2016 IEEE/ACM 13th Working Conference on Mining Software Repositories

Sentiment Analysis in Tickets for IT Support

Cássio Castaldi Araujo Blaz
Instituto de Informática
Universidade Federal do Rio Grande do Sul
Porto Alegre, Brazil
ccablaz@inf.ufrgs.br

Karin Becker
Instituto de Informática
Universidade Federal do Rio Grande do Sul
Porto Alegre, Brazil
karin.becker@inf.ufrgs.br

ABSTRACT

Sentiment analysis has been adopted in software engineering for problems such as software usability and sentiment of developers in open-source projects. This paper proposes a method to evaluate the sentiment contained in tickets for IT (Information Technology) support.IT tickets are broad in coverage (e.g. infrastructure, software), and involve errors, incidents, requests, etc. The main challenge is to automatically distinguish between factual information, which is intrinsically negative (e.g. error description), from the sentiment embedded in the description. Our approach is to automatically create a Domain Dictionary that contains terms with sentiment in the IT context, used to filter terms in ticket for sentiment analysis. We experiment and evaluate three approaches for calculating the polarity of terms in tickets. Our study was developed using 34,895 tickets from five organizations, from which we randomly selected 2,333 tickets to compose a Gold Standard. Our best results display an average precision and recall of 82.83% and 88.42%, which outperforms the compared sentiment analysis solutions.

CCS Concepts

•Information systems → Dictionaries; Information extraction; Sentiment analysis; •Applied computing → *Annotation;* Language translation;

Keywords

Opinion Mining; IT Tickets; Domain Dictionary

1. INTRODUCTION

Information Technology (IT) is the broad subject concerned with different aspects of managing and processing information in organizations (e.g. computers, software, networks). The ability to access software through personal computers is crucial to all levels of organizations, from the support to daily tasks at the operational level, to the access to critical data for strategic decision making. Users needs range from simple access to email, to the interaction with complex and customized software that requires the proper infrastructure. Organizations either have their own IT department, or delegate this responsibility to external companies.

We shall refer to any documented user request directed to the IT team as a *ticket*. Tickets contain a textual description of the issue/request, possibly with additional structured information such as date, priority, impact, system, etc. IT management frameworks consider objective criteria for prioritizing and addressing tickets, as well as for measuring the quality of the service provided. For instance, ITIL recommends prioritizing tickets based on the importance and impact, whereas delivery time is the main service level agreement in COBIT [23]. Nevertheless, when users report a problem or make a request, they may express sentiments that provide additional contextual information. Sentiments can reveal the user's emotional state with regard to the request (*"I urgently need 6 laptops for next Friday!!!!!"*), or provide early feedback about the quality of services provided by the IT team (*"I have no access to email again. It is the third time this week!"*). Thus, it creates an invaluable opportunity for early analysis of customers' satisfaction.

Sentiment analysis involves the automatic identification of opinions, feelings, evaluations, attitudes and emotions expressed by people in the written language [16]. Lexicon-based approaches are very popular [27], and require a dictionary relating terms to some measurement of the sentiment. Most work in sentiment analysis is focused on identifying opinions, a sentiment measured in terms of a polarity that ranges from negative to positive. Classical applications are the identification of opinions in reviews about products/services [16], brand-management (e.g. [2, 9, 17]), and sentiment-based prediction (e.g. [5, 28]).

The importance of sentiment analysis has also been recognized in the software engineering field. The sentiment of developers was tracked in software development tickets [14, 19, 20], comments associated to commits [10, 21] and project reports [11]. The average time to solve issues has been related to the sentiment associated to issue description [20]. The automatic extraction of emotions from reviews is proposed as a means to evaluate software usability [7]. Most works employ generic sentiment analysis solutions, which yield different and conflicting results [13], possibly by not considering how sentiment is conveyed in software artifacts.

The main challenge on analyzing sentiments in IT tickets is separating the object of the request, which involves reporting some issue, from the sentiment embedded in the text. Technical jargon used to describe an issue (e.g. *"prob-*

MSR'16, May 14-15, 2016, Austin, TX, USA

DOI: http://dx.doi.org/10.1145/2901739.2901781

lem", *"defect"*) is intrinsically negative, but does not necessarily convey any particular subjective feeling. Sentiment is conveyed by greetings (*"good morning"*), closing remarks (*"Thank you"*, *"I hope to hearing from you ASAP"*), expressions that denote intensity (*"again"*, *"desperately"*), to mention a few. A similar challenge has been identified in news, when distinguishing bad news from negative opinions [2].

For instance, consider the ticket *"Good morning, kindly check why my computer is slow. Thanks"*. Despite the user reports a performance problem, the expressions *"good morning"*, *"kindly"* and *"thanks"* demonstrate positive sentiments of politeness. The same issue could be reported as *"Check why my computer is slow"* (neutral), or *"Once more I ask you to check why my computer is slow"* (negative).

In this paper, we propose a method to evaluate the sentiment contained in IT tickets, which are classified according to polarity. Our approach includes: a) the automatic creation of a *Domain Dictionary* that contains words and phrases with sentiment in the IT domain, and b) a *Sentiment Analysis* process that assigns polarity scores to tickets words and phrases, and aggregates individual scores for calculating the overall ticket polarity. The Domain Dictionary aims at identifying terms that convey sentiment in tickets, and it is automatically created by expanding a set of input seed words. We propose three methods for calculating the polarity scores of words and expressions contained in tickets: dictionary-based, structure-based and hybrid. Our study was developed using 34,895 tickets from five different organizations, which describe all sorts of requests (defects, incidents, requests, feedback, etc.). To to evaluate the performance of the proposed solution, we created an annotated Gold Standard composed of 2,333 tickets. Our best results display an average precision and recall of 82.83% and 88.42%, respectively, which largely outperforms the compared sentiment analysis solutions.

With regard to previous work, we developed a sentiment analysis method targeted at IT tickets, based on the premise that it is necessary to distinguish between the objective report of a problem, and the embedded sentiment. By design, generic sentiment analysis solutions do not care for such distinction, and thus do not yield consistent results [13]. The proposed solution derives from a better understanding of how sentiment is expressed as part of the communication between the IT team and users. Previous work focused on texts written by people with technical skills (e.g. developers [14, 19, 20], testers [7]), and we complement these attempts by tackling how sentiment is expressed by non-technical user in the broader area of IT. We considered the characteristics of tickets from five different organizations, resulting in a considerable generalization of our findings.

The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 describes the IT corpus, and Section 4 provides an overview of the approach. The methods proposed for creating a Domain Dictionary and sentiment analysis are detailed in sections 5 and 6, respectively. Our experiments are presented in Section 7. Conclusions and future work are discussed in Section 8.

2. RELATED WORK

Sentiment analysis aims at identifying subjective content in texts, classifying sentiment according to some scale, and summarizing the sentiment contained in a set of documents [27]. A popular form of measuring sentiment is polarity (or valence), ranging from positive to negative [16]. Sentiment analysis can be developed at three levels: document, sentence and aspect [16]. The former aims at defining an overall sentiment for the whole document.

The most popular approaches for sentiment classification are sentiment lexicons and supervised machine learning [27]. The former requires lexicons (or dictionaries) in which terms are associated with a sentiment label (e.g. negative) or score denoting intensity (e.g. -0.7). Machine learning involves training a corpus annotated for sentiment using classification/regression algorithms (e.g. SVM). The dictionary-based approach is widely adopted due to the overhead of creating a quality training corpus.

