popular among the company's clients. The company views the tool as an asset and competitive advantage.

The company has clients in several countries in which English is not a first language. Therefore, the company is eager to answer the research problem: can the English-only tool be adapted to serve their non-English-speaking customers through the use of existing machine translation technology? I.e., without the need to manually curate a new blacklist.

Our Strategy to Address Research Problem We proposed two modifications to the existing English-only tool, and tested these modifications in studies with domain experts at two companies: the large multinational (denoted MC throughout the paper for client confidentiality), and a small Brazilian software developer (BC). We studied vagueness detection in two target languages: Spanish and Brazilian Portuguese. The modifications are, generally speaking, different locations at which we place a machine translator. First, we attempt the least expensive idea: to translate the blacklist directly from English into a target language. Second, we build a more complex approach: to translate the entire requirements document from the target language into English, use the English-only tool to mark the document, and then use the word alignment from the translator to highlight the vague words in the target language document. Details and evaluation of the first modification are in Sections III and V. For the second modification, see Sections III, V, and VII.

We found several important "lessons learned" for other researchers, which we present in Section VIII.

II. Background & Related Work

The usage of natural language for the documentation of software requirements brings the shortcoming 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 studied separately. In the domain of software engineering, such phenomena is generally grouped under the class of ambiguity [29] and studied accordingly.

In 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' [5], [14]. It is also recognized that ambiguities can arise due to the linguistic characteristics of language (e.g. the phrase 'as fast as possible', which is ambiguous, because it is open to more than one interpretation, and vague, because it is uncertain) and 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 in [18] is "if a bank customer maintains a minimum balance in his or her account, there is no monthly service charge". Here, the knowledge of the 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' [18] or the 'Software Engineering Context' [5], [20]. It is also evident that the same understanding of the domain can actually help disambiguate a requirement. An example from [18] is the phrase 'generate a dial tone'. The characteristics of a dial tone, may be precise to domain experts but 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 [5], [32], involve costly rework [5], [11], or even invoke litigation [5]. It is seen that fixing errors later in the software development life cycle significantly increases project costs. Kiyavitskaya et al. [20] suggest such costs increase exponentially. Boehm et al. [7] 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.

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. [5], Tjong et al. [29] and Gleich et al. [15]. Supervised learning methods have also been employed [32], [23], [4].

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 [12], [10]. An example of a vague word is 'tall'. It isn't clear what constitutes a 'tall' person. If a person of 6 feet in height is considered tall, then would a person of 5 feet 11 inches not be considered tall? Such borderline cases characterize the vagueness phenomenon [19]. In contrast to vagueness, ambiguity is when there are several but distinct and precise meaning to a sentence [12], [5]. Vagueness is also said to be 'inquiry resistant' [19], [5] i.e. no amount of empirical measurement of the height of a person would clarify if the person is tall.

There have been several studies to automatically identify hedging and uncertainty, which are related to vagueness. Some of the work includes [32], [24] which use supervised machine learning techniques. A semi-supervised learning approach has been made in [24]. Most studies focus on English and a few have been made in other languages such as Hungarian [30].

Jain et al. [16] and Bucchiarone et al. [8] 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%) maps to the phenomenon of vagueness. This tool has been used in several hundred projects and has seen good applicability. Our experience on 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 [9], [22] also showed that the phenomenon of vagueness is the most prominent category of the various problems in software requirements. We thus aim to study the identification of vagueness across languages.

However, all the works in software engineering and computational linguistics focus on a single language, usually English. To the best of our knowledge, our work is the first to study the multi-lingual context where the domain specific knowledge of vagueness captured in English is transferred across languages for requirements documents.

III. Our Approach

This section covers our approach to detect the vague words and phrases in requirements documents. The input to our approach is a requirements document in natural language. The output is a markup of the requirements document that indicates the vague words and phrases. We first describe an approach for English documents. This English approach is based on a procedure in use at the Multinational Corporation (MC). Then, we describe two novel variants we created for Portuguese[1].

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, 3) filter the words for **adjectives** and **adverbs** that match a blacklist of known vague terms or does not match a whitelist of not vague terms. At present, the creation of the English blacklist and English whitelist is proprietary to MC. Note that the research

[1]Specifically, Brazilian Portuguese.

contribution of the approach in this paper is the application of the list to a multilingual environment, not the creation of the English blacklist or English whitelist.

A. Approach for English

The architecture of the approach for English is in Figure 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. Then each requirement is broken into a list of sentences; in our implementation we used a sentence detector from the the openNLP library [2]. Next, we use the parts-of-speech tagger in OpenNLP to mark each word in each sentence with its most likely type (area 2). Note that we do not perform text preprocessing such as splitting or stop-word removal, since it may impact the performance of the POS tagger and may also have a significant impact in other implementations if the requirements document includes some language similar to source code (e.g., camel case word combinations).

