Information Systems xxx (xxxx) xxx

Contents lists available at ScienceDirect

Information Systems

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

Speech-acts based analysis for requirements discovery from online discussions

Itzel Morales-Ramirez [a], Fitsum Meshesha Kifetew [b,*], Anna Perini [b]

[a] INFOTEC, Av. San Fernando 37, 14050 Tlalpan, Mexico City, Mexico
[b] Fondazione Bruno Kessler, via Sommarive 18, 38123 Trento, Italy

ARTICLE INFO

Article history:
Received 15 December 2017
Received in revised form 15 June 2018
Accepted 13 August 2018
Available online xxxx

Keywords:
Requirements engineering
Speech-acts analysis
Sentiment analysis
Classification techniques
Online discussions

ABSTRACT

Online discussions about software applications and services that take place on web-based communication platforms represent an invaluable knowledge source for diverse software engineering tasks, including requirements elicitation. The amount of research work on developing effective tool-supported analysis methods is rapidly increasing, as part of the so called software analytics. Textual messages in App store reviews, tweets, online discussions taking place in mailing lists and user forums, are processed by combining natural language techniques to filter out irrelevant data; text mining and machine learning algorithms to classify messages into different categories, such as bug report and feature request.

Our research objective is to exploit a linguistic technique based on *speech-acts* for the analysis of online discussions with the ultimate goal of discovering requirements-relevant information. In this paper, we present a revised and extended version of the *speech-acts* based analysis technique, which we previously presented at CAiSE 2017, together with a detailed experimental characterisation of its properties. Datasets used in the experimental evaluation are taken from a widely used open source software project (161120 textual comments), as well as from an industrial project in the home energy management domain. We make them available for experiment replication purposes. On these datasets, our approach is able to successfully classify messages into Feature/Enhancement and Other, with F-measure of 0.81 and 0.84 respectively. We also found evidence that there is an association between types of *speech-acts* and categories of issues, and that there is correlation between some of the *speech-acts* and issue priority, thus motivating further research on the exploitation of our *speech-acts* based analysis technique in semi-automated multi-criteria requirements prioritisation.

1. Introduction

Nowadays users of software applications and services provide their feedback based on their perceived quality of experience (QOE), in the form of short comments and rankings, or even get engaged in online discussions, which are supported by social media or dedicated web-based communication platforms, such as App stores, user forums, mailing lists, wikis, newsgroups, and blogs. The rapidly increasing volume of such online user feedback represents an invaluable part of software data that can be analysed to support decision-making in software engineering, in what is known as *software analytics* [1], and motivates the huge growth in the number of research in this area, *e.g.,* in App store analysis for software engineering, as reported in the survey performed by Martin et al. [2], which considers the period 2010 to 2015.

Indeed, the development and consolidation of big data architecture and analytics, the massive volume and variety of available software data, and the high rate at which these data are produced, are offering the opportunity to address in a novel way information needs of software engineers at support of decision-making in different software development tasks, ranging from requirements engineering to maintenance and release planning, as discussed in [3]. For instance, knowing which are the most liked and important features for the customers appears as one of the identified information needs in the category concerning the understanding of customer interests and requirements.

Along this trend, in requirements engineering (RE) the *Crowd-based Requirements Engineering (CrowdRE)* [4] paradigm has been proposed, which defines the set of concepts, methods and techniques necessary to collect, analyse and manage requirements expressed by members of a crowd of users in the form of online feedback. An ontology for online user feedback characterises the user feedback communication process, the diverse formats in which user feedback can be expressed and the type of information that can be extracted from this feedback. Such ontology helps understand which type of analysis techniques are needed to process online user feedback [5]. Focusing on the analysis

* Corresponding author.
E-mail address: kifetew@fbk.eu (F.M. Kifetew).

https://doi.org/10.1016/j.is.2018.08.003
0306-4379/

of online textual feedback, Natural Language Processing (NLP) techniques [6] are first applied to filter out irrelevant data. Text mining, such as sentiment analysis and topic modelling, and machine learning (ML) techniques are then used to identify feedback that can fall into different categories, such as *bug report* or *feature request* [7,8], and serve as input to release planning tools at support of app developers to identify which new feature to include in the next release [9–11]. Manual analysis and supervised ML techniques are applied to investigate tweets related to software applications so as to characterise their relevance to software engineers and to non-technical stakeholders [12–15].

In our research we focus on the analysis of online discussions such as those that take place in open source software (OSS) mailing lists and user forums, where different stakeholders including developers and software end-users, engage in threaded discussions with the purpose of contributing to the software improvement, or in dedicated user-feedback gathering platforms, through which users send their comments directly to software developers, and often to the user community as well. These online discussions can be considered as explicit, directed feedback in which the sender reveals her intention and can affect the receivers' attitude about a subject, through the specific *speech-acts* [16] she uses in her comments [5]. That is, the structure of the sentences through which this feedback is expressed can provide useful information about the intention of the user who expressed it and helps better interpret the user's experience that generated it. In order to capture such users' intentions, other text mining techniques, different from sentiment analysis and topic modelling, are worth exploring.

Our aim is to define an analysis technique, rooted on the *speech-act* theory [16], which can be used alone or in combination with other analysis techniques, such as sentiment analysis, to extract requirements-related information from online discussions that can lead to the formulation of possible requirements.

Speech-act analysis has already been applied for the analysis of online discussions, for example, to investigate most frequent intentions expressed in status messages by users of social networks [17], and in the online teaching domain to understand students' intentions expressed in their queries to teachers [18], or during discussions with peers [19]. In these works, the *speech-acts* that are used in conversations about the specific domain under consideration are first identified, and then used during manual annotation of the conversations, which is performed by independent annotators.

Recently, the automated rule-based identification of *speech-acts* from online discussions has been presented in 2014 [20]. In this previous work, we motivated a mechanism for exploiting the identified *speech-acts* towards extracting information that could enhance requirements elicitation, which we presented at the 29th International Conference on Advanced Information Systems Engineering (CAiSE) [21]. In this article we present an extended version of that work, by further enhancing our *speech-act* based analysis technique as well as the empirical investigation and datasets. Specifically, we extended the experimental evaluation presented at CAiSE [21], which aims at characterising the distribution of different types of *speech-acts* in messages that contain requirements-related information, by considering a larger dataset extracted from threaded discussions in the issue tracking system of the Apache OpenOffice (AOO) project and a new dataset containing user feedback messages of a software application in the home energy management domain, called interactive Energy Saving Account (iESA). Results from the application of the *speech-acts* analysis to the datasets consolidate further our previous findings about the significantly higher occurrence of certain types of *speech-acts* in Enhancement requests compared to requests of Other type. We investigate the capability of *speech-acts*, combined with other properties that can be extracted from online feedback, to predict whether new messages contain Feature or Enhancement requests or not. The experiments conducted on the two datasets provide encouraging results. For instance, Enhancement requests are predicted with higher F-measure values than those reported previously in [21].

Moreover, we investigate whether there exist relations between the various *speech-acts* and the importance associated with issues, represented by the priority level assigned to the issues by the project managers. The investigation reveals that there is in fact dependence between *speech-acts* and issue priority. Furthermore, it is also revealed that there is a positive evidence about correlations between groups of *speech-acts* expressed in the feedback messages, and the associated issues marked with a given level of priority.

The remainder of the paper is as follows: Section 2 presents two motivating scenarios taken from the AOO and from the iESA projects respectively. Section 3 describes the *speech-acts* based approach to the analysis of online discussions. Section 4 introduces the three research questions which guide the experimental evaluation of the key properties of *speech-acts* based analysis. Section 5 shows the results through some plots, and Section 6 presents a discussion of them along the threats to validity. Section 7 presents main related work, and Section 8 concludes the paper and outlines our future work.

2. Motivating scenarios

Online user feedback plays an increasingly important role in the development, maintenance and evolution of a piece of software. In this section, we present two scenarios in which online discussions and user messages are used by developers to make decisions about the maintenance and evolution of software. The first scenario concerns the analysis of comments in threaded discussions extracted from issue tracking systems in large OSS projects, such as Apache OpenOffice, where a developer has to analyse several user messages on a daily basis. The second scenario is taken from one of the industrial use cases of the SUPERSEDE[1] project, which aims at creating a set of integrated methods and tools to enable a data-driven approach to software evolution. This case study concerns a small-medium enterprise, where the help-desk responsible, a highly experienced person, manages feedback messages which are collected through different channels such as via telephone, e-mail, online tool etc., building an effective bridge between end-users and product developers.

2.1. Scenario 1: stakeholder feedback analysis for software maintenance and evolution in OSS

Active stakeholders in OSS communities, which include users of the developed software, as well as different types of collaborators such as developers and analysts working in a distributed way can be up to 100,000.[2] These active stakeholders are involved and interested in maintaining and evolving different types of software, such as web servers, IDEs, productivity suites, etc. Some developers that work for companies are also contributors of OSS projects since companies use open source software as core elements of their day-to-day activities, such as operating systems, databases management systems and development tools.[3] These

1 http://www.supersede.eu.

2 http://www.slideshare.net/blackducksoftware/open-source-by-the-numbers.

3 http://www.slideshare.net/blackducksoftware/2016-future-of-open-source-survey-results.

Please cite this article as: I. Morales-Ramirez, F.M. Kifetew and A. Perini, Speech-acts based analysis for requirements discovery from online discussions, Information Systems (2019), https://doi.org/10.1016/j.is.2018.08.003.

stakeholders convey their concerns, for example bugs and new features, or they may suggest modifications. There is a continuous exchange of messages that must be read, analysed, replied to and considered for making decisions and put them into action by the OSS (*e.g.,* OpenOffice) developers or analysts. We refer to these messages as stakeholder feedback, more specifically as user feedback when those messages come from end-users of the software.

The stakeholder feedback is collected through e-mails, discussion forums, or issue tracking systems. This implies an asynchronous mode of communication expressed as a chain of written messages in natural language (*e.g.,* "One of my spreadsheets is no longer showing in my documents"). Sometimes there are messages expressing a praise that motivates volunteers to continue their work towards improving the OSS, other times there are only complaints. The important point is that OSS volunteers form a special kind of community that achieves the maintenance and evolution of software in a distributed setting with continuous online communication.