In general, resources are scarce for languages other than English. For the Portuguese language, addressed in this work, the best sentiment lexicon and natural language processing (NLP) tools are, respectively, SentiLex-PT [24], and Palavras[1], which do not perform well for Brazilian Portuguese [28]. Different studies have demonstrated that automatic translation is a mature approach for sentiment analysis. An extensive experimental study [3] involving automatic translations to three different languages, revealed that the quality of sentiment analysis does not correlate to the quality of translations, measured in terms of BLEU score. The baseline was a manually translated text, and the study compared four different automatic translators. The best results were obtained for the Spanish language, which is close to Portuguese. Automatic translation was successfully applied to the automatic generation of sentiment dictionaries for multiple languages [18, 25]. A triangulation approach [25] creates a parallel corpus using automatic translation, from which sentiment words that have a corresponding meaning are extracted to compose a dictionary in a third language. The approach was validated using several languages, where Spanish and Italian yielded the best results.

The are several generic sentiment lexicons (e.g. SentiWordNet [1], WordNet-Affect[2]) and lexicon-based sentiment solutions (e.g. SentiStrenght [26]), but related work has stressed that the expression of sentiment is dependent on the domain. Semi-automatic techniques for the automatic creation of a customized lexicon from domain-documents were proposed, constructed through an expansion process based on semantic (e.g. synonyms) and/or linguistic relationships, or through distribution probabilities [27]. Results are improved by adopting only a domain dictionary [9, 12, 22], or by methods that combine generic and domain resources [4].

Different works reveal the potential benefits for the software engineering field. Most of them explore off-the-shelf tools for sentiment analysis, particularly the state-of-the-art SentiStrength [26]. A model to evaluate software usability by extracting automatically emotions from usability reviews was proposed in [7], based on SentiStrength. Several works explore artifacts produced by software developers in the context of open-source projects. Using Parrot's framework of basic emotions, Murgia et al. [19] concluded that tickets do express emotions towards design choices, maintenance activity or colleagues, but some emotions are more meaningful in the domain (e.g. joy, sadness). Ortu et al. [20] concluded that some affective states expressed in tickets (e.g. extreme un/politeness, joy) partially explain the mean time to solve

[1]http://beta.visl.sdu.dk/

[2]http://wndomains.fbk.eu/wnaffect.html

Table 1: Tickets Source Description

Business	User	IT Support Coverage	Tickets
Telecom	Internal	Support for infrastructure many internal and acquired systems	29,748
SW House	Internal	Administrative tasks and support for in-house developed software	2,898
SW House	External	Support for a software developed for many customers	1,745
Insurance	Internal	Support for an in-house developed software	362
Research	External	Suggestions for provided infrastructure and IT area	142

issues. They experimented with different forms of sentiment, such as polarity using SentiStrength, basic emotions using machine learning over an annotated corpus [19], and other affective states using off-the-shelf tools. Gusmaz et al. [10] also adopted SentiStrength to analyze emotions expressed by developers in source code repositories using commit comments, and concluded that is important to not consider only the average of the scores to assign the sentiment of a whole commit. Guzman and Bruegge [11] proposed an approach for automatically extracting and summarizing emotions expressed in collaboration artifacts by combining probabilistic topic modeling with SentiStrength for sentiment analysis, as a means to create motional awareness in large or distributed teams.. All these works consider tickets or comments issued by developers, which share a common technical language, and interest in the development of a software product.

A study compares the differences according to the sentiment analysis solution adopted, and highlights severe discrepancies with regard to a manually annotated corpus [13], revealing the importance of methods targeted at the way sentiment is expressed in software artifacts. We developed a solution that addresses the challenges inherent to IT tickets as a means to produce more accurate results. We also contribute to the field by addressing IT tickets, which have a broader coverage of topics compared to software tickets, and which are issued by non-technical people.

3. IT CORPUS

We collected 34,895 tickets from five different companies, distributed in different business areas, as detailed in Table 1. These tickets were extracted from the respective specific support system, are reported by users internal or external to the company, and their content vary according to the coverage of services provided by the IT team. Based on a preliminary analysis, we grouped tickets into three categories, illustrated in Table 2. Issue tickets report infrastructure problems or software errors. Request tickets involve demands concerning data extraction, system maintenance, infrastructure needs or administrative tasks. Feedback tickets are directed to IT team or department, as a means to suggest improvements. We deal with all these ticket types uniformly. All tickets are written in Brazilian Portuguese.

Tickets have a similar structure when compared to emails: a) greetings, where the users write salutations; b) report, where the user requests something or reports an issue; and c) closure, where the user writes a farewell message or expresses some sense of anxiety. There are some expressions that characterize the greetings (e.g. *"dear"*, *"hello"*, *"good afternoon"*) and closure segments (e.g. *"regards"*, *"best wishes"*, *"looking forward"*). The report part is the most important

Table 2: Types of Tickets and Examples

Type	Subtype	Example
Issue	Infrastructure	*"I'm having trouble sending external emails, I urgently need help, because I need to send commercial proposals."*
	Software	*"Good morning, kindly check why when I open the certificate it is not stating the covered contract"*
Request	Data Extraction	*"Please send us the list of cancelled contributions."*
	System Maintenance	*"We need to create a report of payment for a given period. Today we looked one by one to generate it manually."*
	Infrastructure	*"Install CRM on my computer."*
	Administrative	*"Register John's work hours immediately."*
Feedback	IT Area	*"Excellent support of the team involved with the environment maintenance."*
	Software	*"What about automate importing data from the old client system?"*
	Infrastructure	*"Enabling the submission of more jobs without compromising the cluster can help people not delay in the pool."*

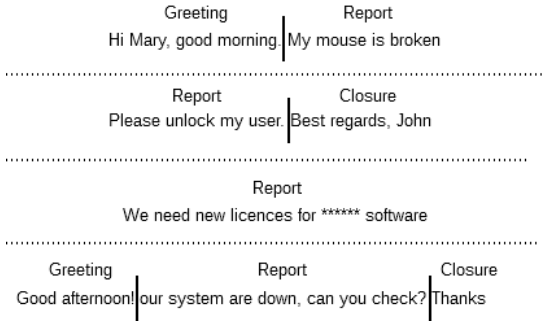

Figure 1: Examples of tickets and their structure

one, and it is always present in the text, while the other ones are optional. Figure 1 illustrates tickets based on the four possible combinations of these segments.

Initially, we performed a superfial analysis in a sample of our corpus to identify whether sentiments are also expressed in IT tickets. Despite most of tickets were neutral, we did observe subtle or explicit forms of sentiment expression according to terms employed. Greetings and closure expressions are weaker forms of sentiment, while emotional expressions inside the report segment are stronger indicators of affective states. For example, in the text *"Unlock my user. Thanks, John"*, the closing word *"thanks"* is subtle, but brings a positive sentiment to the ticket. In *"We have a serious problem in suppliers module, we cannot make payments"*, the word *"serious"* has a negative connotation. It is possible to have both positive and negative terms, and in certain cases it is difficult to determine the ticket polarity. In *"The system has a beautiful layout, but the menu is horrible"*, it is not possible to detect which feeling is dominant due to similar intensity, and therefore, we consider it neutral. On the other hand, in *"Dear John, we have a dangerous security problem in our data storage"*, although *"dear"* is positive, *"dangerous"* is a stronger negative term, so the ticket denotes a negative sentiment. This intuition was confirmed when 3 annotators manually developed a Gold Standard, as detailed in Section 7.1.

4. STUDY OVERVIEW

As mentioned, the main challenge in analyzing sentiments contained in IT tickets is to automatically detect possible sentiments within the objective facts about the issue being reported by the user. When a user issues a ticket, most of the time he/she is reporting something intrinsically negative: a defect, an error, an incident, a request for help. However, it does not mean that he/she is expressing negative feelings about it. We devised an approach that can automatically identify and separate these two pieces of information. The proposed approach is illustrated in Figure 2, and it includes two sub-processes: the automatic creation of a customized *Domain Dictionary*, and the *Sentiment Analysis* process.