We then filter the parts of speech to remove any words that are not modifiers, e.g. adjectives and adverbs (area 3). Our rationale is that the vagueness from modifiers is more difficult for readers to resolve than vagueness from nouns or verbs, as discussed in Section II. Then, we apply another filter to remove words that are not in a blacklist of known vague terms (area 4). The blacklist we used is created in a proprietary process. We used the proprietary blacklist because it was in use at MC, which is where we conduct our evaluation. However, freely-available solutions exist that could be used to implement our approach in another environment [13]. In addition to the blacklist we also use a whitelist for cases where a term is not present in the blacklist. The whitelist contains words that are considered as not vague. The whitelist we used is also created in a proprietary process. Another possible alteration to our 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 is to markup the requirements document so that the vague words and phrases are readily-available to stakeholders (area 5). In our implementation, we convert the requirements document to a PDF, and then add highlighting colors to each word that is detected as vague. We also create an index of the phrases that contain these words. The index is navigable via a PDF viewer.

B. Variants for Target Languages

We 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 2, the procedure is identical to the English approach, except that we use a machine translator to translate the English blacklist and English Whitelist into a blacklist and whitelist for the target language (Figure 2, area 1). In cases where one word in English has

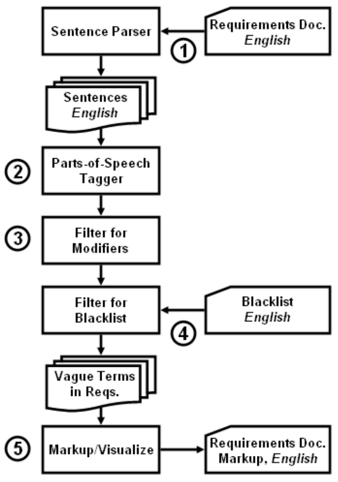

Fig. 1. Architecture of the Approach for English. Details describing this approach are in Subsection III-B

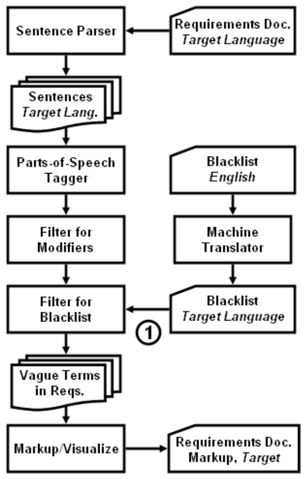

Fig. 2. Architecture of Variant 1, an adaption of the English tool to a target language. Details describing this Variant are in Subsection III-B

multiple translations in the target language, the blacklist and the whitelist 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 and whitelist. We explore the degree of this overestimation in Section IV. We choose to implement the variants for Portuguese and for Spanish due to the request from MC. However it is possible to configure the machine translator to read and translate other languages, which we did not explore in this study.

The second variant (`V2`), shown in Figure 3, uses a machine translator to translate the requirements document from the target language into English (Figure 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 target language (Figure 3, area 2). We accomplish this conversion by maintaining an "alignment" list during the translation of the target language document into English. The

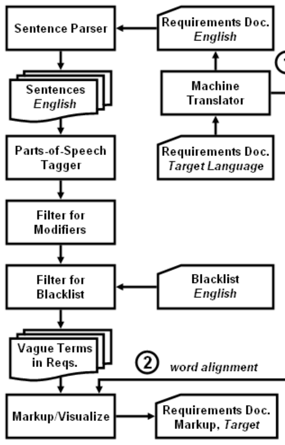

Fig. 3. Architecture of Variant 2, an alternative adaption of the English tool to a target language. Details describing this Variant are in Subsection III-B

alignment document records which word in the target language is translated into which word or words in English, for each sentence. In Figure 3 area 2, we 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 Portuguese 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.

C. Implementation

We implemented both variants for Portuguese. We also implemented V2 for Spanish, given our experience evaluating V1 and V2 for Portuguese (see Section IV). The machine translator we used in both cases was version 1.0 of Moses [21].

IV. Pilot Study Procedure

This section describes two pilot studies we 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 our research; a full evaluation of the final results is in Section VII.

A. Research Questions

Broadly speaking, our objective for the pilot studies is to gain knowledge about how the variants behave for different languages. Due to resource constraints, it is not feasible to conduct a full scale evaluation on every possible configuration of the approach. Therefore, we 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, we pose RQ_1 to provide data to assist in the decision. Variant $V1$ is a less-expensive option overall because the translation of a blacklist is limited to a dictionary. To justify the cost of the complex machine translation tool required for $V2$, it is important that there is evidence that $V2$ 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". But in Portuguese, the word is not vague because it is translated as "pele" , which definitively means the body tissue and not to remove. This pilot study is a first step to identifying the degree of this problem, and correcting it for later development and evaluation. Note that this pilot study is intended to guide our development – a thorough evaluation of this research question is conducted in Section V.