The high volume of feedback sent by stakeholders is processed manually by any available volunteer, but sometimes it takes the involvement of other specialists to solve the issue who are better suited to deal with such types of stakeholder feedback. The idea of providing a tool to support the filtering of feedback and redirect it to the right role (analyst or developer) may be crucial to saving time and effort.

2.2. Scenario 2: user feedback analysis for software maintenance and evolution in SUPERSEDE

SEnerCON,[4] a partner in the SUPERSEDE project, is a small-medium enterprise with more than 25 years of experience in the domain of energy efficiency management. It currently employs about 15 software developers and engineers. SEnerCON provides several web applications including: (1) an application that enables end-users (house owners) to monitor and analyse their energy consumption, called interactive Energy Saving Account — iESA; and (2) applications that guide and advise end-users on how to save energy in every day life through behavioural or technical changes.

The iESA application counts thousands of users. Most of the features of the application are free, and the only obligation for users is to register for an account and accept that their data (including usage logs) are used and analysed by the company upon anonymisation.

SEnerCON collects hundreds of user feedback messages per month by means of five main channels, namely contact form, e-mail, hotline, forum, and app stores. The end-users express their feedback using natural language text. According to the helpdesk, end-users may prefer to use a forum rather than a dedicated feedback channel because "they might perceive to get a broader audience in the forum and make them think that SEnerCON will pay more attention to a feedback elaborated through discussions in a forum".

The initial analysis of end-user feedback is performed by the helpdesk who can contact the user to ask for clarifications, and also with members of the development team to get help from them. Such initial analysis can motivate the inclusion of new tickets in the issue tracking system, used by the development team to keep track of pending issues, which can be later addressed during software maintenance or product evolution cycle. Specifically, the helpdesk reads the user comments from the forum, try to understand if it is an information request he may answer directly, a complaint for something not working properly or if they contain suggestions for new features. This task seems to be not so effort demanding, considering that feedback arrives at a relatively low rate of about five per day, however it does seem to require a strong experience and knowledge about the product in order to manage it in an efficient and effective way. This becomes evident in situations where the person in charge of the helpdesk takes a holiday, and it is "almost impossible to find some one who can perform his task at an acceptable level".

The previous scenarios are examples of cases wherein an automated tool can analyse the end-user feedback with the following purposes: (1) suggest if end-user feedback concerns a feature request or enhancement of existing features, or other type of feedback (e.g., bug fix request); and (2) indicate which requests are more important and need to be urgently addressed, this plays an important role in facilitating the management of end-user feedback and the extraction of relevant information that supports the maintenance and evolution of the software system under consideration. The work presented in this paper takes a step in this direction and proposes the analysis of textual feedback by means of a *speech-acts* based analysis technique.

3. Speech act based analysis of online discussions

This section explains our approach by giving a brief background from previous research work [20–22]. After this, we present the conceptual definition of the approach and the last section gives the technical details of its implementation.

3.1. Speech-acts

In our work we focus on artefacts that contain conversations between stakeholders, such as between end-users and the helpdesk team, or between users and developers, thus the dialogue structure, which rests on dialogue acts, can provide relevant information.

Dialogue acts can be studied using the *speech-act* theory of Searle [16]. In a nutshell, *speech-act* theory claims that when a person says something she/he attempts to communicate certain things to the addressee, which affect either their believes and/or their behaviour. So, for instance, Alice says "Please, bring me the keys" to Bob. This utterance expresses Alice's intention (technically named illocutionary act) to make Bob aware that she is expecting him to bring her "the keys", and the effect (technically named perlocutionary act), is that Bob accepts her intention and is willing to bring her the keys. Or if a software user says "I'm confused about this functionality…", this can be classified as a *Concessive speech-act*, *i.e.,* a statement that reveals a belief contrary to what the sender would like to believe or contrary to what he/she previously believed. *Speech-acts* represent the meaning of an utterance at the level of illocutionary force [23].[5]

The main motivation about investigating *speech-acts* is that they constitute the basis of communication, with a wide variety of applications. For example, a meeting summariser needs to keep track of who said what to whom, and a conversational agent needs to know whether it was asked a question or ordered to do something [23,24].

The *speech-acts* are taken from a taxonomy proposed by the linguistic community, specifically, we revised the taxonomy of Bach and Harnish in [25] and adapted it for the context of software development. Such an adaptation has been based on the selection of certain lexicon commonly used for describing software features, modifications or bugs. Example of lexicon for asking a new feature are: "add feature" and "would love this feature"; for describing a problem: "first…then…after…" and "when…the app/software/application…"; for describing a modification: "please change" and "it would improve".

[4] http://www.senercon.de.

[5] An illocutionary force refers to the pronouncing of a statement with an intention.

Please cite this article as: I. Morales-Ramirez, F.M. Kifetew and A. Perini, Speech-acts based analysis for requirements discovery from online discussions, Information Systems (2019), https://doi.org/10.1016/j.is.2018.08.003.

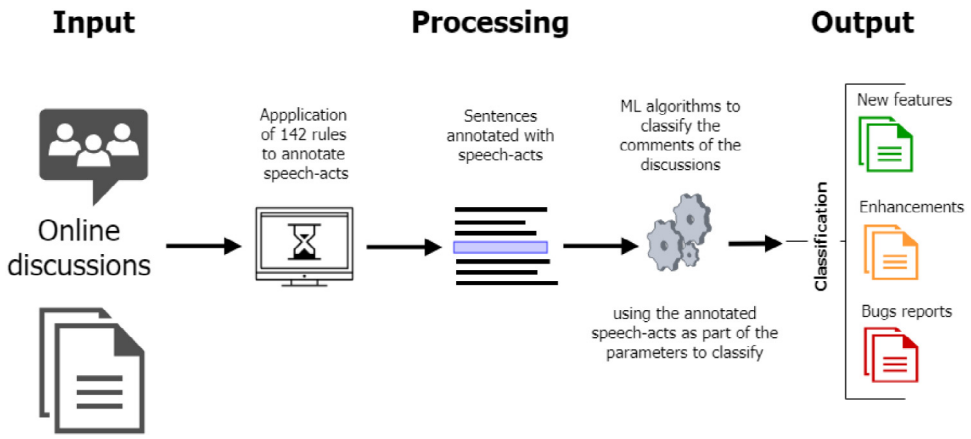

Fig. 1. *Speech-acts* based analysis approach.

Table 1
List of *speech-acts* that support the analysis.

c-Assertive	c-Requestive	c-Responsive	c-Attachment	c-Other
Assertive	Requestive	Responsive	Attach	Descriptive
Confirmative	Requirement	Suggestive	Code line	Accept
Concessive	Questions	Suppositive	URL link	Reject
			Log file	Negative opinion
				Positive opinion
				Thank
				Informative

3.2. Speech-acts based analysis

In a previous work we studied the effort needed to manually annotate a set of seventeen *speech-acts* found in online discussions, by asking human annotators to perform such an activity [22]. As a result of this work, we realised that in order to make the automated annotation of *speech-acts* possible, it was necessary to group the *speech-acts* into the following analysis categories (or grouped *speech-acts*): c-Assertive, c-Responsive, c-Requestive, and c-Attachment.

Moreover, the evolution of this research has led us to consider the usage of NLP techniques to allow us manipulate different *speech-acts* more precisely. We revised the analysis categories and split them, in such a way that, for instance the category c-Assertive corresponds to the individual original *speech-acts*: Assertive, Confirmative, Concessive; the category c-Responsive corresponds to the individual *speech-acts*: Responsive, Suggestive, Suppositive, etc., as indicated in Table 1.

Based on the lexicon and specific verbs or adverbs we can determine the difference between *speech-acts*. For instance, the Requestive *speech-act* differs from the Requirement *speech-act* in that the first one has verbs such as: insist, solicit, urge; meanwhile the second *speech-act* has verbs such as: need, demand, require. Another difference is in the formulation of the *speech-acts*, for example the Suppositive *speech-act* can be formulated as "I guess the problem…", and for the Attach *speech-act* the formulation can be "attachment created".

The first row of Table 1 shows the names of the higher level analysis categories, including c-Other *speech-acts*. Then, the corresponding individual *speech-acts* are listed in each column.

Our approach relies on lexico-syntactic rules used to process the textual comments, and for this journal we have redefined them, resulting in a larger set, from 136 to 142 rules. Main benefits obtained from the extension and improvement of the lexico-syntactic rules previously used in [21] are: (1) an improved handling of case sensitivity in the various rules; (2) the identification of some abbreviations, *e.g.,* "WDYM?", "AFAIK", "HTH"; and (3) the inclusion of verbs that are synonyms of the *speech-acts* verbs. The synonyms have been taken from Wordnet 3.1 (available online[6]).

To apply the *speech-acts* based analysis to online feedback we have defined a tool-supported process, whose conceptual approach is depicted in Fig. 1. The input of our approach are the online discussions in a digital format, which are pre-processed in order to remove words or text that can be considered noise.

For the processing of the input, the tool performs the annotation of *speech-acts*. This tool considers some components that exploit the 142 rules (lexico-syntactic rules) defined by some of the authors, based on the requirements knowledge they have acquired along the experience and also based on the manner stakeholders express their needs in the online discussions.

The sentences annotated with *speech-acts* are used as parameters to train machine learning algorithms for the purpose of supporting the classification of text comments into new features, enhancements and bug reports. It is important to clarify that the *speech-acts* parameters are used along with other types of parameters better explained later on.

The output of the tool is the classification of the comments into different categories. Since the interest of this research is the discovery of requirements, we have configured our tool to differentiate between comments whose textual content has the potential of containing requirements-related information and the comments that discuss other type of information.

3.3. Implementation of the approach

To exploit the textual-based content of the online discussions we use the NLP framework called GATE (General Architecture for Text Engineering) [26]. Details of the GATE library are given in the Appendix A. The proposed *speech-acts* based analysis technique builds on the hypothesis that textual messages sent by software users are meant to suggest a new feature, or to request enhancements of an existing functionality, or simply to complain. These messages contain different types of *speech-acts*, so that *speech-acts* could be used as indicators of potentially relevant information from a requirements engineering perspective.

The implementation of the actual process is depicted in Fig. 2. The online discussions of the AOO issue tracking system are extracted as XML files which are cleaned, parsed and stored in a MySql database, this step is done by our tool.