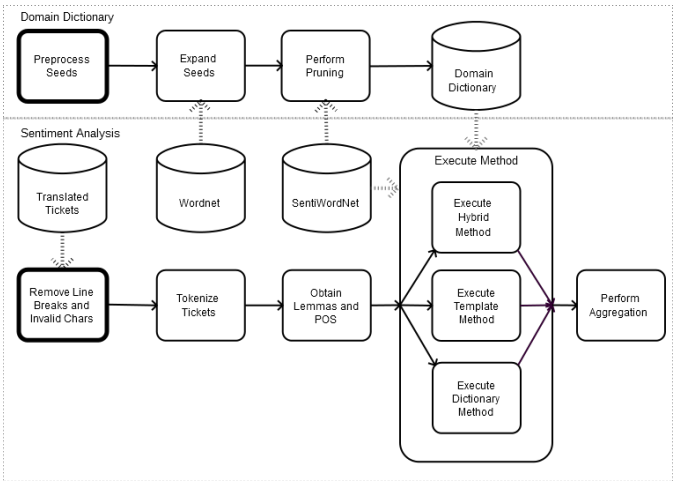

Figure 2: Approach Overview

We decided to use a dictionary-based approach for sentiment mining, since the adoption of machine learning requires a considerable annotated corpus. However all tickets were written in Brazilian Portuguese, for which NLP and sentiment resources are scarce. We overcame this problem by automatically translating our corpus, given that translation systems have reached a good level of maturity for sentiment analysis purposes [3, 18, 25].

The Domain Dictionary aims at supporting the distinction between objective and sentiment words in the IT domain. Basically, it contains only words and expressions that embed some sentiment in the domain, such that it can be used as a prior filter for determining their sentiment. It is thus complementary with regard to generic sentiment lexicons. The customized Domain Dictionary is created by the automatic expansion and pruning of seeds, using a thesaurus and a sentiment lexicon. We chose the seeds using a list of salutations/closing expressions, and a set of manually inspected sentiment words extracted from the most frequent ones found in the Gold Standard. We adopted two popular dictionaries: SentiWordNet and the thesaurus WordNet [8], although other dictionaries could be used. Further details about the creation of this dictionary are provided in Section 5. In the remaining, we will adopt the term *token* to refer indistinctly to words and expressions.

Sentiment analysis is performed according to three steps. First, tickets are *pre-processed* using NLP tools that extract and prepare textual information for sentiment analysis (e.g. tokenization, POS-tagging, etc). We translated all tickets from our corpus to English using the state of the art Google Translate[3]. In the second step, all the tokens contained in the pre-processed tickets will be assigned a polarity score, where the Domain Dictionary is used to filter out candidate sentiment expressions. We developed and tested three algorithms for assigning polarity to words/phrases: Dictionary Method (DM), Template method (TM) and Hybrid Method (HM), detailed in Section 6.2. DM is a purely lexicon-based method, whereas the others consider in addition structural information. All of them perform sentiment analysis at document level, because the ticket is evaluated as a whole and commonly targets a single entity. Hence, the final step aggregates these scores using summation. Figure 4 illustrates the main steps of the process, highlighting the different scores according to each proposed method. The sentiment analysis process is detailed in Section 6.

Section 7 shows that the proposed solution outperforms two popular off-the-shelf sentiment analysis solutions, including SentiStrength, and compares the performance of DM, TM and HM methods.

5. DOMAIN DICTIONARY CREATION

Initial experiments on the use of the general purpose SentiWordNet enabled us to detect the following issues:

- Despite the lexicon associates a polarity to certain terms or expressions, they are neutral in the IT domain. For instance, they correspond to neutral jargon used in the area (e.g. *"save"*, *"folder"*, *"process"*), or terms underlying the description of issues or requests in a ticket (e.g. *"error"*, *"problem"*, *"defect"*);
- Many terms used to express sentiment in this context are not present in the dictionary (e.g. *"urgently"*, *"prioritize"*, *"extremely slow"*);
- The polarity of some terms/expressions is opposite to the one recorded in the lexicon, when considering their usage in the IT domain for reporting issues. Examples as *"standby"* (positive in SentiWordNet, but negative in IT domain), *"important"* (positive in SentiWordNet, but negative in IT domain).

Hence, we propose a process for creating a customized IT sentiment dictionary, which is depicted in the upper box of Figure 2 ("Domain Dictionary"). The process depends on three inputs: a) a set of seeds; b) a sentiment lexicon (SentiWordNet); and c) a thesaurus with synonyms (WordNet).

Similarly to other proposals of domain oriented lexicons [9, 12, 16], the process includes three steps: a) pre-processing the seeds; b) seed expansion and c) pruning. Pre-processing consists of selecting and preparing the seeds for creating the dictionary. The seeds are words or expressions extracted from IT tickets, which are prepared by: a) manually assigning a polarity orientation to the seed according to the domain (e.g. *"horrible"* is negative, and *"excellent"* is positive); b) identifying among the seeds the ones that delimit initial (greeting) and ending (closure) segments of a ticket (e.g. *"dear"* is an initial delimiter, and *"awaiting return"* is an ending delimiter); c) replacing blank spaces by "_" in a whole expression is considered (e.g. *"good morning"* as *"good_morning"*); d) classifying each term candidate for expansion according to one of the following POS (Part of Speech) tags: *"a"* (adjective), *"n"* (noun), *"r"* (adverb) or

[3]https://translate.google.com/

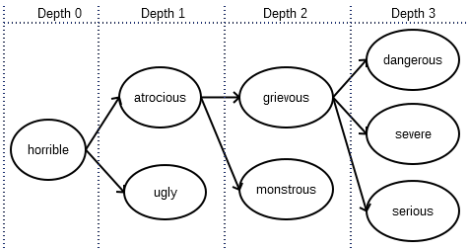

Figure 3: Depths for seed expansion

"v" (verb), and appending it to the token using the special mark *"#"* (e.g. *"excellent#a"*, *"extremely#r slow#a"* and *"good_afternoon#n"*). Each entry in the Domain Dictionary is uniquely identified by <token>#<POS>.

POS tags are required for the expansion process, and well as for the search in both Domain and SentiWordNet dictionaries. Our sentiment analysis methods explore the assigned polarity orientation, as well as positional classification (i.e. initial/ending delimiters) to dynamically quantify the sentiment of a token extracted from a ticket in terms of a polarity score, as it will be discussed in Section 6.

The expansion of domain-related vocabulary from seeds is a common approach for the creation of customized sentiment lexicons, in which Synsets are used to find similar terms recursively, at a maximum depth [27]. However, our seeds contained three patterns to be expanded: a) unigrams that could be replaced by synonyms for expressing sentiments (e.g. *"excellent"*); b) n-grams in which the whole is responsible for the polarity (e.g. *"good morning"*, *"looking forward"*); and c) n-grams in which just some terms could be replaced by independent synonyms, forming all possible combinations latter (e.g. *"extremely difficult"*, is similar to *"highly difficult"* or *"extremely hard"*).

Given a maximum depth and a set of preprocessed seeds, the expansion step recursively explores the Synsets to generate more tokens [9, 16]. An example is provided in Figure 3. If depth is defined as 1, only direct synonyms are retrieved (e.g. *"atrocious"*), otherwise longer paths are explored (e.g. *"grievous"* and *"dangerous"* for depth 3). During expansion, a synonym token inherits all polarity and positional information from its corresponding seed. For instance, the seed *"dear"* is classified as positive, and as an initial delimiter, and therefore its synonym *"dearest"* is similarly classified. Only tokens with the same POS of the seed are considered during the expansion. We used JWI[4] to access WordNet.