B. Methodology

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

We addressed RQ_1 in the first study in Portuguese. Generally speaking, our procedure had two steps: 1) the first author (a native speaker of Portuguese) created a goldset for six Portuguese requirements documents, then 2) we compared the output of each variant on these documents, to the goldset. Full precision and recall values are reported in Section V – in brief, these results showed that $V2$ performed better than $V1$.

For RQ_2, we recruited two native Portuguese-speaking programmers (not otherwise affiliated with the authors) to evaluate the output of the $V2$ implementation for Portuguese. Likewise, we recruited four native Spanish-speaking programmers to evaluate the output of the Spanish $V2$. Our procedure was to: 1) build one survey containing the output of the tool (both variants) for each requirements document in each language, totaling six Portuguese surveys and six Spanish surveys. Then, 2) the human evaluators completed each survey.

A survey for one document showed each requirement, followed by four radio buttons indicating four choices for every word 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 word: -1, -0.5, 0.5, and 1 respectively. We added the scores for each word for all human evaluators. E.g., if two evaluators rated a word Definitely Vague, the word would receive a score of 2.

To answer RQ_2 from these results, we created three "tiers" of words in the blacklist, based on a three-tiered evaluation procedure used by Rodeghero *et al.* [28]. In one tier, we removed words from the blacklist with scores less than or equal to -20. In a second tier, we removed words with scores -10 or less. In a third tier, we removed words with scores of 0 or less. The size and contents of these tiers allows us to answer RQ_2 (see Section V and improve the performance of our approach for the field studies (see Section VI).

C. Subject Materials

The key subject materials in our pilot study were the Portuguese and Spanish requirements documents. We identified a range of requirements documents that have been released in the public domain. The documents range in size from seven requirements up to 40 requirements. Since these requirements are publicly available, we will provide copies at our online appendix for reproducibility post the acceptance of this paper in Section VI-B3.

D. Threats to Validity

As a pilot study, this section contains threats to validity not suitable for a thorough evaluation – we caution that our results were intended for calibration and exploration in a resource-restricted commercial environment. (For a more thorough evaluation see our field study in Section VI.) 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 first author. It is possible that results would vary with different evaluators or requirements.

V. PILOT STUDY RESULTS

In this section, we present our answer to RQ_1 and RQ_2, as well as our data and rationale. These answers are the basis for the improvements of the field studies, which we present in section VI.

TABLE I
TABLE SHOWING THE AVERAGE VALUES OF PRECISION AND RECALL FOR VARIANT V1 AND VARIANT V2.

Variant	Word Precision	Word Recall	Phrase Precision	Phrase Recall
V1	2.90%	11.93%	33.69%	90.56%
V2	3.64%	32.20%	34.01%	94.17%

TABLE II
TABLE WITH PRECISION AND RECALL FOR THE DIFFERENT TIERS FOR PORTUGUESE DOCUMENTS.

Setting	Word Precision	Word Recall	Phrase Precision	Phrase Recall
Tier 1	4.23%	32.20%	34.52%	82.40%
Tier 2	4.85%	32.20%	35.70%	93.06%
Tier 3	4.91%	31.09%	35.12%	90.83%
Original	3.64%	32.20%	34.01%	94.17%

A. RQ_1: Comparison of Variants V1 and V2

For Portuguese we found evidence that variant `V2` is better than variant `V1` in terms of precision and recall. The data supporting this finding is in Table III. Table III shows that precision and the recall of variant `V2` is greater than the precision and the recall of variant `V1`. Variant `V2` has word precision of 3.64% and word recall of 32.20%, while variant `V1` has lower word precision of 2.90% and lower word recall of 11.93%. For phrase detection variant `V2` is also better than variant `V1`. Variant `V2` has a phrase precision of 34.01% and phrase recall of 94.17%, while variant `V1` has 33.69% for phrase precision and 90.56% for phrase recall. The superior quality of variant `V2` means that it detects correctly more vague words than variant `V1`.

For Spanish, since none of the authors were native Spanish speaker, it was not possible to create a gold set for gathering precision and recall. Therefore, we decided to follow the same procedure described for Portuguese pilot study. We collected the words and comments from the questionnaires and applied 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 section VI.