For the cleaning step, we used Java regex patterns to remove dates, identify links and unify them into a unique codification (*i.e.,*

[6] https://wordnet.princeton.edu/wordnet/download/current-version/.

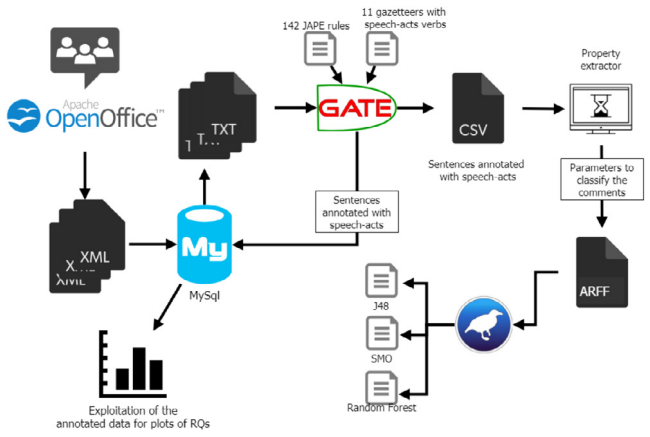

Fig. 2. Implementation of the *speech-acts* based analysis approach.

http://www.); common contractions and file extensions are also eliminated. Smiley faces and sad faces are replaced by the words SMILEY and SAD, respectively. Other elements, for example –, =, -, *"* .. are removed from the text; or double elements replaced with one, for example *??* to *?*. After this cleaning step, each comment contained in the discussions are parsed into sentences that are processed by our tool that exploits the GATE library (see Appendix A). GATE makes possible the implementation of the lexico-syntactic rules (rules whose grammar is established by GATE and are in the format of JAPE rules[7]). Each rule uses gazetteers to annotate the twenty *speech-acts* (defined previously in Table 1), whose occurrence is found in the text of the comments. A gazetteer[8] consists of a set of lists containing the verbs related to each *speech-act*. These lists are used to find occurrences of the verbs in text, e.g. for the task of the rule based recognition.

Each online discussion has some properties such as type of issue (e.g., Enhancement, Feature) and each of the comments contained in the discussion are stored in the database, along their properties. Then a file, in plain text format, is generated per comment, which is the required input for our tool in order to proceed with the annotation of each sentence of the comments with *speech-acts* that are found in the text. This means that our approach considers a level of analysis at the sentence level. There are two ways of storing the annotations, the first one is as a CSV (comma separated values)[9] file and the other is to save the annotations in a database. The CSV file is used by another tool to extract some parameters and generate an ARFF (Attribute-Relation File Format)[10] file to be used for building a classifier, using Weka[11] for training three common ML algorithms, *i.e.,* J48, SMO and Random Forest.

Our tool extracts the parameters using Stanford CoreNLP[12] and SentiWordNet[13] as part of the process extraction. We have formulated 43 parameters as "characteristics" to classify each comment into an Enhancement or Other type of feedback. Table 2 displays the total number and type of parameters. With respect to the parameters used in the conference paper at CAiSE17 [21], we have revised them and introduced 2 additional parameters.

With the CAiSE17 paper we clearly identified that there were more parameters related to sentiments than to *speech-acts*. For this reason we evolved our list of parameters towards increasing the number of the *speech-acts* parameters and reducing the sentiment ones, as a new strategy for improving the results obtained during the classification. The increment was due to the improved rules to annotate *speech-acts*, so that we are considering them as "enhancers" of what is expressed in a sentence.

Table 2
List of parameters to be used by Weka.

Type of parameter	CAiSE17 [21]	Current version
Related to *speech-acts*	7	18
Related to sentiments	13	9
Related to the structure	20	15
Type of class	1	1
Total	41	43

The *speech-acts* parameters are quantified as the frequency of occurrence of them in the text of the comments. Example of these parameters are: number of informative / responsive / requestive / and assertive verbs, number of assertive / confirmative / concessive / requestive / requirement / question / responsive / suggestive / suppositive / attach / code / logFile / urlLink / and descriptive expressions.

The parameters related to sentiments are the number or positive/negative adjectives/nouns and their corresponding sentiment score, as well as the overall sentiment score of a sentence. The parameters related to the structure are for instance the number of sentences in a comment, the number of verbs/nouns/adjectives, number of questions marks, number of certain type of brackets, the length of the comment, etc. The last parameter refers to the type of class that a comment is assigned, for example, it can be an Enhancement or Other.

Each online discussion is already labelled into Enhancement, Feature or other type, as explained before; and for the current work we have grouped the feedback types Enhancement and Feature as a single class Enhancement, and the types of feedback labelled as Patch, Defect and Task are assigned to the class Other.

4. Experimental evaluation

In this section we describe the experiments we have carried out using two datasets that correspond to the scenarios presented in Section 2. We first present the research questions that guide the experimental evaluation of the properties of the *speech-acts* based technique. We then describe the datasets we used, the experimental setting and procedures we followed, results are presented by means of a variety of plots.

4.1. Research questions

We investigate the following main research questions:

- RQ1 *What are the speech-acts expressed in online discussions that can lead to the discovery of new requirements?*
- RQ2 *Can the speech-acts be used as parameters,*[14] *towards building a classifier for feedback messages into feature/enhancement requests or other?*
- RQ3 *How are speech-acts and the importance of feedback messages correlated?*

7 https://gate.ac.uk/sale/thakker-jape-tutorial/GATE%20JAPE%20manual.pdf.
8 https://gate.ac.uk/sale/tao/splitch13.html.
9 https://en.wikipedia.org/wiki/Comma-separated_values.
10 https://www.cs.waikato.ac.nz/ml/weka/arff.html.
11 http://www.cs.waikato.ac.nz/ml/weka/.
12 http://stanfordnlp.github.io/CoreNLP/.
13 http://sentiwordnet.isti.cnr.it/.

14 We use the term *parameter* as synonym of *feature* in machine learning parlance, to avoid confusion with the term feature, which in the context of this paper indicates an observable behaviour of a software application that can be linked to detailed level requirement or property. A commonly accepted definition for this term in software engineering, however, is still missing [27].

With *RQ1* we explore the occurrence and distribution of the various *speech-acts* in the feedback messages of threaded discussions. By doing so, we can spot any trends in the distribution of the *speech-acts* which could lead to the discovery of interesting behaviours. Furthermore, we can also observe the prevalence of each *speech-act* with respect to the type of feedback message, *i.e.,* Feature/Enhancement request and Other type of feedback.

After exploring the distribution of the various *speech-acts* in *RQ1*, with *RQ2*, we investigate the possibility of using the *speech-acts*, together with other information extracted from the feedback, to build a classifier that can automatically label new feedback messages as Feature/Enhancement requests or Other types of requests.

With *RQ3* we investigate the relation, if any, between *speech-acts* and the importance that has been assigned by the development team to the particular feedback message. The importance is a property that reflects how urgent a certain feedback message is. We would like to study which *speech-acts* are likely to influence this urgency.

4.2. Datasets

Here we describe the datasets we used in our experiment. They correspond to the two scenarios discussed in Section 2, namely *OpenOffice* and SEnerCON.

4.2.1. OpenOffice

Our first dataset is composed of user feedback gathered from the issue tracking system of the Apache OpenOffice (AOO) community.[15] We use AOO because the community is very active in collaborating and communicating through written messages. Moreover, the comments in such systems are more complex and amenable to *speech-acts* analysis than the commonly used reviews from App stores. Furthermore, the data refers to online threaded discussions whose first comment has the characteristic of being already labelled by members of the community as defect, feature, enhancement, patch, or task. Besides this, we believe the data contains *speech-acts* which are more complex than sentiments (which are popularly used in the literature [8,28]) and can provide extra information useful to developers and requirements analysts. Another important characteristic of the AOO dataset is that it has other properties such as *status* of the issue (*e.g.,* confirmed, unconfirmed), *priority* (*e.g.,* P1-highest priority, P5-lowest priority), and *severity* (*e.g.,* blocker, critical) that together constitute a combined property, referred to as *importance*.

In our previous work [21], we set-up a dataset of 40,872 comments, in 6568 threads, that we obtained from the AOO issue tracking system between the years 2012–2013. For the current work, we extended this dataset by gathering more issues from the AOO issue tracking system via a custom made crawler. The crawler saves each thread in the issue tracking system in XML format. We parsed these files and stored them in a MySql database for a better manipulation of the information contained in the comments. Each thread contains at least one comment, but some can have more than one hundred comments. The total number of comments in the new dataset amounts to 161,120, which are contained in 23,740 threads, that occurred in the period between the years 2001–2017. We split each comment into sentences in order to store each one in the database for later analysis.

[15] https://bz.apache.org/ooo/.

4.2.2. SEnerCON

The second dataset is obtained from the feedback gathering system of SEnerCON (see Section 2). It is composed of 575 user feedback messages received by the company regarding one of its applications — iESA. The dataset is labelled by domain experts in SEnerCON as feature, enhancement, and defect. The number of feedback messages in this dataset is small, as compared to that of the AOO dataset, because most users of the iESA application are German speaking, and hence the feedback messages they provide are also in the German language. Since our analysis technique works on text in the English language, they needed to be manually translated by the domain experts in SEnerCON. Such manual translation being time consuming, we were able to obtain only 575 messages translated to English. In future work, our proposed techniques could be extended to other languages as well. Regardless of the small data size, we apply our analysis on the dataset to assess its applicability, as this dataset represents situations in small-medium enterprises where the rate of user feedback arrival is not so high.

Both datasets are available as part of a replication package, described in Appendix C.

4.3. Experiment setup

The overall procedure we followed for conducting the experiment is as follows: (1) filter dataset and identify relevant portions to be used in the experiment, (2) determine reasonable parameters, (3) process dataset to generate dataset in required formats (CSV, ARFF), (4) apply measures for handling unbalanced datasets, (5) define metrics to observe, and (6) perform statistical analysis to be able to answer the research questions.

In order to facilitate future replication studies, we make available a replication package containing the various datasets as well as the rules for identifying *speech-acts*, described in Appendix C.