Pruning is necessary because WordNet is neither a dictionary appropriate for the IT domain, nor a sentiment lexicon one. Therefore, not all tokens returned by the expansion step are relevant for our purposes. For example: when expanding the token *"persist#v"*, a possible synonym is *"run#v"*, which is a common term in IT, with no related sentiment (i.e. neutral). Thus, we retrieve the polarity associated to the expanded token in SentiWordNet, and discard the expanded token if one of the following conditions is verified: a) the token is not present in the sentiment lexicon or has a polarity score equals to 0; b) the expanded token has an inverse polarity compared to the original seed (e.g. *"slowly#r"* is a negative seed, but the synonym *"easy#r"* is positive), which references the issue of opposite polarities we detected in SentiWordNet usage in IT domain, presented in the beginning of this section; or c) the expanded token has a polarity score between -0.1 and 0.1, which excludes terms with very weak sentiment [9].

Using both the Gold Standard and IT corpus, we evaluated the proposed expansion method by experimenting different depths. We used 181 seeds, which were extracted from two sources: a) a list of salutation and closing expressions, and b) manual analysis of the 25% most frequent terms found in the Gold Standard to identify the ones with sentiment. The expansion resulted in 208, 224 and 240 tokens considering depths 1, 2 and 3, respectively. However, the number of tokens matched in the tickets of the Gold Standard did not increase significantly: 668 using only the seeds, compared to 691 in all other depths. We also tested the matches of expanded token in the corpus: the difference in the number of matches comparing depths 1 and 3 was insignificant (only 10 additional matches). Thus, the Domain Dictionary used in the experiments described in Section 7 was created using depth=1.

The three leftmost columns of Table 3 illustrates some entries of the Domain Dictionary. The methods discussed in next section will be used to automatically assign polarity scores according to different strategies. These scores are highlighted in boldface in the table.

6. SENTIMENT ANALYSIS

6.1 Ticket Preprocessing

Pre-processing is the first step of the process (Figure 2), needed to derive for each ticket a list of tokens, with the respective POS and lemma, to be used as key for search in both the Domain Dictionary and SentiWordNet.

After tickets are translated, basic normalization actions are performed, such that the text can be correctly tokenized (e.g. removal of line breaks and special characters). The final text contains only alphanumeric characters, blank spaces and punctuation. Texts were tokenized using the Java API *java.text.BreakIterator*[5]. Blank spaces and punctuation marks are used to split the text into tokens. Sequences of tokens that corresponds to an expression in the Domain Dictionary are combined in a single token (e.g. "*good_morning*").

Then we used TT4J[6], a Java implementation of TreeTagger[7], to obtain the lemma (i.e. canonical form) and the POS for each token, as illustrated in Figure 4. The result is a set of tokens in the format <lemma>#<POS>.

6.2 Methods for Assigning Polarity Scores

6.2.1 Dictionary Method

The DM method, for which the pseudocode is provided in Algorithm 1, is purely based on dictionaries (domain and generic). For each token from a ticket, the method searches whether it exists in the Domain Dictionary, using the token lemma and POS as key for the search (line 6). If found, it is a candidate for sentiment, with a positive or negative polarity; otherwise it is a neutral word in the domain. To assign a score denoting intensity, the algorithm searches SentiWordNet using the same key (line 7): if found, it retrieves the

[4]http://projects.csail.mit.edu/jwi/

[5]http://docs.oracle.com/javase/tutorial/i18n/text/about.html

[6]https://reckart.github.io/tt4j/

[7]http://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger/

Table 3: Domain Dictionary and Illustration of Score Calculations

Domain Dictionary			SentiWordNet	DM Classification		TM Classification		
Token	**Polarity**	**Delimiter**	**Score**	**Mean of Modules**	**Polarity Score**	**Category**	**Mean**	**Polarity Score**
horrible#a	Negative	No	-0.625	0,573	**-0.625**	NR	-0,667	**-0,667**
difficult#a	Negative	No	-0.708		**-0.708**	NR		**-0,667**
frequently#n	Negative	No	null		**-0,573**	NR		**-0,667**
kindly#n	Positive	No	0.5		**0.5**	PR	0.5	**0.5**
good morning#n	Positive	Initial	null		**0,573**	PG	0.459	**0.459**
dear#a	Positive	Initial	0.459		**0.459**	PG		**0.459**

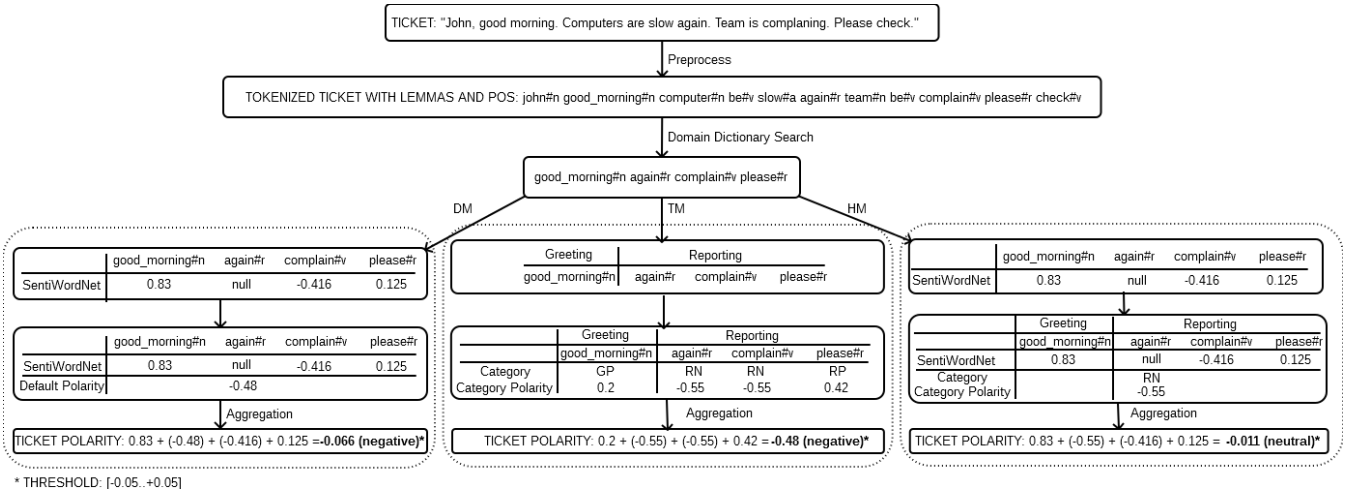

Figure 4: Sentiment Analysis Illustration according to DM, TM and HM Methods

```
1:  polarizedTokensList ← newList()
2:  while tokenizedTicket.hasNext() do
3:   token ← tokenizedTicket.next()
4:   lemma ← token.lemma
5:   polarityScore ← null
6:   if DomainDictionary.contains(lemma,pos) then
7:    if SentiWordNet.contains(lemma,pos) and
      SentiWordNet.getPolarity(lemma,pos) != 0) then
8:     polarityScore ← SentiWordNet.getPolarity(lemma, pos)
9:    else
10:    domainDicToken ←
       DomainDictionary.getToken(lemma, pos)
11:    if domainDicToken.PolarityOrientation == "POSITIVE"
       then
12:     polarityScore ← DefaultPolarity.positive
13:    else
14:     polarityScore ← DefaultPolarity.negative
15:    end if
16:   end if
17:   polarizedTokensList.add(token,polarityScore)
18:  end if
19: end while
20: return polarizedTokensList
```