B. RQ_2: Configuration Settings

For Portuguese we found evidence that adding or removing terms to variant `V2` blacklist and whitelist (list of terms that are considered definitely not vague) can improve precision and recall. Variant `V2` has better precision than the original setting when using aggressive settings. Table IV 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 one change to the whitelist. Note that in the aggressive approach, tier 3, we added 78 words to the whitelist and removed 21 from the blacklist more than twice the combined amount from the other two tiers. Table V shows that the values of precision and recall for the different tiers. All tiers had better values of word precision and phrase precision when compared to the original setting. The highest word precision 4.91% came from the aggressive approach. Even though Table V shows that tier 2 has greater values for

TABLE III
TABLE SHOWING THE AVERAGE VALUES OF PRECISION AND RECALL FOR VARIANT V1 AND VARIANT V2.

Variant	Word Precision	Word Recall	Phrase Precision	Phrase Recall
V1	2.90%	11.93%	33.69%	90.56%
V2	3.64%	32.20%	34.01%	94.17%

TABLE IV
TABLE SHOWING THE NUMBER OF CHANGES TO PORTUGUESE BLACKLIST AND WHITELIST ACCORDING TO TIER.

		Tier 1	Tier 2	Tier 3
Blacklist	added	0	0	0
	removed	1	4	21
Whitelist	added	1	16	78
	removed	0	0	0

TABLE V
TABLE WITH PRECISION AND RECALL FOR THE DIFFERENT TIERS FOR PORTUGUESE DOCUMENTS.

Setting	Word Precision	Word Recall	Phrase Precision	Phrase Recall
Tier 1	4.23%	32.20%	34.52%	82.40%
Tier 2	4.85%	32.20%	35.70%	93.06%
Tier 3	4.91%	31.09%	35.12%	90.83%
Original	3.64%	32.20%	34.01%	94.17%

TABLE VI
TABLE SHOWING THE NUMBER OF CHANGES TO SPANISH BLACKLIST AND WHITELIST ACCORDING TO TIER.

		Tier 1	Tier 2	Tier 3
Blacklist	added	0	0	8
	removed	4	2	6
Whitelist	added	19	15	156
	removed	0	0	1

phrase precision and phrase recall. We decided to use tier 3, because it incorporates comments and suggestions from the questionnaires and because it has the best word precision. These suggestions and comments were not added to lower tiers, because, according to our rating system, they received values that fit the tier 3 configuration.

For Spanish we found evidence that the Spanish configuration files will have differences from the Portuguese files. According to our findings, while using Spanish variant V2 requires changes to both whitelist and blacklist that are not present 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 VI. The difference is number of additions to the Spanish whitelist for aggressive configuration, tier 3, compared to the Portuguese, tier 3. The Spanish tier 3 whitelist received 156 additions, while in Portuguese tier 3 whitelist 78 additions. Table VI 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, differently from Portuguese filters, for Spanish filters it was necessary to add words to the blacklist and remove words from the whitelist.

C. Summary of the Pilot Study Results

We derived two results from our pilot study. First, variant V2 has better precision and recall, when compared to variant V1. We decided to use the aggressive set of filters, tier 3, as configuration files of variant V2, because it contains feedback not present in different tiers and it obtained the best overall word precision. Table IV shows the number of changes that the Portuguese tiers received. Table VI shows the changes to Spanish tiers. Table V shows variant V2 values from precision and recall processing Portuguese documents. Our 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.

VI. FIELD STUDY EVALUATIONS

This section describes the research questions and methodology of three field studies of our approach: one at a BC (in Portuguese), and two at MC (one in Spanish and one in Portuguese). During the Pilot Study in Sections IV and V, we found that $V2$ had a higher level of performance than $V1$. In these field studies, we evaluate $V2$ in industry.

A. Research Questions

Our research objective is to evaluate $V2$ in both Portuguese and Spanish, in an industrial setting. As a result of the pilot studies, we found that even though $V2$ outperformed $V1$, there was evidence of further improvement after different adjustments to the blacklist in each language. Therefore, we pose the following research questions:

RQ_3 What is the degree of difference in performance of $V2$ 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 for Portuguese and Spanish?

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 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 an expert-generated gold set), before recommending this procedure it is necessary to evaluate its impact. Thus, we 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 words are perceived by specialists will cause differences in performance. Therefore, we pose RQ$_5$ and RQ$_6$.

B. Methodology

We used a cross-validation user study design to answer our research questions. An overview of this design is in Table VII. In this design, different groups of human experts evaluated the output of different tools, when those tools were given different datasets as inputs. We rotated the tools and datasets over three sessions. The purpose of this rotation is to avoid biases that are introduced from the human experts, such as fatigue. Also, "learning bias" is reduced, since the experts see the tools in various orders.

In our experiment, we 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 colocate our experiment with a second, unrelated experiment. For clarity and reproducibility, we report the full study design, but not the results of the unrelated study. The rows in Table VII marked "null" may be ignored for this paper. Since the rotation of tools and datasets is preserved, it is our view that colocating these experiments has not reduced the quality of our results.