4.3.1. Filtering the dataset

Before conducting the actual experiments, we had to make further considerations with respect to the AOO dataset. In particular, since the AOO dataset is composed of issues reported by users in the community, they could potentially contain inaccurate data. In particular, the type of issue indicated by the user might not be set correctly at the time of submission. For instance, an issue reported by the user as a *Defect* may actually turn out to be a *Feature*. Consequently, we filter the issues we collected from the issue tracking system by selecting only those issues whose *status* is acknowledged by the development team as *confirmed*, *accepted*, or *reopened*. This way, we are fairly certain that the issue is labelled with the right *type*. This filtering reduced the number of threads to 21,477 (6907 for Enhancement and Feature, 13,852 for Defect, and 718 for Patch and Task).

We further group the dataset into two categories, i.e., Enhancement and Other, based on their assigned types. This is motivated by two main reasons: (1) since our research is focused on exploring information which is relevant for identifying new requirements, we focus on those issues that request for new features or enhancement of existing functionalities (which we refer to as Enhancement); the rest of the issues fall into the second category (which we refer to Other), and (2) a binary classification problems lead to classifier models with better accuracy than multi-class classifiers, hence reducing the number of categories into two is expected to boost the classifier accuracy without significant loss in information obtained from the dataset.

Consequently, for the AOO dataset, we group as Enhancement those issues with type *Feature* and *Enhancement*, while we group as Other those issues with type *Defect*, *Patch*, and *Task*. For the SEnerCON dataset, we group as Enhancement those issues with type *Feature* and *Enhancement*, while the group Other contains those issues with type *Defect*.

Please cite this article as: I. Morales-Ramirez, F.M. Kifetew and A. Perini, Speech-acts based analysis for requirements discovery from online discussions, Information Systems (2019), https://doi.org/10.1016/j.is.2018.08.003.

Table 3
List of parameters that have been weighted.

Parameters related to *speech-acts*	Weight
num_assertive_expressions	4.3
num_confirmative_expressions	4.2
num_concessive_expressions	4.3
num_requestive_expressions	4.9
num_requirement_expressions	5.1
num_question_expressions	2.2
num_responsive_expressions	2.1
num_suggestive_expressions	4.5
num_suppositive_expressions	4.3
num_attach_expressions	4.0
num_code_expressions	5.1
num_descriptive_expressions	2.3
num_logFile_expressions	4.1
num_urlLink_expressions	2.3

4.3.2. Generate required dataset formats from the raw dataset

We apply our technique on the datasets to process them and extract the various parameters (recall Section 3) that are required for training the classifier model. Ultimately, the raw dataset is transformed into a set of property vectors in Weka's ARFF format, ready for training the required models. Furthermore, from the original dataset we generate simplified representations in CSV format which facilitates the plotting of the data for further analysis, as described in Section 3.

4.3.3. Setting parameter weights

We observed that not all *speech-acts* are of equal importance from the perspective of building a classifier that classifies issues. Consequently, we have selectively applied *weights* to a subset of the parameters that we have identified (recall Section 3). We discarded the parameters related to the *speech-acts* verbs, and only considered the *speech-acts* expressions because we believe they are more powerful in communicating an intention, for this reason we applied the weights shown in Table 3. These weights are defined by starting from custom values defined based on our observation of the various *speech-acts*, then improved in an iterative manner through sensitivity analysis trials. As we can see from Table 3 some of the 14 parameters have equal weights, for instance, we consider that code expressions and requirement expressions are very distinctive and we assigned a weight of 5.1. It is important to clarify that while we believe that these set of weights are reasonable, a future research work could explore the possibility of employing optimisation heuristics for automatically finding optimal parameter values.

4.3.4. Dealing with unbalanced dataset

A typical problem when working with datasets for building classifier models is the problem of unbalanced classes. Since in both of our datasets the classes are not balanced, we applied the SMOTE (Synthetic Minority Over-sampling Technique) [29], to increase the number of instances in the smaller class. SMOTE applies oversampling of the class with less observations, to overcome the problem of unbalanced datasets. The other alternative would have been to drop the excess instances from the larger classes so that they eventually have the same number of instances as the smallest class. However, this approach will significantly reduce the size of the dataset. Furthermore, in the context of user feedback, it is typical to have fewer feature requests as compared to other types of requests (e.g., bug fix requests). Hence, the issue of unbalanced datasets is inherent to datasets involving user feedback, and we believe that oversampling is an effective means for dealing with this problem.

In our experiments, we use the implementation of SMOTE available in the Weka tool. For the AOO dataset, after applying SMOTE with 110%, the resulting dataset is composed of a total of 29,074 instances, 14,504 labelled as *Enhancement* and 14,570 labelled as *Other*. Similarly, for the SEnerCON dataset, we applied SMOTE with 150%, resulting in a total of 812 instances, 395 labelled as *Enhancement* and 417 labelled as *Other*. Note that the oversampling with SMOTE is applied on the feature vectors in ARFF format, and not on the original (textual) datasets.

4.3.5. Metrics and analysis tools

In order to answer our research questions, we employ a number of metrics for measuring and quantifying our experimental results, as well as a number of statistical analyses techniques. In particular, to answer *RQ1* we plot the distribution of the various *speech-acts* over several threads in the AOO dataset so that we can easily observe any emerging trends. The plots are accompanied with descriptive statistics that give insight into the observed phenomena. For *RQ2*, we employ the widely used precision, recall, and F-Measure metrics in conjunction with 10-fold cross validation to objectively assess the accuracy of predictive models trained with our approach. For *RQ3* we make use of statistical tests of independence in conjunction with association plots to explore the interaction between among the various *speech-acts* and the importance of issues.

5. Results

In this section, we discuss the results and provide answers to the research questions based on the observations from the results. The results are presented according to the metrics defined previously.

5.1. Distribution of speech-acts

Fig. 3 reports the distribution of *speech-acts* that have been found in the online discussions of AOO. The *y*-axis shows the percentage of *speech-acts* and the *x*-axis shows the types of *speech-acts*. As previously discussed in Section 4.3.1, the classes *Feature* and *Enhancement* are merged into *Enhancement*, while the classes *Defect, Patch, Task* correspond to *Other*.

Overall, Fig. 3 shows the proportion of *speech-acts* in the AOO dataset. In particular, we can see that for the *Enhancement* class two important *speech-acts*, namely *Requestive* and *Requirement*, stand out over the other *speech-acts* in these types of discussions. On the other side, for the *Other* class, three *speech-acts*, namely *Attachment*, *CODE_LINE*, and *LOG_FILE*, particularly stand out with nearly 90% occurrence. The rest of the *speech-acts* are present in both classes with varying levels of frequencies.

While Fig. 3 depicts the distribution of the frequencies of the various *speech-acts* in the first comment sent by the user of AOO, it does not offer further details about the *speech-acts* in subsequent comments in the threads. To get a more detailed insight into the distribution of the *speech-acts* in subsequent discussions to the initial comments, i.e., the thread of discussions that follows after initial issue is reported by a user, we plot the distribution of the *speech-acts* in each thread. For reasons of space and readability of the plots, we only present the distribution of *speech-acts* that are discussed in the first 10 comments per each online discussion thread. Furthermore, we focus on those *speech-acts* that showed higher frequencies of occurrence in Fig. 3, i.e., the *speech-acts Requestive*, *Requirement* (for *Enhancement*), *Attachment*, *CODE_LINE*, and *LOG_FILE* (for *Other*).

Fig. 4 depicts the distribution of the aforementioned *speech-acts* in the *Enhancement* discussions, and the important points are marked with the highest value that appears in the corresponding position of the comments. The *y*-axis displays the percentage of *speech-acts* annotated and the *x*-axis corresponds to the order

Please cite this article as: I. Morales-Ramirez, F.M. Kifetew and A. Perini, Speech-acts based analysis for requirements discovery from online discussions, Information Systems (2019), https://doi.org/10.1016/j.is.2018.08.003.

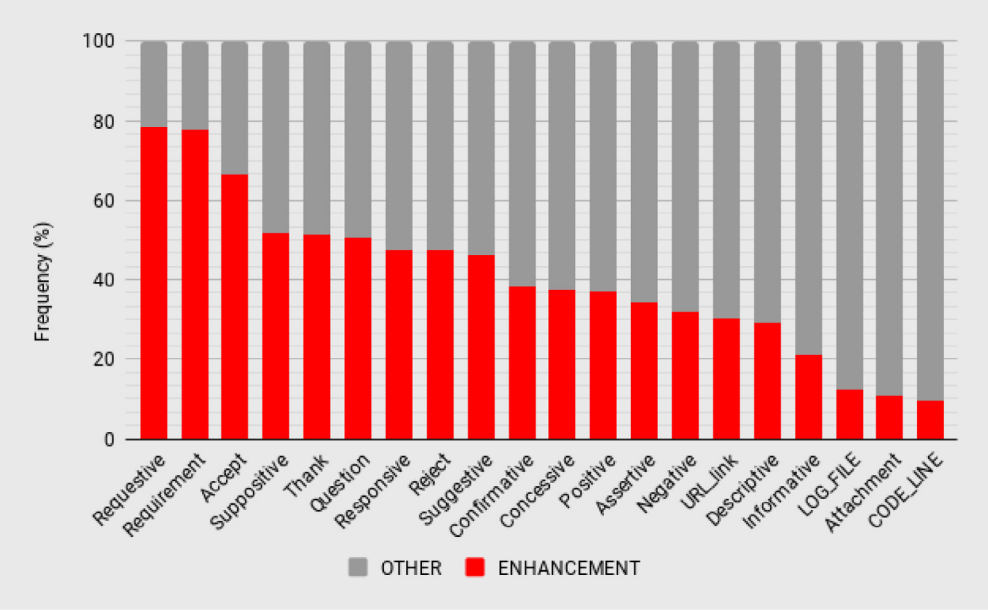

Fig. 3. Distribution of the *speech-acts* on the first comment of the 21,477 threads.