Algorithm 1: Dictionary Method Pseudocode

corresponding score (line 8). Recall that the pruning step (Section 5) does not include in the Domain Dictionary terms that in SentiWordNet are neutral, denote extremely weak sentiment or that have an inverse polarity compared to the seed. Otherwise (i.e. it is not included in SentiWordNet), a value referred as *default score* is assigned (lines 11-15), a case that occurs only for seed terms, given that in the pruning step, all tokens generated by expansion will have their polarities checked in SentiWordNet before being added to Domain Dictionary (cf. Section 5). Rather than assigning these scores manually to the Domain dictionary, we compared different strategies to calculate the default score based on the polarity of the expanded tokens, which are reported in the experiments are detailed in Section 7.2.

```
1:  polarizedTokensList ← newList()
2:  splitTicket ← Template.splitTicket(tokenizedTicket)
3:  while tokenizedTicket.hasNext() do
4:   token ← tokenizedTicket.next()
5:   lemma ← token.lemma
6:   polarityScore ← null
7:   if DomainDictionary.contains(lemma,pos) then
8:    domainDicToken ←
      DomainDictionary.getToken(lemma, pos)
9:    orientation ← domainDicToken.orientation
10:   position ← splitTicket.getTokenPosition(token)
11:   polarityScore ←
      CategoryPolarity.getPolarityScore(orientation, position)
12:   polarizedTokensList.add(token,polarityScore)
13:  end if
14: end while
15: return polarizedTokensList
```

Algorithm 2: Template Method Pseudocode

6.2.2 *Template Method*

Many tickets follow a template similar to emails (Figure 1). The Template method (TM) explores the structure of the document [15, 20]. The strategy underlying TM is that tokens in the same position in the ticket receive a common score, referred to as *"category score"*. The combination of polarity and position results in 6 categories: *Positive Greeting* (PG), *Negative Greeting* (NG), *Positive Report* (PR), *Negative Report* (NR), *Positive Closure* (PC) and *Negative Closure* (NC). *Greeting* tokens are present in the initial segment of the ticket, *closure* ones in the ending segment, and *report* tokens are located in between. Algorithm 2 presents the pseudocode for the TM method.

Recall that some of the seeds used for creating the Domain

```
1:  polarizedTokensList ← newList()
2:  polarityScore ← null
3:  while tokenizedTicket.hasNext() do
4:   token ← tokenizedTicket.next()
5:   lemma ← token.lemma
6:   polarityScore ← null
7:   if DomainDictionary.contains(lemma,pos) then
8:    if SentiWordNet.contains(lemma,pos) and
      SentiWordNet.getPolarity(lemma,pos) != 0) then
9:     polarityScore ← SentiWordNet.getPolarity(lemma, pos)
10:   else
11:    domainDicToken ←
       DomainDictionary.getToken(lemma, pos)
12:    orientation ← domainDicToken.orientation
13:    position ← splitTicket.getTokenPosition(token)
14:    polarityScore ←
       CategoryPolarity.getPolarityScore(orientation, position)
15:   end if
16:   polarizedTokensList.add(token,polarityScore)
17:  end if
18: end while
19: return polarizedTokensList
```

Algorithm 3: Hybrid Method Pseudocode

Dictionary are classified as initial or ending delimiters, in addition to positive/negative, an information that is inherited by all terms resulting from the expansion step. So, the first step in TM is to analyze this structure (line 2): first, all initial and ending tokens are located in list of tokens, where the last initial token delimits its greeting segment, and the first ending token delimits its closure segment; the remaining text between these delimiters is regarded as included in the reporting segment. Then the method iterates over the tokenized list received as input, and searches for each token in Domain Dictionary, based on the respective lemma and POS (line 7). For every token found, it identifies the respective polarity orientation in the ticket (line 9) and its position within the ticket (line 10), and assigns the corresponding category score (line 11); otherwise, the token is regarded as neutral. We tested different ways to automatically calculate category scores, which are described in Section 7.2.

6.2.3 Hybrid Method

The Hybrid method (HM) combines properties of the two previous methods. As in DM, the Domain Dictionary is used to detect a candidate sentiment word (line 7), and SentiWordNet is used to assign a positive/negative score (line 9). If the token is not found in SentiWordNet, then the token is classified according to its position and polarity orientation, according to one of the 6 category scores, as in TM (lines 11 to 14). Algorithm 3 shows the pseudocode of HM.

6.3 Exclamation Marks

Exclamation marks are indicators of intensity for both positive/negative sentiments [17]. As a common step to all three aforementioned methods, whenever a token is as followed by one exclamation mark, it is added twice to the list of polarized tokens. If it followed by two or more exclamation mark (e.g. !!, !!!!!!!!!), the token is added three times.

6.4 Aggregation for Ticket Polarity Score

The final step aggregates individual polarity scores using summation [27]. Given positive/negative thresholds, if the result is less than the negative threshold, the ticket is negative; and if greater than a positive threshold, positive. Otherwise it is neutral, meaning that either it contains only neutral words with regard to the IT domain, or contradictory terms that annul themselves in such a way it it is not possible to detect a tendency. In the example of Figure 4, given a threshold of -0.05 and +0.05, the ticket is negative according to DM and TM, and neutral according to HM.

Table 4: Illustration of Tickets in the Tutorial

Difficulty	Ticket
Easy	*"Apparently the power problems have been solved and the performance of the cluster is excellent."*
Medium	*"Good morning! After completing the registration form, I'm facing redirection issues. Attached, follows the print screen of error. Thanks."*
Hard	*"Dear, when I try to download the backup, an error occurs. See attached file. Please prioritize."*

Table 5: Gold Standard Inter-Annotator Agreement

Annotator	Positive	Negative	Neutral	Disagreement
A	273	154	1942	B: 12%
				C: 10%
				BC: 7%
B	358	172	1803	A: 12%
				C: 21%
				AC: 7%
C	187	67	2079	A: 10%
				B: 21%
				AB: 7%
Total	**233**	**105**	**1995**	

7. EXPERIMENTS

7.1 Gold Standard

In order to evaluate the results of our techniques for determining ticket polarity, we randomly selected a sample with 2,333 tickets (about 6% of the corpus) to perform manually annotation. We used three distinct annotators: a member of an IT team, majored in computer science (A), a person who works at an IT company, but graduated in another major (B), and a non-technical user (C). They all annotated every sample ticket as positive, negative or neutral, in a process similar to the ones reported in [28, 30]. After merging the results, the final polarity of a ticket was determined with the value that had been set by the majority [17]. When all annotators disagreed, they discussed to reach an agreement.

A tutorial with three steps was developed [30]: instructions reading, visualization of examples and supervised annotation. The instructions to be followed were:

1. There are no fixed rules about how a ticket must be annotated based on its tokens. Some examples will be discussed, but it does not mean that specific tokens determine a polarity;
2. Tickets should be interpreted considering the IT context, and based on what is explicitly written. Annotators should not take texts out of context and assume what they "could" imply;
3. Annotators should be consistent in their own annotations and with the examples given to them for training.

Next, eighteen examples selected from the corpus were discussed, six for each polarity (positive, negative, neutral). These tickets were also classified according to the complexity for determining the polarity: easy, medium and hard. Examples of positive tickets are illustrated in Table 4.

As the final step of the tutorial, annotators performed a supervised annotation, i.e. we annotated some tickets to-

gether with the annotators. To this end, we chose eighteen tickets using the same criteria used for the examples.