The tools in the study are two configurations of $V2$. The first, $L1$, is $V2$ prior to any modifications from the pilot study (marked "original" during the analysis of RQ_2). The second, $L2$ is also $V2$, but with the "tier 3" filter applied. The "tier 3" filter was determined during and after the pilot study, as described in Section V. Note that the filter is different for each language: we created the Portuguese filter for Portuguese, and the Spanish filter for Spanish (see Section IV).

The datasets that 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. We divided the requirements at each company into three equally-sized datasets. In Table VII, we refer to these as P-A1, P-A2, and P-A3, for Portuguese-MC, etc.

We conducted three experiments, with three sessions each. Table VII shows only one experiment: at MC in Portuguese. A second experiment was done at MC in Spanish, and a third experiment was done at BC in Portuguese. The second and third experiment had the identical design as the first experiment (Table VII), except that we used different datasets. We 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. We highlighted the words in the requirements that the tool identified as vague. The analysts then rated each word 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", "Ambíguo", "Parcialmente Ambíguo", "Não Ambíguo."

TABLE VII
THE CROSS-VALIDATION DESIGN OF OUR USER STUDY FOR $V2$. THE GROUP MARKED AS "NULL" PARTICIPATED IN AN UNRELATED STUDY, AND THOSE RESULTS ARE NOT REPORTED IN THIS PAPER.

Session	Group	Tool	Dataset	Survey
1	A	L1	P-A1	S1
	B	L2	P-A2	S2
	C	*null*	*null*	*null*
2	A	*null*	*null*	*null*
	B	L1	P-A3	S3
	C	L2	P-A1	S4
3	A	L2	P-A3	S5
	B	*null*	*null*	*null*
	C	L1	P-A2	S6

We used the following procedure to analyze the experiments' results. First, we grouped the words according its vagueness classification. Each classification received a specific weight. For instance "Very Vague" receives the weight 1.5, "Vague" 1.0, "Partially Vague" 0.5 and "Not Vague" the negative weight -1.0. We chose these values, because the word classification of how vague or how ambiguous is subjective to specialist interpretation. Thus those are not strong assertions and need to be balanced. We classify the "Not Vague" value as important as "Very Vague", because we need this information to remove not vague worlds from the blacklist. To calculate the vagueness value of a word, the weights were multiplied by the number of occurrences while adding them and totaled. The total is divided by the number of its occurrences through all surveys in a experiment. We mapped the occurrences of each word to a matrix that contains surveys identifiers. The matrix contains value of 1, if the word is present in the survey and 0 otherwise. This matrix is then multiplied by the words normalized values. The resulting matrix is used for the Wilcoxon nonparametric tests.

1) Subject Requirements: The summaries in the study corresponds 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 we used 450 requirements, half functional requirements and half non functional. Each requirement was identified by a number and was described in a paragraph. The *BC* based field study received a smaller dataset with high level requirements. All high level requirements was described in paragraphs without identification. The *Spanish MC* based field study received very detailed dataset of functional and non functional requirements. The dataset contained around 375 requirements all identified by number and described in a paragraph.

2) Participants: We had a total of 6 participants in our field study. For the Portuguese portion of the field study had 4 participants, 2 software engineers from the BC and 2 from MC. The 2 professionals from the BC were all bi-lingual, being native Portuguese speakers (Brazilian dialect) and proficient English speaks. The 2 software engineer from the BC had programming and requirements elicitation experience. The 2 professionals from MC had a programming and requirements elicitation experience. Both MC professionals spoke at least two languages, one was native Portuguese speaker and the other was fluent in Portuguese. Both professionals were advanced English speakers. The 2 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 2 years of experience with programming and with requirements elicitation. They were all native from Latin American and were capable of speaking more than 2 languages.

3) Reproducibility: For the purposes of reproducibility and independent study, we have made all non proprietary data available via an online appendix:
`http://www.cse.nd.edu/~cmc/projects/vague/`

4) Threats to Validity: As with any study, our evaluation also carries threats to validity. We identified the following sources of validity threats. First, our evaluation was conducted by human experts, who may be influenced by factors such as stress, fatigue, or variations in software development experience. We attempted to mitigate these threats though our cross-validation study design, which altered the order in which the participants viewed the results from $V2$. We also recruited our participants from different areas of expertise and different levels of experience. Another source for a threat to validity is the set of requirements we selected. We chose sets of requirements from the respective companies. For MC we chose two groups of requirements provided by the same MC, one for Portuguese and other for Spanish. For the BC, they provided us with a recent document used for software development by the company. The last threat that we identified is the accuracy of the OpenNLP POS tagger. Our approach evaluates adjectives and adverbs identified by the tagger. If the tagger is not accurate vague terms can be overlooked by the tool. We confirmed our results with accepted statistical testing procedures. Still, we cannot guarantee that a different group of participants would not produce a different result.