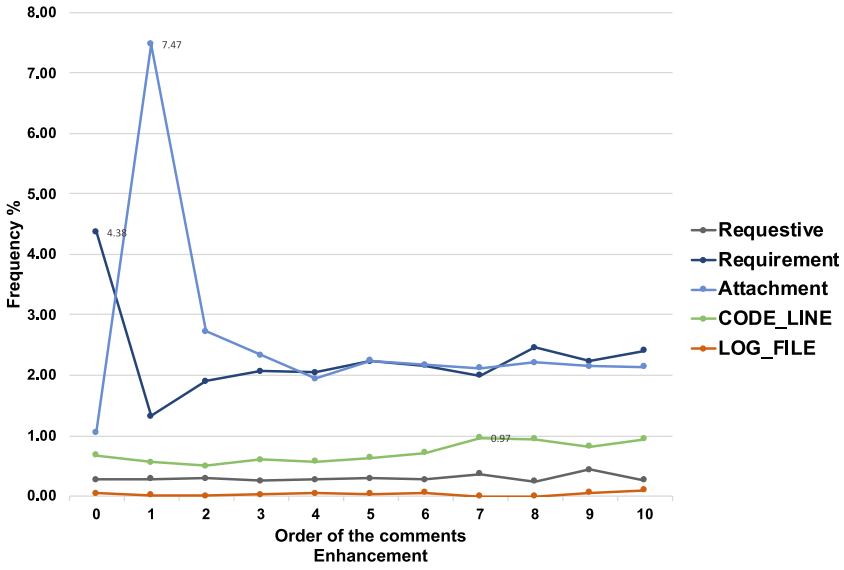

Fig. 4. Distribution of *speech-acts* by the order of the comments, for feedback type Enhancement.

of the comments. The *speech-act Requirement* appears in the first comment 4.38% of the time while the *speech-act Attachment* appears mostly in the second comment 7.47% of the time. The *speech-act CODE_LINE* reaches the highest appearance in the 7th and 8th comments, with a frequency of almost 1%.

On the other side, Fig. 5 presents the distribution of the same *speech-acts* discussed above, but for the class *Other*. Hence the differences could easily be observed in a comparative manner for the two classes. The *speech-act CODE_LINE* shows a peak in the first comment with a frequency of 2.69%, while the *speech-act Attachment* again stands up in the second comment with 20.13%. The *speech-act Requirement* only has an occurrence of less than 1%, in the 10th comment.

5.2. Performance of the classifier

We trained three ML algorithms (Random Forest, J48, SMO) in Weka, using the parameters described earlier, for classifying comments as *Enhancement* or *Other*. We present different combinations of parameters, to see any differences between excluding the *speech-acts* parameters or the sentiment parameters, with the objective of understanding their effect on the classification of the comments.

Table 4
AOO:Using the 43 parameters.

	RF			J48			SMO		
	P	R	F-M	P	R	F-M	P	R	F-M
Enhancement	.87	.76	**.81**	.79	.74	.77	.77	.53	.63
Other	.79	.89	**.84**	.76	.81	.78	.64	.84	.73

The results for the AOO dataset are presented in Tables 4, 5, 6. The results in all the tables with different combinations of parameters show that the Random Forest (RF) algorithm gives better results compared to J48 and SMO. In Table 4 we observe that

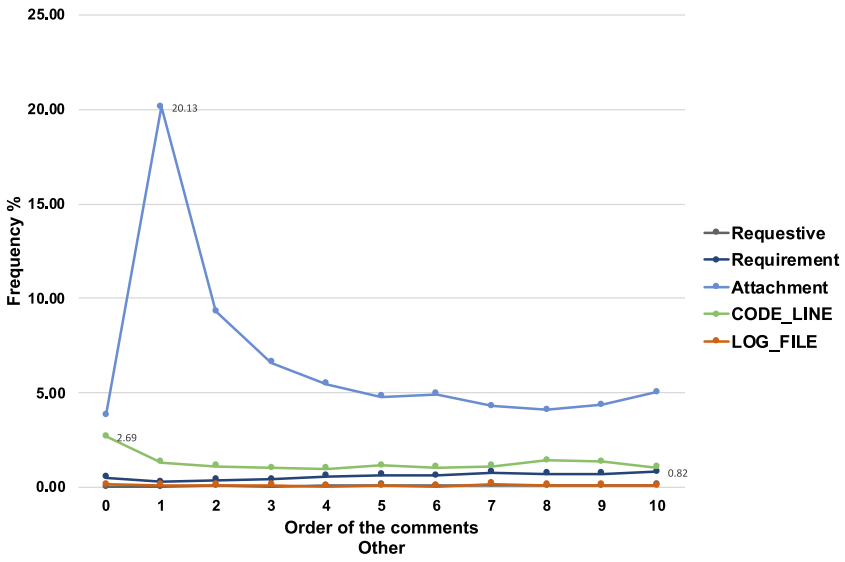

Fig. 5. Distribution of *speech-acts* by the order of the comments, for feedback type Other.

Table 5
AOO:Using 34 parameters (no sentiment).

	RF			J48			SMO		
	P	R	F-M	P	R	F-M	P	R	F-M
Enhancement	.85	.76	**.80**	.83	.71	.77	.78	.51	.62
Other	.78	.87	**.82**	.75	.86	.80	.64	.85	.73

Table 6
AOO:Using 25 parameters (no speech acts).

	RF			J48			SMO		
	P	R	F-M	P	R	F-M	P	R	F-M
Enhancement	.84	.74	**.79**	.77	.71	.74	.70	.48	.57
Other	.77	.86	**.81**	.73	.79	.76	.60	.80	.69

Table 7
SEnerCON: Using 43 parameters.

	RF			J48			SMO		
	P	R	F-M	P	R	F-M	P	R	F-M
Enhancement	.86	.77	**.81**	.75	.70	.72	.73	.52	.61
Other	.80	.88	**.84**	.73	.77	.75	.64	.82	.72

Table 8
SEnerCON: Using 34 parameters (no sentiment).

	RF			J48			SMO		
	P	R	F-M	P	R	F-M	P	R	F-M
Enhancement	.86	.76	**.81**	.75	.66	.70	.70	.46	.56
Other	.80	.88	**.84**	.71	.80	.75	.62	.82	.70

Table 9
SEnerCON: Using 25 parameters (no speech acts).

	RF			J48			SMO		
	P	R	F-M	P	R	F-M	P	R	F-M
Enhancement	.83	.77	**.80**	.73	.68	.70	.71	.46	.56
Other	.80	.85	**.83**	.71	.76	.74	.62	.82	.71

by applying the 43 parameters (*i.e.,* the *speech-acts*, sentiment parameters, and structure parameters) the F-Measure (F-M) for *Enhancement* is .81 and for *Other* is .84. In Table 5 we see that using only 34 parameters (excluding the sentiment parameters) the F-Measure for *Enhancement* is reduced to .80 and for *Other* to .82, while in Table 6, using 25 parameters (excluding the *speech-acts* parameters) the F-Measure for *Enhancement* is .79 and for *Other* .81. These results, compared to the conference version of this paper, have greatly improved. The best F-Measure values in the conference version are .68 for *Enhancement* and .74 for *Defect*.

The results for the SEnerCON dataset are shown in Tables 7, 8, 9. By using the 43 parameters, see Table 7, we have F-Measure of .81 for *Enhancement* and .84 for *Other* with the RF algorithm. The same results occur for the AOO dataset using the 43 parameters; and in Table 8 we see that the same results are obtained by using 34 parameters (no sentiment). If we remove the *speech-acts* parameters, it means using only 25 parameters (Table 9) we notice that the performance is .80 for *Enhancement* and .83 for *Other*, which is not a high impact. The other algorithms J48 and SMO perform lower than RF in all cases.

5.3. Correlation between speech-acts and importance of the feedback

In order to analyse the interaction, if any, between *speech-acts* and feedback type, as well as *speech-acts* and importance associated with the issue, we perform statistical tests of correlation on these variables. Furthermore, we perform correspondence analysis to further explore the bilateral interactions present. In particular, we analyse the interaction between *speech-acts* and issue *priority*. We focus on issue *priority* because it is assigned to the issues by the development team, in particular by managers or team leaders, according to the *severity* of the issue reported by the user (who created the issue) as well as the developers' assessment of the issue. Furthermore, other factors, such as resource availability, are also likely to be taken into consideration while assigning a particular priority to an issue. Hence, we assess how *speech-acts* interact with issue priority to see if any patterns emerge and could serve as indicators of overall issue importance. Towards this objective, we first perform a statistical test to establish significant dependence between *speech-acts* and priority. Then we analyse the association between individual *speech-act* types and priority levels.

5.3.1. Test of independence between speech-acts and issue priority

In order to assess the relation between *speech-acts* and issue *priority*, we performed the Chi-squared statistical test of independence, since both variables are categorical. The null hypothesis is formulated as follows:

Please cite this article as: I. Morales-Ramirez, F.M. Kifetew and A. Perini, Speech-acts based analysis for requirements discovery from online discussions, Information Systems (2019), https://doi.org/10.1016/j.is.2018.08.003.

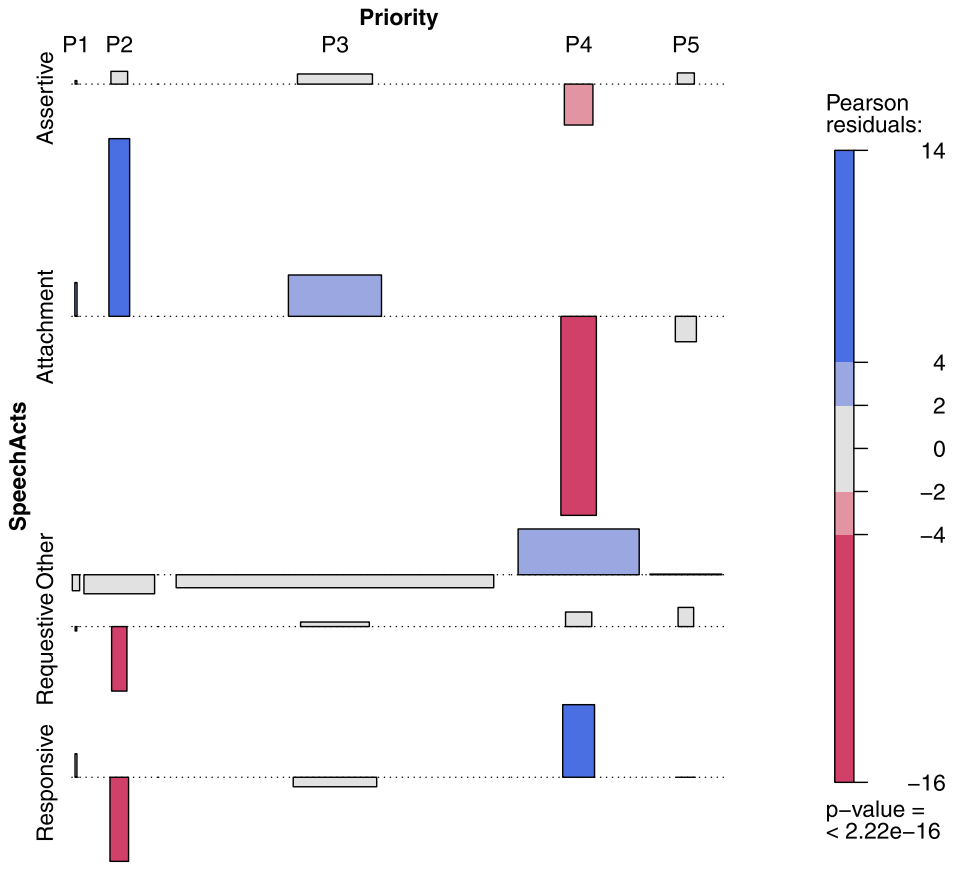

Fig. 6. Association interaction between *speech-acts* and issue priority in AOO dataset. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