After the training, all annotators received the tickets from the sample, and individually annotated the tickets on their own time. The disagreements intra-annotators are displayed in Table 5. The disagreement level is comparable to related work [28, 30]. The final row *Total* shows the final number of positive, negative and neutral tickets after merging all tickets polarities set by the three annotations.

7.2 Experimental Setup

Goals: The goals of the experiments are:
a) to evaluate the value added by a customized solution for sentiment analysis in IT tickets. We compared our method to two popular solutions in the software engineering field, SentiWordNet and SentiStrength (Section 2);

b) to evaluate the role played by ticket structure in the analysis of sentiments. We compared the results produced the dictionary-based method (DM), with the two alternatives that consider the position of tokens in tickets (TM and HM);

c) to identify a suitable strategy to automatically assigning polarity scores to tokens in the IT domain. We experimented different strategies for deriving default scores for seed words in DM, as well as determining category scores when considering tokens position in the ticket (TM and HM).

Metrics: Using the Gold Standard as ground truth, we compared the results using standard metrics: accuracy, precision, recall and F-measure. Accuracy measures the percentage of tickets that are correctly classified. Given a class (e.g. negative), precision is the percentage of predictions for that class that are correct, while recall is the percentage of actual tickets of that class classified as such. F-measure combines both precision and recall. A bi-caudal T-test was applied to compare the methods and their respective implementation variations, with a significance level of 0.05.

Baselines: Baseline results were produced using a DM variation that uses only SentiWordNet, and the SentiStrength tool available for academic purposes. We adopted 0 as threshold for ticket polarity, as it produced the best overall results. The SentiWordNet implementation does not employ the Domain Dictionary, i.e. all tokens are directly searched at SentiWordNet, from where the scores are extracted, if available. The tokenization, handling of exclamation marks and aggregation are performed as described in Section 6. SentiStrength was executed with the default configuration, which returns both a negative and a positive emotional score for each text. We considered the overall ticket polarity as the difference between the positive and negative scores.

Default/Category Score Strategies: due to the difficulty of assigning and curating manually scores to IT tokens, we experimented different ways of using the values available in SentiWordNet to calculate automatically default and category scores. The seeds and expanded tokens thus guide how to compute these scores. We combined two strategies referred to as *formula* and *mirroring*. In the former, we experimented the mean and the median of a set of scores extracted from SentiWordNet. Mirroring does not take to account the polarity of words, just the module of the scores from SentiWordNet, allowing that positive and negative tokens annul each other in neutral tickets.

Implementations of the Proposed Methods: by combining the formula and mirroring strategies, four variations of each method (DM, TM and HM) were implemented. These implementations consist in different manners to dynamically quantify the sentiment of a token extracted from a ticket in terms of a polarity score, by combining information contained in the Domain Dictionary and SentiWordNet. Each implementation covers all steps described in 6.

In DM, the default score is the mean or median of scores from SentiWordNet (when available) of the tokens present in the Domain Dictionary. Without mirroring, scores of positive and negative tokens are calculated independently, and thus there is a different default score for each polarity. With mirroring, such a difference is disregarded. i.e. the module of all scores from SentiWordNet are applied to compute the mean/median, which is then converted to a positive or negative number accordingly. Table 3 displays an example of DM using mean and mirroring.

In TM and HM, the score of each one of the six structural categories (Section 6.2.2) is also automatically calculated considering the scores of tokens in SentiWordNet that are present in the Domain Dictionary. Tokens in Domain Dictionary classified as initial delimiter and as ending delimiter provide scores for the greeting (PG, NG) and closure categories (PC,NC), respectively. The remaining tokens are used for computing scores for the report categories (PR, NR). If mirroring is not adopted, each category is calculated from the mean/median of tokens of a specific polarity. When using mirroring, such distinction is not done. Table 3 illustrates TM using mean without mirroring.

7.3 Results and Discussions

Baseline Comparison: Table 6 displays the metrics for the baseline methods, as well as for all four variations of DM, TM and HM. The best results are highlighted in boldface. All metrics were generated considering 0 as threshold for distinguishing neutral tickets from positive/negative ones.

It is possible to observe that any implementation of the proposed methods greatly outperforms the baselines. There is a single exception, which is the precision of neutral tickets. SentiWordNet achieved a precision of 100% for neutral tickets, but at the expense of a very low recall (16%), meaning that when a ticket is classified as neutral, it is always correct (precision), but very few neutral tickets are recognized as such (recall). As a consequence, the corresponding F-measure is also low (27%). The precision achieved by any implementation of our methods for neutral tickets is comparable to SentiWordNet (98-99%), but they display an excellent recall (96%). Thus, our solution always presents a higher F-measure (97%). Considering F-measure, which weights recall and precision, SentiStrength delivered the best result, when compared to SentiWordNet.

The weak point of all baseline solutions is the classification of negative tickets, presenting inferior performance for all three metrics. Precision is almost equal in SentiWordNet and SentiStrength (7% and 8%, respectively), with a corresponding impact on the corresponding F-measure. This result shows that our methods handle efficiently the distinction between the terms used to report an issue, and the ones used to describe sentiment.

The metrics for positive tickets reveal discrepant results for the baselines. SentiWordNet recognizes most positive tickets (91% of recall), but misclassifies negative/neutral

Table 6: Comparison of Baseline and Proposed Methods

	Baseline		DM				TM				HM			
			Without		Mirroring		Without		Mirroring		Without		Mirroring	
	Senti WordNet	Senti Strength	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median
Accuracy	25%	71%	**95%**	**95%**	**95%**	**95%**	**95%**	**95%**	**95%**	**95%**	**95%**	**95%**	**95%**	**95%**
Positive Precision	17%	53%	84%	83%	83%	84%	83%	80%	82%	83%	84%	84%	**85%**	84%
Negative Precision	07%	08%	67%	67%	68%	68%	62%	67%	**69%**	64%	60%	65%	66%	61%
Neutral Precision	**100%**	94%	99%	99%	98%	98%	99%	99%	99%	98%	99%	99%	99%	99%
Positive Recall	91%	78%	95%	**97%**	94%	94%	91%	**97%**	**97%**	94%	90%	94%	94%	90%
Negative Recall	50%	37%	78%	76%	70%	71%	78%	69%	70%	70%	**80%**	78%	78%	78%
Neutral Recall	16%	71%	**96%**	**96%**	**96%**	**96%**	**96%**	**96%**	**96%**	**96%**	**96%**	**96%**	**96%**	**96%**
Positive F-measure	29%	63%	**89%**	**89%**	88%	88%	87%	87%	**89%**	**89%**	87%	**89%**	**89%**	87%
Negative F-measure	12%	14%	**72%**	71%	69%	70%	69%	68%	70%	67%	69%	71%	71%	69%
Neutral F-measure	27%	81%	**97%**	**97%**	**97%**	**97%**	**97%**	**97%**	**97%**	**97%**	**97%**	**97%**	**97%**	**97%**

tickets as positive (17% in precision). In comparison, SentiStrength recognizes less positive tickets (78% of recall), but more tickets are correctly classified as positive (53% of precision). Our F-measure for positive tickets ranges from 87% to 89%, outperforming the baseline solutions in both precision and recall.

Methods and Strategies Comparison: When considering only the implementations of DM, TM and HM, we observe in Table 6 that results are very close. The same accuracy is achieved in all methods and their variations: 95%. The same happens with metrics for neutral tickets, where recall and F-measure are equal (96% and 97%), and the difference in precision is irrelevant (98-99%). The results for the other classes are also very good, showing a very small difference (e.g. 69-72% of F-measure for the negative class, and 87-89% of F-measure for the positive one).