VII. Field Studies Results

In this section, we present our answer to RQ_3, RQ_4, RQ_5 and RQ_6 as well our data and rationale.

A. RQ_3: V2 Portuguese performance level

For Portuguese we found that 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 the average vagueness value of -0.194, for the "original" setting, was higher than the value of -0.366 from "tier 3".

For the BC field study the average vagueness value of the detected words was -1.453, for the "original" setting, was lower than the value of -1.171 from "tier 3".

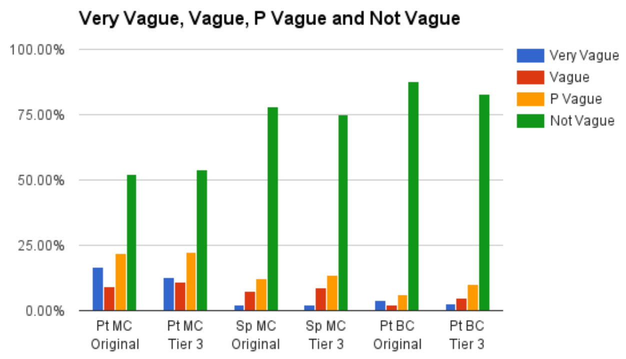

Fig. 4. Distribution of the analysts rating during the MC Portuguese, MC Spanish and BC Portuguese study for the terms.

Table VIII shows the statistical information for RQ$_3$ found by the tool for MC and BC studies. Figure 4 shows the classification of the detected words for Portuguese MC and BC studies.

The difference in performance can be understood as the difference in the context that the requirements are insert on. For example a word can be detected as a vague term by our tool, but for a certain development team this word can be the name of a functionality or name for a identifier in the project, which is not vague for that context. So words can have its vagueness changed according context that the development team is in.

B. RQ_4: V2 Spanish performance level

For Spanish we found that variant `V2` had higher performance with the "tier 3" setting. The average vagueness value of the detected words by for the variant while using "tier 3" was higher than the value while using the "original" filter. Table VIII shows that the average vagueness value of -2.224 for the "tier 3" and -2.383 for the "original" filter. Figure 4 shows the classification of the detected words for Spanish MC.

C. RQ_5: MC performance difference of "original" versus "tier 3" filter

For the MC we found statistical significance that there is a difference in performance when using the tool for either Spanish or for Portuguese. The "original" setting had higher performance than the "tier 3" for Portuguese documents,

TABLE VIII
Statistical summary of the results for RQ$_3$, RQ$_4$, RQ$_5$ and RQ$_6$. Wilcoxon test values are U, U_{expt}, and U_{vari} and p.

RQ	Samples	Experiment	Tool Version	$Std.Dev$	μ	Vari.	U	U_{expt}	U_{vari}	p
RQ_3	649	MC Portuguese	Original Tier 3	1.696 2.007	-0.194 -0.366	2.877 4.028	50285.000	37539.000	4685475.000	<1e-4
	214	BC Portuguese	Original Tier 3	5.209 2.134	-1.453 -1.171	27.136 4.554	7589.000	6930.500	376339.250	0.283
RQ_4	315	MC Spanish	Original Tier 3	10.585 10.673	-2.383 -2.224	112.042 113.911	4060.000	6847.500	350597.625	<1e-4
RQ_5	649	MC Portuguese	Original Tier 3	1.696 2.007	-0.194 -0.366	2.877 4.028	50285.000	37539.000	4685475.000	<1e-4
	315	MC Spanish	Original Tier 3	10.585 10.673	-2.383 -2.224	112.042 113.911	4060.000	6847.500	350597.625	<1e-4
RQ_6	214	BC Portuguese	Original Tier 3	5.209 2.134	-1.453 -1.171	27.136 4.554	7589.000	6930.500	376339.250	0.283

while “tier 3” had higher performance than “original” for Spanish documents. The statistical information is available in Table VIII.

D. RQ_6: BC performance difference of “original” versus “tier 3” filter

For BC, we found a different result to the result we found in MC for Portuguese. We 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 VIII. The mean for the “tier 3” setting was -1.171, while the mean for “original” was -1.453 for the Portuguese document.

E. Summary of the Field Study Results

We derived several results from our 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. We also derived that the context of the text is also important for the vagueness detection. The same word in different texts can have different connotations for different development teams. 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. Our conclusion to the Field Study is that for variant `V2` 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 the target language and the size of the document.