H_0: ***speech-acts* of issues in AOO are independent of the priority assigned to the issues by AOO team.**

We constructed a contingency table from *speech-acts* and issue *priority* (shown in Table 10). The values shown in Table 10 are at sentence level, because the *speech-acts* are determined for each sentence. We then applied Pearson's Chi-square test in the R statistical package, with significance level of 0.05. Notice that the frequencies for priority level *P1* contain a few values below 5 (see Table 10), hence the computation of the Chi-square statistic is approximate. The *p*-value obtained from the Chi-square test is less than 2.2e−16, which is below the significance level of 0.05. Hence, we reject the null hypothesis which states that *speech-acts* and issue priority are independent.

5.3.2. Analysis of association between speech-acts and issue priority

The fact that null hypothesis in the previous section was rejected implies that *speech-acts* and issue *priority* do indeed interact. We now analyse the interactions among the various *speech-acts* and priority levels by means of association plots and Pearson residuals, according to what is suggested for categorical variables [30].

Fig. 6 shows the association plot between *speech-acts* and priority, generated using the `assoc` function in the R statistical package [31]. To enhance readability, the *speech-acts* shown in Fig. 6 are grouped following the *speech-act* categories shown in Table 1.

As can be seen from Fig. 6, the interactions among the various *speech-acts* and priority levels are visible. In the plot shown in Fig. 6, the shaded tiles represent deviations (residuals) from independence. The dimensions of the tiles are proportional to the absolute value of the residuals, *i.e.,* differences between observed and expected frequencies. The sign (positive or negative) depends on the direction of the difference, *i.e.,* whether the expected frequency was higher than the observed, and vice versa. Values (absolute) below 2 are not shaded, values below 4 are shaded in light colour, and values above 4 are shaded with full saturation. Positive residuals are shaded in blue, while negative residuals are shaded in red. The horizontal dotted lines indicate the line of independence. The association plot visually depicts the interactions between the *speech-act* types and priority levels. For instance, we can observe a strong interaction between the *speech-act* category *c-Attachment* and all priority levels, in particular priority levels *P2* and *P4*. Similarly, we can easily see that the *speech-act* category *c-Other* has meaningful interaction only with priority level *P4*, while the analysis category *c-Requestive* interacts with priority level *P2*.

6. Discussion

In the following, we discuss the main findings of the results and threats to validity.

The results with respect to *RQ1* show the likelihood that certain types of *speech-acts* are more frequent when reporting an *Enhancement* issue than the type *Other*. In particular, we notice from Fig. 3 that the distribution of *speech-acts* in the first comments submitted by users of the OpenOffice suite exhibit distinctive trends. Certain *speech-acts* (*Requestive*, *Requirement*, *Accept*) occur with high frequency in comments of type *Enhancement*, while other *speech-acts* (*CODE_LINE*, *Attachment*, *LOG_FILE*, *Informative*) occur with higher frequencies in comments of type *Other*. The rest of the *speech-acts* tend to be present in both types of comments with more or less similar frequencies. Keep

Table 10
Contingency table showing frequencies of the various *speech-acts* and issue priority (P1 is highest).

Analysis category	Individual SA/Priority	P1	P2	P3	P4	P5
c-Assertive	Assertive	9	724	13653	1795	687
	Concessive	0	56	1542	229	100
	Confirmative	0	17	449	84	35
c-Requestive	Requestive	0	14	504	78	32
	Requirement	0	124	4553	668	273
	Question	6	381	8052	1203	404
c-Responsive	Suggestive	3	302	8170	1295	390
	Suppositive	12	366	8567	1432	490
	Responsive	1	83	2422	372	108
c-Attachment	Attachment	13	720	18237	2058	746
	CODE_LINE	8	852	4429	361	336
	URL_LINK	1	56	1439	115	74
	LOG_FILE	0	34	343	32	3
c-Other	Informative	42	6583	108023	15531	5147
	Positive	42	2277	58405	8745	3239
	Negative	37	3645	87110	13106	4552
	Descriptive	7	877	17364	2655	941
	Thank	3	203	6211	988	348
	Reject	1	19	417	68	26
	Accept	0	15	626	133	48

in mind that the data used for plotting the *speech-acts* is not been balanced; the *Enhancement* feedback represents 32.16% of the total data.

It is interesting to observe that for the class *Enhancement* the occurrence of the *speech-acts Requestive*, *Requirement*, and *Accept* is higher. This could mean that community members tend to ask for more functionalities, while other participants agree through accepting (67%) or rejecting (47%) new ideas.

Worth noticing are the *speech-acts Informative* and *Descriptive*, in *Other* feedback, that seem to complement *speech-acts* such as *Attachment*, *CODE_LINE*, *URL_LINK* and *LOG_FILE*, with explanations and details of what is happening.

Albeit the *speech-act Reject* being a bit lower in the *Enhancement* category, we believe that both *Accept* and *Reject speech-acts* are more frequent in the *Enhancement* category than *Other*, attributable to discussions about new functionalities, or modifications. Proof of this is that even though the number of *Enhancement* feedback (6907) is far lower than the *Other* feedback (14,570), the percentage of appearance for the *speech-acts Accept* and *Reject* is high enough in *Enhancement*, 68% and 48% respectively.

It seems that when a problem is reported via an issue, the *speech-act Descriptive* is mandatory almost 70% of the time, together with details expressed through the *speech-acts Informative* 80% and *Attachment* with 90% of the time, and there is no need of further discussion.

We are aware that our interpretations are only based on assumptions, but trying to understand the attitude of stakeholders through the way they write their needs represents a step forward in our research.

There are other *speech-acts* that also emphasise no need for discussing a topic, for example by stating the *speech-act Assertive* "I have a problem with…" with the *speech-act Descriptive* "…when selecting…". We also see the presence of sentiments mainly in the feedback type *Other*, for example *Positive opinion* appears 62% of the time and *Negative opinion* 69% of the time.

The previous interpretations of the results give us a clear path of which are the *speech-acts* highlighting when there is a requirements-related piece of information that needs to be analysed. This is the kind of support we are looking forward to provide to the practitioners that deal with the tasks of reading and analysing online discussions. Therefore, when the identification of *speech-acts* such as *Requestive* and *Requirement* is made on textual content, we can extract those comments out of the discussion for a further analysis.

In answer to RQ1, we have identified that speech-acts exhibit distinctive trends in such a way that useful information regarding the discovery of new requirements can be inferred. Specifically, speech-acts Requestive, Requirement, Accept *are strongly present in issues requesting for new functionalities or enhancement of existing ones.*

Towards answering *RQ2* we observed that *speech-acts* are good candidates for serving as parameters when building a classifier that recognises online feedback as either *Enhancement* or *Other*, regardless the size of the dataset.

This is mainly due to the distinctive patterns we observed when analysing the distribution of the *speech-acts* as part of RQ1. In particular, we have observed that there are a core set of *speech-acts* that characterise each category of online feedback (shown at the left and right extremes of Fig. 3). Consequently, we have defined parameters, derived mainly from *speech-acts*, that could uniquely capture the characteristic of online feedback in such a way that a machine learning algorithm could be trained to classify such data. Our investigation further evidences that by augmenting the set of parameters derived from *speech-acts* with sentiment related parameters, we can achieve even better results.

Regarding the selection of the ML algorithms used in our research, we have tried different algorithms, as well as different settings, in order to identify the best performing algorithms for the classification. From all the algorithms available in Weka, we run tests using for example the classifiers under the categories of Bayes, functions, meta, and trees. Random Forest, J48 and SMO resulted with the best performances. Random Forest and J48 belong to the classifiers that use decision trees, while SMO is an algorithm that estimates a function.

The parameters used for training the algorithms, hence for the classification, were selected based on the acquired empirical knowledge about the emphasis that *speech-acts* can offer to one statement, the well known sentiment parameters (i.e., positive/negative adjectives) and default parameters used in other classification tasks, such as the number of nouns, adjectives, verbs. We could confirm that by using both parameters of sentiments and *speech-acts* a better classification is obtained.

In answer to RQ2, we can say that speech-acts can be used as parameters in building a classifier for online feedback into Enhancement *or* Other *category. When complemented with sentiment information, the classifier performance further improves.*

Please cite this article as: I. Morales-Ramirez, F.M. Kifetew and A. Perini, Speech-acts based analysis for requirements discovery from online discussions, Information Systems (2019), https://doi.org/10.1016/j.is.2018.08.003.