Therefore, we performed two additional analysis: a) comparison of the three methods using the average performance of their variations, in order to evaluate the contribution of ticket structure analysis over a simple dictionary-based approach; b) comparison of the default/category score calculation method according to each strategy, namely formula and mirroring. We performed an analysis of the differences using a bi-caudal T-test, with significance level of 0.05 using the Weka Experimenter environment[8].

The results of the methods' comparison are displayed in Table 7, where the average value for each metric with the respective standard deviation is shown. Using Weka's convention, the symbol * denotes a score that is statistically inferior when compared to other values. In addition to the results of Precision, Recall and F-measure per class, we added a macro-averaged row for each of these metrics.

Results are very similar, with no statistical difference in most cases. TM results have proved to be, in average, inferior to the ones of both DM and HM in most cases. Compared to DM, these differences are statistically significant only in two cases: averaged F-measure, and F-measure for the positive class. This means that the weighted combination of precision/recall per class proves the inferiority of the results. Average results for the HM methods are slightly inferior when compared to DM, but this difference is not statistically significant. Hence, the influence of ticket structure on determining the sentiment could not be confirmed.

We then compared all the implementations according to the formula strategy (mean and average). Regardless the method or adoption of mirroring, results were equal or similar, and no statistical difference was observed. This means

[8]http://www.cs.waikato.ac.nz/ml/weka/

Table 7: Comparison of DM, TM, and HM Methods

Metric	DM	TM	HM
Positive Precision	83.00(0.82)	82.50(1.73)	83.75(0.50)
Negative Precision	66.75(2.06)	66.25(2.87)	65.00(3.56)
Neutral Precision	98.75(0.50)	98.50(0.58)	98.50(0.58)
Positive Recall	95.75(1.50)	94.00(2.45)	92.75(1.89)
Negative Recall	73.50(4.12)	72.00(4.08)	75.00(4.69)
Neutral Recall	96.00(0.00)	96.00(0.00)	96.00(0.00)
Positive F-measure	89.00(0.00)	87.50(0.58)*	88.00(0.82)
Negative F-measure	70.00(2.16)	69.00(0.82)	69.75(0.96)
Neutral F-measure	97.00(0.00)	97.00(0.00	97.00(0.00)
Average Precision	82.83(13.70)	82.42(13.87)	82.42(14.44)
Average Recall	88.42(11.25)	87.33(11.63)	87.92(10.00)
Average F-measure	85.33(11.88)	84.50(12.15)*	84.92(11.86)

Table 8: Comparison of the Effects of Mirroring

Metric	No Mirroring	Mirroring
Positive Precision	83.00(1.55)	83.17(0.75)
Negative Precision	64.67(3.01)	67.33(1.75)
Neutral Precision	99.00(0.00)	98.17(0.41)*
Positive Recall	94.00(2.97)	94.33(1.37)
Negative Recall	76.50(3.89)	70.50(0.55)*
Neutral Recall	96.00(0.00)	96.00(0.00)
Positive F-measure	88.00(1.10)	88.33(0.52)
Negative F-measure	70.00(1.55)	69.17(1.17)*
Neutral F-measure	97.00(0.00)	97.00(0.00)
Overall Precision	88.83(9.39)	86.94(12.01)
Overall Recall	88.83(9.39)	86.94(12.01)
Overall F-measure	85.00(11.60)	84.83(11.99)

that both median and mean are suitable for calculating default/category scores. Due to space limitations, these results are not shown in this paper.

Finally, Table 8 compares the performance of the implementations without mirroring, with the one using mirroring, regardless the method and formula strategy. Results are slightly inferior in most cases when Mirroring is adopted, but statistically significant only in three cases: recall and F-measure for negative tickets, and precision for neutral tickets. Thus, the positive/negative influence of mirroring on the automatic calculation of default/category scores could not be confirmed.

Influence of Translation: to increase the confidence on our premise that automatical translation is mature for sentiment analysis [3, 18, 25], we decided to compare the effects of automatic and manual translations, considering a small set of tickets. We randomly selected thirty (30) tickets, distributed in equal number per polarity. Then, we compared the results produced considering the manual and automatic translations of a same ticket. Figure 5 displays the differences in the quantitative sentiment scores using a boxplot.

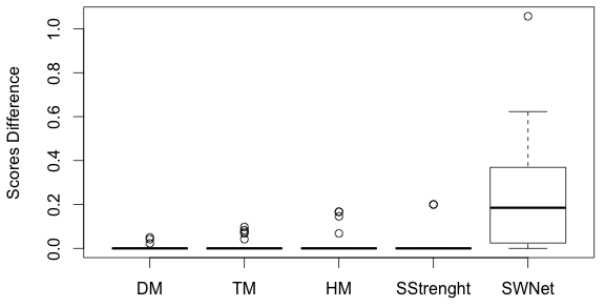

Figure 5: Manual vs. Automatic Translations: Differences on Sentiment Scores

We adopted the implementation based on median and without mirroring for DM, TM and HM.

There was a single case of polarity shift, observed for SentiWordNet (represented as the outlier). In all other cases, we observed only different sentiment scores within the same polarity. This means that the automatic translation tends to preserve the valence of terms, eventually with a different intensity. In particular, we observed less negative terms, and more positive ones. Our methods are less sensitive to such a change, because they calculate the score as an aggregated value (median or average). HM was the method that presented most differences (5 tickets), and DM, the least (2). Results for SentiStrenght were normalized, because it measures sentiment in a scale of [-5..5]. We observed only two differences in the score. The most sensitive method to translation was SentiWordNet (50% of the tickets). The range of differences is wider for SentiWordNet, unlike the other methods, where they are perceived as outliers.

Discussion: As shown in Table 5, only 105 (4.5%) tickets from the Gold Standard were annotated as negative, so any error related to negative classes has a bigger effect on the metrics, compared to the ones referring to the positive and neutral tickets. This partially explains why all results for negative tickets are inferior when compared to the ones related to positive/neutral tickets. Likewise, neutral tickets stand for 85.5% of the Gold Standard, and therefore, errors are more subtly represented into the respective metrics. This also affected the statistical power of the tests performed.

A superficial analysis revealed that the main reason for misclassification is the absence of some sentiment expressions from the Domain Dictionary. For example, *"The mobile system is not working. What a struggle!!!"* was misclassified as neutral because *"struggle"* was not frequent enough in the corpus to be not included in the Domain Dictionary.

A possible solution to the aforementioned issue would be to tune the Domain Dictionary to include more seeds. However, this solution needs to be carefully evaluated, as it may affect the interpretation of neutral tickets. For example, the word *"already"* is used to express both objetive and subjective ideas. In the text *"I already asked you to fix it several times..."* it is certainly negative, but in the ticket *"We already bought the licenses, please proceed with software installation"* it is neutral.

8. CONCLUSIONS AND FUTURE WORK

In this paper we described and evaluated an approach for sentiment analysis in IT tickets. It relies on a Domain Dictionary to filter out candidate sentiment terms in the IT domain, and a sentiment analysis process to assign polarity scores based on a sentiment lexicon and/or ticket structure. The three proposed methods largely outperform SentiStrength, a popular sentiment analysis tool in the software engineering field, revealing the value of solutions that take into account the way sentiment is conveyed in software artifacts. We applied our methods to the whole IT corpus, and found 5% of negative tickets, and 10% of positive tickets, a proportion that is very similar to distribution found in our Gold Standard.

Despite the creation of the Domain Dictionary requires a set of seeds, these were selected and prepared through a fairly easy process. We adopted existing lists of greetings and closing expressions, as well as frequent tokens extracted from the Gold Standard. The polarity scores were automatically assigned through different strategies, based on the seeds and expanded tokens. The resulting Domain Dictionary is thus suitable to IT tickets in general, but the process could be replicated to other artifacts with similar characteristics without great effort.