VIII. Conclusions and Lessons Learned

This research project involved international cooperation of several researchers and project domain experts over two years across the United States, India, and Brazil. In this section, we highlight a few key lessons learned for the benefit of teams attempting a similar project.

a) Recall vs. Precision: It has long been observed that, for many situations, recall is more important than precision [26], [3]. The idea is that it does not cost a reader much effort to discard false positive results, but that the cost of a missing correct answer (a false negative) is unknown, and potentially very high. We gently dispute the generalizability of this idea to the context of detecting vague requirements. The reason is twofold. First, it was our experience that false positives damage the trust that the domain experts placed in the tool. In practice, programmers did not desire to give every result the same scrutiny if there were many false positives – the domain experts in the studies in this paper had to be specifically instructed and compensated to do so.

A second reason is that false negatives are not all alike. In our work, some vague phrases cause high risk while others are much lower risk. As noted in Figure 4, there are a substantial number of cases in the “Very Vague” category. It is important to maintain high recall for this group. In contrast, it is better to accept lower recall in trade for higher precision for the less critical “Partially Vague” group. Given that “Partially Vague” and “Not Vague” cases outnumber the “Vague” and “Very Vague” cases, it is preferable to trade recall to gain precision overall because we focused on separating “Not Vague” from other groups. Our advice to future researchers in this area is to focus on identifying what separates the “Partially Vague”, “Vague”, and “Very Vague” cases, and then adopt different precision/recall balances for each group.

b) Academia vs. Industry: We brushed against a few cultural differences between academia and industry. One way to summarize the misunderstandings is that in academia, value is given to the most “theoretically correct” approach, even if it is more complex. In industry, value is placed on simple and maintainable low-cost approaches, even if it means compromising perfection. This difference impacted our work in a few ways. For example, academic feedback was frequently in the direction of curating a new list for each language. That solution is the highest quality, but it is not feasible due to cost. The pilot study in Section IV is another example. The least expensive option by far would be to use a dictionary to translate the blacklist, using every possible translation word. That approach is certain to decrease precision because there is no context for the translation, and words have multiple meanings (e.g., the English word “bear”). Nonetheless, it was necessary to at least test that approach, given the potentially very high cost savings if the approach performed better than expected. We found the academic world to be less understanding of these “moon shot” cost tests than industry. Note that we do not imply a value judgement about academic or industrial research, though we do recommend that future researchers consider how these differences may affect their work.

c) Semi-automated Solution is Likely: Our aim at the start of the project was to design a fully-automatic approach. While this remains our long-term goal, a likely interim step is a semi-automated solution in which the machine translator is used for the first version of the blacklist in a new target language, that a human expert modifies. The cost savings would still be substantial compared to a creating a new blacklist. Also, our study results indicate that `V2` produces results that are generally acceptable to the customer. One way to add human expert feedback to the process is during the Markup/Visualize phase (recall Figure 3): the markup component could filter certain words that a human expert has determined not to be vague in the target language. Our recommendation to other researchers is to consider the possibility of a semi-auotmated solution earlier, and to quantify a number of hours that the expert would need to spend on the semi-automated solution. Those hours could be compared to the effort used to manually curate the English blacklist to estimate the cost advantage.

d) Vagueness across Languages: We found that vagueness seems to be perceived differently across languages. While we cannot make a strong generalizable statement on the subject, our experience was that the English and Portuguese speakers were concerned with the vagueness of individual words, whereas the Spanish speakers were not concerned with individual words, but concentrated more on their comprehension of the high level concepts in phrases and sentences. For example, representative comments from Portuguese speakers include (translated to English):

- "The word "devia" is missing"
- Missing word "e"
- Word "devem" was not marked
- What does it stands for "RL's"

Whereas examples of translated comments from Spanish speakers include:

- "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", which can be translated to either "small van" or "parcel shelf"

Again, we cannot make a strong generalizable statement because of our limited sample sizes, and it is not appropriate for us to speculate on the root cause for the difference. Still, our experience was a surprise, and led us to recommend highlighting vague phrases in the Spanish tool, versus vague words in the English and Portuguese tools, for this particular set of customers. Our recommendation to other researchers is to consider that the granularity of the results from the approach might vary due to external factors such as language or culture.

IX. Acknowledgements

The authors thank the analysts at the anonymous Multinational Corporation for providing time and effort to evaluate our approach. Likewise, we thank the analysts from the anonymous Brazilian Company for participating. Author Breno Cruz is supported by the Capes foundation 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 authors' and do not necessarily reflect those of the sponsors.

References

[1] C. Arora, M. Sabetzadeh, L. Briand, and F. Zimmer. Automated checking of conformance to requirements templates using natural language processing.

[2] J. Baldridge. The opennlp project. *URL: http://opennlp.apache.org/index.html (accessed 31 Jan 2017).*

[3] S. Banerjee and A. Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL Workshop*, volume 29, pages 65–72, 2005.