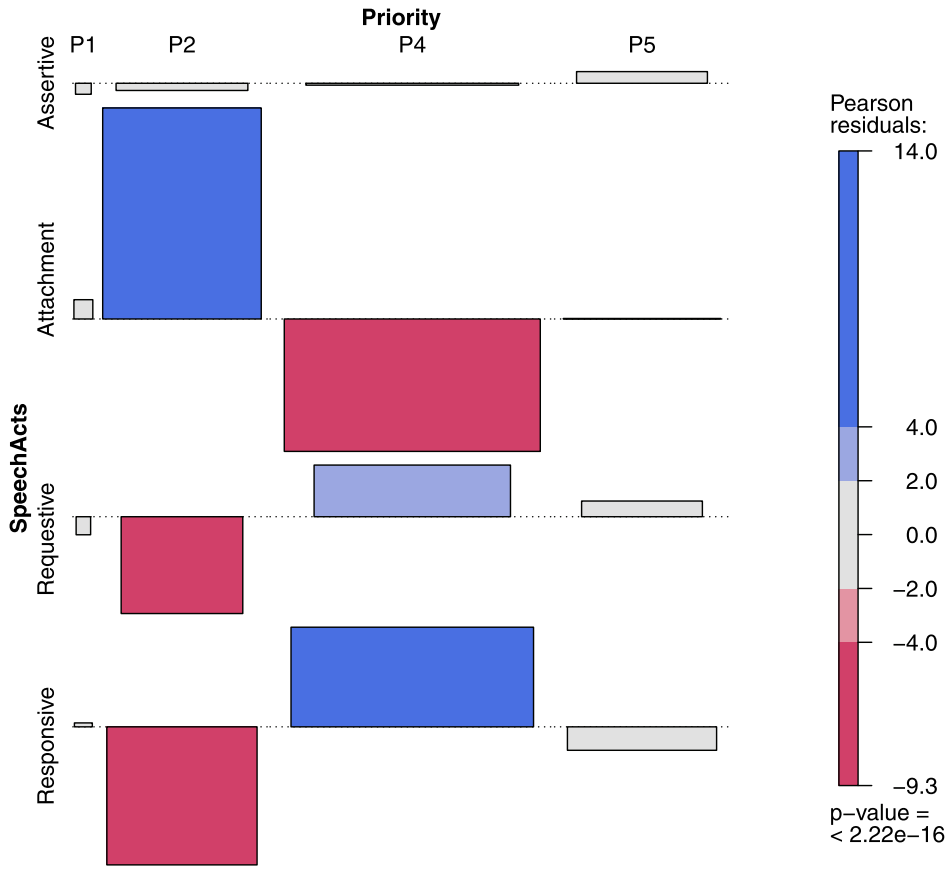

Fig. 7. Association interaction between *speech-acts* and issue priority in AOO dataset, removing the most common subset with priority *P1* and *speech-acts* category *c-Other*.

With respect to RQ3, the objective is to study the interaction between the various *speech-acts* and the priority assigned to the issue by experts. Such a study could ultimately be useful in determining the priority level of a user reported issue in an automated manner, hence minimising the manual intervention required by the expert. The first step towards this goal would be to establish if there exists an interaction between these two variables, i.e., the set of *speech-acts* and the *priority* of online discussions. The Chi-squared test of independence we conducted showed that the two variables are not independent. Meaning that these two variables do interact in some way. However, further investigation is required to understand how they interact.

Since both variables are categorical, i.e., assume values from a set of discrete valued categories, we adopted the use of association plots to get insight into the interaction between *speech-acts* and priority, as presented in Section 5.3.2. In particular, we observe that the majority of the entries in the dataset are assigned to priority level of *P3* (which corresponds to medium priority level) and they exhibit *speech-acts* of the group *c-Other* (see Table 10). This accounts for 63% of the entire dataset. This means that the bulk of the online discussions are of medium (normal) priority and contain commonly occurring *speech-acts* (such as *Informative*), which overshadows and dwarfs the other categories representing more important categories. In order to get better insight, we plot again the association, by removing this common category. The new plot is shown in Fig. 7. The plot with the full dataset is shown in Fig. 6 in Section 5.3.2.

From Fig. 7 we can observe that interactions between *speech-acts* and priority is a bit more visible. In particular, we can see that the *speech-acts Attachment*, *Requestive*, and *Responsive* interact strongly with the priority levels *P2* and *P4*. Specifically, the *speech-act Attachment* is observed with much more frequency, than that expected under independence, in issues with priority level *P2*, while it is observed less than expected in issues with priority level *P4*. On the other hand, the *speech-acts Requestive* and *Responsive* both show similar patterns in which they are observed less than expected in issues of priority *P2*, while their observed frequency is higher than the expected in issue of priority *P4*. For priority levels *P1*, *P3*, and *P5* the Pearson residuals show that the observed deviations from independence are low, hence the interactions do not seem to be significant. However, it is important to note here that in our dataset the number if issues with priority level *P1* and *P5* is quite low with respect to the others. Hence, the observations could possibly change with a dataset in which the number of instances in these categories are significantly different from those in ours.

> *In answer to RQ3, we can conclude that there are significant interactions between speech-acts and issue* priority, *which is a proxy to the issue's* importance. *In particular speech-acts categories* c-Attachment, c-Requestive, *and* c-Responsive *seem to associate strongly with priorities* P2, P4, *while the other speech-acts seem to have minor to none associations with priority.*

6.1. Threats to validity

In this section we discuss the main threats to validity [32] that concern our work. *Conclusion validity* threats concern issues that affect the ability to draw the correct conclusion on the observed phenomenon. The results reported in this work give a positive usage of *speech-acts* towards identifying requirements relevant information from online discussions in AOO and end

user feedback in SEnerCON. To mitigate potential conclusion validity threats, we have used a relatively large dataset and drawn conclusions supported by appropriate statistical tests.

Internal validity threats concern the possible confounding elements that may hinder a well performed experiment. Our *speech-act* based analysis rests on rules which are not extensive but have been improved and the dataset has been already labelled by the OpenOffice community, which avoids potential bias in assigning the categories issues as Other or Enhancement. Another threat concerns the selection of weights for the parameters to train the classification algorithms, due to the fact that for this work the weights have been assigned based on our observations and not in an automatic way. The decision of using oversampling is also another source of threat, since we had an unbalanced dataset, however oversampling has shown to be a rigorous scientific technique to overcome such problems in data science. The selection of the algorithms to train might count as another threat, because even if we experimented with several algorithms before selecting the best performing ones, we cannot be exhaustive enough to declare that there are no other better classifiers.

Construct validity threats concern the relationship between theory and observation. So far there is no theory explaining any correlation between *speech-acts* and categories of issues reported in online discussions related to software applications. Our technique of analysis represents a first hypothesis of such relationship. We have carefully assessed the rules developed and assessed their effectiveness via manual inspection of selected entries, but this is not a warranty that all the *speech-acts* annotations were performed correctly by the tool.

The *speech-acts* are refined concepts of what senders elaborate in their minds to convey their needs and influence or persuade a receiver to do something. The scope of our research is not evaluating this part because measuring this is not straightforward due to subjectivity. But our attempt is to find correlations between *speech-acts* and the property *priority* so that urgent topics of online discussions could be handled with the appropriate urgency (priority).

External validity threats concern extending the validity of observations outside the experimental context. While we performed the main analysis on the AOO dataset, which we extended with respect to the conference version of this paper, we also applied our analysis technique on an additional dataset of user feedback data from SEnerCON, obtaining results consistent with that of AOO. Applying the approach on further industrial datasets will increase confidence with respect to its external validity.

7. Related works

Research on automated analysis techniques of online user feedback at support of software engineering tasks has increased significantly in the last years. In this section, we focus on studies that concern the analysis of online user feedback, which is expressed using NL textual messages, for software evolution and maintenance purposes. We group them depending on the type of online user feedback they consider, namely app reviews, tweets, and online discussions, which include forum and mailing lists.

App reviews are characterised by a timestamp showing when the review was created, a user rating, and a textual comment, which, especially in case of mobile app is usually shorter than comments in user forum and mailing lists.

There are commercial tools, like App Annie [33] that supports decision making based on collected information about number of downloads, revenue, rating, usage and other characteristics about apps, with the goal of understanding how to develop a successful app. Research on mobile app review analytics is receiving a lot of attention since 2012, as reported by Martin et al. [2], which presents a survey of the research on App Store analysis for software engineering in the period 2010–2015.

NLP techniques are applied to filter out irrelevant information in app reviews, moreover topic modelling, sentiment analysis and machine learning are among the techniques that have been exploited to classify user comments into bug reports, feature requests, or polarity of sentiments, e.g. [7,9,34]. Semantic frames are used to generate lower dimensional and more accurate models in comparison to text classification methods by Jha et al. [35]. Di Sorbo et al. [36], define a two level classification model which considers the user's intention and the review topic. Moreover, they propose a summariser called SURF that automatically extracts topics, classifies the intention and group sentences covering the extracted topics for recommending developers which software changes to implement. The research work by Keertipati et al. [37] uses four attributes to be exploited to prioritise feedback, i.e. frequency of a feature, rating, emotions and deontics. They propose three prioritisation approaches: (1) individual attribute-based — when ranking features based on their frequency, the ratings are not considered; (2) weighted approach — enables the combination of two or more attributes in the prioritisation; (3) regression-based approach and data-driven approach to examine influential variables for determining the severity of reviews.

App review analytics can provide support to developers when deciding about maintenance tasks [8], or (semi-)automated release planning tools [9–11]. Concerning techniques at support of release planning in app development, worth mentioning is the work presented in [9] that aims at analysing comments from users of software applications. Moreover, user feedback trend analysis have been also investigated since reviews can be associated to app versions, with the purpose to support developers prioritising issues [38], and identifying the occurrence of serious issues [38,39]. The work of Guzman et al. [40] presents an approach called DIVERSE that aims at recognising the diversity of opinions on a set of App reviews. Moreover, this approach also helps developers and analysts recognise conflicting opinions regarding a feature. The evaluation is performed on a dataset of 170,829 App reviews and a truth set of 2800 manually labelled reviews.

Tweets are unstructured, short (less than 140 characters) textual messages which are posted on *Twitter*.

The relevance for software engineering tasks of tweets talking about software applications has been investigated in [12–14]. The huge flow of tweets, together with the fact that the proportion of the contained information that is relevant for software engineers is small compared to the overall volume of tweets [14], is motivating the development of automated filtering and classification techniques for the exploitation of such online data for requirements engineering purposes. Guzman et al. [15], exploit ML techniques for automatically classifying tweets requesting improvements. Semantically related tweets are grouped by using topic modelling, and ranked according to a weighted function defined in terms of specific attributes, such as content category, sentiment and number of retweets. Similarly, Williams et al. [41] exploit Support Vector Machines and Naive Bayes to categorise technically informative tweets. Additionally, multiple summarisation strategies are proposed for generating meaningful summaries of informative software-relevant tweets.