The devised methods presented similar performance, with a small advantage for DM and HM. The power of the statistical tests did not enable us to determine the role played by ticket structure in sentiment identification, nor whether there are significant differences among the devised strategies for automatically deriving default/category scores, although the performance of mirroring was slightly worse. Overall, we reached very good results in classifying the polarity of tickets, particularly for neutral and positive tickets, but the identification of negative tickets requires improvements.

The proposed solution employs resources targeted at the English language, and it was applied to our corpus in Portuguese with the aid of automatic translation. Thus, in addition to English, it suits any language for which mature automatic translators are available. The analysis of a small sample revealed that our methods were the least sensitive to translation issues.

Future work includes improving DM, TM and HM algorithms in different directions. In addition to devising new ways of exploring the structure of tickets, we will experiment other strategies for the definition of the default and category scores. The limitations of SentiWordNet with regard to the IT domain could be overcome with different alternatives, such as combination of generic sentiment solutions (e.g. SentiStrenght), or assignment of scores directly in the Domain Dictionary, for instance, through crowd-sourcing.

The Gold Standard was randomly created, and therefore the largest corpora are more represented. Nevertheless, these are tickets that address the widest class of issues, systems and departments. In the future, we will study how to compose better samples to generalize our findings with regard to all kinds of IT tickets and corporations.

Finally, the consideration of other forms of sentiments could provide new insights to the analysis of sentiments in IT tickets. We are planning to consider emotions (e.g. [6, 29]). However, the development of a Gold Standard through annotation is far more complex for emotions [17], and alternative forms of validation need to be considered.

Acknowledgments

This research is sponsored in part by CNPq (Brazil) under Grant No. 459322/2014-1.

9. REFERENCES

[1] S. Baccianella, A. Esuli, and F. Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proc. of the International Conference on Language Resources and Evaluation (LREC), Valletta, Malta*, 2010.

[2] A. Balahur, R. Steinberger, M. Kabadjov, V. Zavarella, E. Van Der Goot, M. Halkia, B. Pouliquen, and J. Belyaeva. Sentiment analysis in the news. In *Proc. of the International Conference on Language Resources and Evaluation (LREC), 2010, Valletta, Malta*, volume 10, page 2216, 2010.

[3] A. Balahur and M. Turchi. Comparative experiments using supervised learning and machine translation for multilingual sentiment analysis. *Computer Speech & Language*, 28(1):56–75, 2014.

[4] D. Bollegala, D. Weir, and J. Carroll. Cross-domain sentiment classification using a sentiment sensitive thesaurus. *IEEE Transactions on Knowledge and Data Engineering*, 25(8):1719–1731, Aug 2013.

[5] J. Bollen, H. Mao, and X. Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011.

[6] P. Ekman. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200, 1992.

[7] A. M. El-Halees. Software Usability Evaluation Using Opinion Mining. *Journal of Software*, 9(2):343–349, feb 2014.

[8] C. Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[9] N. Godbole, M. Srinivasaiah, and S. Skiena. Large-scale sentiment analysis for news and blogs. In *Proc. of the First International AAAI Conference on Weblogs and Social Media (ICWSM)*, volume 2, 2007.

[10] E. Guzman, D. Azócar, and Y. Li. Sentiment analysis of commit comments in github: an empirical study. In *Proc. of the 11th Working Conference on Mining Software Repositories, MSR 2014, Hyderabad, India*, pages 352–355, 2014.

[11] E. Guzman and B. Bruegge. Towards Emotional Awareness in Software Development Teams. In *Proc. of the 9th Joint Meeting on Foundations of Software Engineering*, pages 671–674, New York, New York, USA, aug 2013. ACM Press.

[12] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proc. of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 168–177. ACM, 2004.

[13] R. Jongeling and A. Serebrenik. Choosing Your Weapons : On Sentiment Analysis Tools for Software Engineering Research. In *Proc. of the IEEE International Conference on Software Maintenance and Evolution*, pages 531–535, Bremen, 2015.

[14] F. Jurado and P. Rodriguez. Sentiment Analysis in monitoring software development processes: An exploratory case study on GitHub's project issues. *Journal of Systems and Software*, 104:82–89, 2015.

[15] N. Kobayashi, K. Inui, and Y. Matsumoto. Extracting aspect-evaluation and aspect-of relations in opinion mining. In *EMNLP-CoNLL 2007, Proc. of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic*, pages 1065–1074, 2007.

[16] B. Liu. *Sentiment analysis and opinion mining*. Morgan & Claypool Publishers, 2012.

[17] S. M. Mohammad. Sentiment analysis: Detecting valence, emotions, and other affectual states from text. In H. Meiselman, editor, *Emotion Measurement*. Elsevier, 2016.

[18] M. D. Molina-Gonzalez, E. Martinez-Camara, M.-T. Martin-Valdivia, and J. M. Perea-Ortega. Semantic orientation for polarity classification in spanish reviews. *Expert Systems with Applications*, 40(18):7250 – 7257, 2013.

[19] A. Murgia, P. Tourani, B. Adams, and M. Ortu. Do developers feel emotions? an exploratory analysis of emotions in software artifacts. In *Proc. of the 11th Working Conference on Mining Software Repositories, MSR 2014, Hyderabad, India*, 2014.

[20] M. Ortu, B. Adams, G. Destefanis, P. Tourani, M. Marchesi, and R. Tonelli. Are Bullies more Productive ? Empirical Study of Affectiveness vs . Issue Fixing Time. In *Proc. of the IEEE/ACM Working Conference on Mining Software Repositories*, Florence, 2015.

[21] D. Pletea, B. Vasilescu, and A. Serebrenik. Security and Emotion: Sentiment Analysis of Security Discussions on GitHub. In *Proc. of the IEEE/ACM Working Conference on Mining Software Repositories*, pages 348–351, 2014.

[22] G. Qiu, B. Liu, J. Bu, and C. Chen. Expanding domain sentiment lexicon through double propagation. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, IJCAI'09, pages 1199–1204, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.

[23] S. Sahibudin, M. Sharifi, and M. Ayat. Combining itil, cobit and iso/iec 27002 in order to design a comprehensive it framework in organizations. In *Proc. of the Second Asia International Conference on Modeling Simulation, 2008*, pages 749–753, May 2008.

[24] M. Silva, P. Carvalho, and L. Sarmento. Building a sentiment lexicon for social judgement mining. *Computational Processing of the Portuguese Language*, pages 218–228, 2012.

[25] J. Steinberger, M. Ebrahim, M. Ehrmann, A. Hurriyetoglu, M. Kabadjov, P. Lenkova, R. Steinberger, H. Tanev, S. VÃązquez, and V. Zavarella. Creating sentiment dictionaries via triangulation. *Decision Support Systems*, 53(4):689 – 694, 2012.

[26] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas. Sentiment in short strength detection informal text. *J. Am. Soc. Inf. Sci. Technol.*, 61(12):2544–2558, Dec. 2010.

[27] M. Tsytsarau and T. Palpanas. Survey on mining subjective data on the web. *Data Mining and Knowledge Discovery*, 24(3):478–514, 2012.

[28] D. Tumitan and K. Becker. Sentiment-based features for predicting election polls: A case study on the brazilian scenario. In *Proc. of the IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014*,

volume 2, pages 126–133, Aug 2014.

[29] A. B. Warriner, V. Kuperman, and M. Brysbaert. Norms of valence, arousal, and dominance for 13,915 English lemmas. *Behavior research methods*, 45(4):1191–207, Dec. 2013.

[30] J. Wiebe, T. Wilson, and C. Cardie. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2-3):165–210, 2005.