[4] D. Berry, R. Gacitua, P. Sawyer, and S. F. Tjong. The case for dumb requirements engineering tools. In *Requirements Engineering: Foundation for Software Quality*, pages 211–217. Springer, 2012.

[5] D. Berry, E. Kamsties, and M. Krieger. From contract drafting to software specification: Linguistic sources of ambiguity. *Online at URL https://cs.uwaterloo.ca/~dberry/handbook/ambiguityHandbook.pdf*, pages 1–80, 2003.

[6] D. M. Berry and E. Kamsties. Ambiguity in requirements specification. In *Perspectives on software requirements*, pages 7–44. Springer, 2004.

[7] B. Boehm and V. R. Basili. Software defect reduction top 10 list. *Foundations of empirical software engineering: the legacy of Victor R. Basili*, 426, 2005.

[8] A. Bucchiarone, S. Gnesi, and P. Pierini. Quality analysis of nl requirements: an industrial case study. In *RE*, pages 390–394, 2005.

[9] F. De Bruijn and H. L. Dekkers. *Ambiguity in natural language software requirements: A case study*, volume 6182 LNCS, pages 233–247. 2010.

[10] N. Drave. Vaguely speaking: a corpus approach to vague language in intercultural conversations. *Language and Computers*, 2001.

[11] H. Femmer, D. M. Fernández, E. Juergens, M. Klose, I. Zimmer, and J. Zimmer. Rapid requirements checks with requirements smells: two case studies. In *RCSE Workshop*, pages 10–19, 2014.

[12] B. Fraser. Pragmatic competence: The case of hedging. *New approaches to hedging*, pages 15–34, 2010.

[13] V. Ganter. Detecting vagueness in two languages and two domains. *g Semantic Approaches in Natural Language Processing*, page 141, 2010.

[14] V. Gervasi and D. Zowghi. On the role of ambiguity in re. *Requirements Engineering: Foundation for Software Quality*, pages 248–254, 2010.

[15] B. Gleich, O. Creighton, and L. Kof. Ambiguity detection: Towards a tool explaining ambiguity sources. In *Requirements Engineering: Foundation for Software Quality*, pages 218–232. Springer, 2010.

[16] P. Jain, K. Verma, A. Kass, and R. G. Vasquez. Automated review of natural language requirements documents: generating useful warnings with user-extensible glossaries driving a simple state machine. In *India SE conference*, pages 37–46. ACM, 2009.

[17] E. Kamsties and B. Peach. Taming ambiguity in natural language requirements. In *ICSSEA*, 2000.

[18] E. Kamsties and B. Peach. Taming ambiguity in natural language requirements. *ICSSEA*, pages 1–8, 2000.

[19] R. Keefe, J. Dancy, J. Haldane, and G. Harman. *Theories of vagueness.*

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

[21] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, et al. Moses: Open source toolkit for statistical machine translation. In *ACL Tool Demos*, pages 177–180. ACL, 2007.

[22] S. J. Korner, W. F. Tichy, J. Keim, J. Krisch, et al. Denom: a tool to find problematic nominalizations using nlp. In *AIRE*, pages 1–8, 2015.

[23] G. Lucassen, F. Dalpiaz, J. M. E. van der Werf, and S. Brinkkemper. Forging high-quality user stories: Towards a discipline for agile requirements. In *RE*, pages 126–135. IEEE, 2015.

[24] B. Medlock and T. Briscoe. Weakly Supervised Learning for Hedge Classification in Scientific Literature - Microsoft Research. *ACL*, pages 992–999, 2007.

[25] J. M. Novick, A. Kim, and J. C. Trueswell. Studying the grammatical aspects of word recognition: Lexical priming, parsing, and syntactic ambiguity resolution. *Journal of Psy. Res.*, 32(1):57–75, 2003.

[26] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: bringing order to the web. 1999.

[27] K. Pohl. *Requirements Engineering: Fundamentals, Principles, and Techniques*. Springer, 1st edition, 2010.

[28] P. Rodeghero, C. McMillan, P. W. McBurney, N. Bosch, and S. D'Mello. Improving automated source code summarization via an eye-tracking study of programmers. In *ICSE*, pages 390–401, 2014.

[29] S. Tjong and D. Berry. The design of sree – a prototype potential ambiguity finder for requirements specifications and lessons learned. *Proceedings of REFSQ*, pages 80–95, 2013.

[30] V. Vincze. Uncertainty detection in Hungarian texts. *Proceedings of Coling*, pages 1844–1853, 2014.

[31] W. M. Wilson. Writing effective natural language requirements specifications. *Naval Research Laboratory*, 1999.

[32] H. Yang, A. De Roeck, V. Gervasi, A. Willis, and B. Nuseibeh. Speculative requirements: Automatic detection of uncertainty in natural language requirements. *RE*, pages 11–20, Sep 2012.