Differently from the above mentioned works, we focus on online discussions that occur in forum, issue tracking systems and mailing lists, but with a similar purpose, that is of deriving requirements-related information. This type of online user feedback usually contains longer, unstructured text, and can develop as conversation threads, where messages refer to previous ones in a thread.

Please cite this article as: I. Morales-Ramirez, F.M. Kifetew and A. Perini, Speech-acts based analysis for requirements discovery from online discussions, Information Systems (2019), https://doi.org/10.1016/j.is.2018.08.003.

Among the research work which investigate this form of online feedback for requirements engineers purposes, worth to be mentioned are the followings. Kanchev et al. [42], propose a systematic approach for extracting and querying online discussions. It rests on a combination of human and artificial intelligence techniques, that is crowdsourcing is exploited to annotate online discussion. A tool, called DECA, for the analysis of emails in mailing list is proposed by Di Sorbo et al. [43]. Its purpose is that of identifying information useful to developers for specific maintenance tasks. Results from the application of DECA to the analysis of mailing lists related to Qt and Ubuntu projects provide experimental evidence of its effectiveness in terms of precision and recall.

8. Conclusion and future works

In this paper we presented a technique for the analysis of online discussions, which involve users of software applications who can post their messages in user forums, mailing-lists, wikis, newsgroups, and blogs. The proposed technique aims at supporting developers to extract requirements-relevant information that helps them prioritise users' needs, in the context of software evolution and maintenance tasks. This technique is based on a linguistic technique called *speech-acts* based analysis, which we introduced in previous work [20,21].

We have revised and improved our *speech-acts* based analysis technique, and characterised its properties by executing a set of experiments on two different datasets, namely a dataset taken from the Apache OpenOffice project, which contains 161,120 textual comments, and a dataset containing 575 feedback messages provided by users of a software application in the home energy management domain, called iESA.

We have found that there is a potential association between types of *speech-acts* (e.g. *Informative*, *Responsive*, *Requestive*, etc.) and categories of issues (e.g. Enhancement, Other). Therefore, some assumptions of combinations of *speech-acts* can be implied based on the type of issue discussed by the stakeholders.

The experimental results provided evidences that certain types of *speech-acts* are more used when reporting an Enhancement request rather than Other type of requests. For instance, the *speech-act Attachment* and *CODE_LINE* resulted to be used in 90% of the messages expressing Other type of request, while in Enhancement requests the *speech-act Requestive* is used in 80% of the cases.

We used the *speech-acts* and the sentiment as parameters for training three machine learning algorithms (Random Forest, J48 and SMO) and classified comments into Enhancement, and Other categories, improving the precision and recall previously obtained, which we reported in [21]. Specifically, the computed F-measure for the Enhancement category ranges between .79 to .81 for the AOO dataset, and in the new dataset from iESA the computed F-measure ranges between .80 and .81.

Furthermore, results from the experiments executed on the AOO dataset provide evidence of the existence of correlation between *speech-acts* and issue priority, in particular for the case of analysis category c-Attachment and issue priority level P2 on a 5-level scale where priority P1 means highest priority. How to exploit this correlation for supporting developers in issue prioritisation deserves further investigation.

In future work, we will apply our analysis on a larger dataset of user feedback from the SEnerCon's iESA project and validate findings with the company's development team. Moreover, we plan to investigate other correlations between *speech-acts* and characteristics of the OpenOffice dataset besides importance, such as severity. Finally, we intend to exploit the proposed *speech-acts* based analysis technique into a multi-criteria requirements prioritisation tool to guide the selection of the requirements to be prioritised on the basis of user feedback properties such as intention, sentiment and severity [44].

Acknowledgements

This work is a result of the SUPERSEDE project, funded by the H2020 EU Framework Programme under agreement number 644018. The first author is partially funded by INFOTEC under the project "081-022-00-FORTALECIMIENTO E INVERSIÓN".

Appendix A. Technical details: tools

GATE [26] is a Java suite of tools (or library), developed by the University of Sheffield in UK, for building and deploying software components to process human language. GATE can support a wide range of NLP tasks for information extraction.

Information extraction refers to the extraction of relevant information from unstructured text, such as entities and relationships between them, thus providing facts to feed a knowledge base [45].

GATE is an object library that allows the processing of language for different purposes, in our case for supporting our *speech-acts* based analysis approach. With this library one can manipulate different types of file formats (e.g., PDF, RTF, HTML) and perform coreference, entity recognition, part-of-speech tagging and other more complex tasks using ontologies.[16] The version we are using is the GATE Embedded, see Fig. A.8 for details of the components of GATE.[17]

Embedded is a programmers' tool and is delivered as a set of Java archives (JARs). It is used natively in Java, Groovy or other JVM-based languages, and via JNI or similar gateways from other languages. If you are not a programmer, look at Developer or Teamware instead.

GATE is widely used both in research and application work in different fields (*e.g.,* cancer research, web mining, law, etc.). The tool is composed of three main components for performing language processing tasks, namely (1) the *Language Resources* component that represents entities such as lexicons, corpora or ontologies; (2) the *Processing Resources* component, which contains a library of executable procedures, such as parsers, generators or n-gram modellers; and (3) the *Visual Resources* component that provides visualisation and editing functions that are used in GUIs.

Appendix B. Plots

In the following, we show all the *speech-acts* and their distribution along the first ten comments for all the 21,477 threads of the online discussions.

Figs. B.9, B.10, B.11, B.12 show different combinations of *speech-acts* that are present in online discussions whose issue type is *Enhancement*. Each plot has an uppercase letter on the bottom to indicate it is a subset of the *speech-acts*, it means that subset A in the plots of the Enhancement comments corresponds to the same subset A of the Other comments.

In Fig. B.9, it is worth to notice how the initial comment (order equal to zero) contains more *Negative* opinions followed by *Informative* and *Positive* opinions. Then the sentiments decrease in the immediate comment, while there is a peak with a distribution of 7% for *Attachment*, 13% for *Positive opinion*, 17% for *Negative opinion*, and 41% for *Informative*. After comment number 1 the *speech-act Attachment* decreases but maintain the same trend of 2% approximately, while the *speech-act Descriptive* only has a relevant appearance in the comment 0 with a 6%. The sentiments negative and positive have a high presence in the first comment >20% but after that, they decrease and later the *Negative opinion*

[16] The library can be downloaded here https://gate.ac.uk/download/.

[17] Taken from https://gate.ac.uk/family/embedded.html.

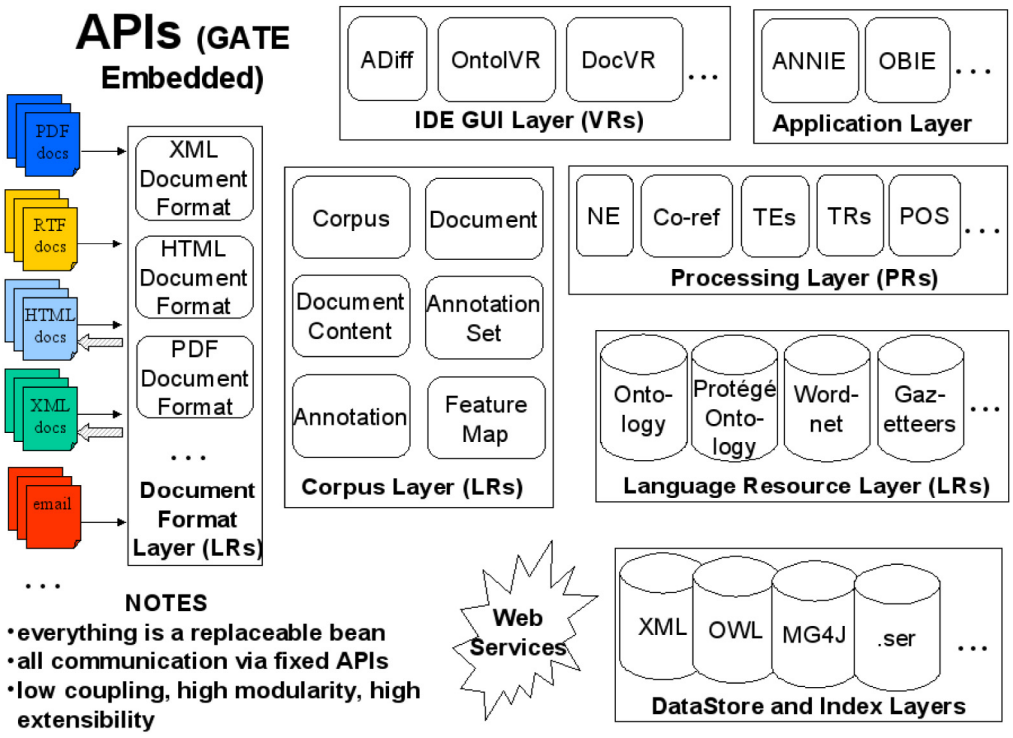

Fig. A.8. APIs that compose GATE Embedded.

goes up and overpasses the *Positive* and *Informative*. We can observe in Fig. B.10 that the *speech-act Requirement* has a frequency of 4.4% in the comment 0, then it drops in the comment 1 but then it goes up incrementally from comment 2 to 10 with a highest increment in comment 10 of 2.4%. The other *speech-acts* remain below 1.5% approximately. Fig. B.11 shows how the *speech-act Assertive* has a presence of 4.4% in the comment 0, and the *speech-act Suppositive* 3%. Then, there is a decrement for *Assertive* and comment 2 shows an increment for *speech-act Suggestive* of 5%.

Fig. B.12 shows how the *speech-acts* that are more characteristic of describing feedback labelled as Other, have a less important appearance in the online discussions labelled as Enhancement, less than 1% of occurrence.

Figs. B.13, B.14, B.15, B.16 show different combinations of *speech-acts* that are present in online discussions whose issue type is *Other*. The graphs aim at presenting to the reader how is the behaviour of occurrence of *speech-acts* along a discussion.

In the plot in Fig. B.13 we observe that the *speech-act Informative* has a frequency of 37% in the comment 0, while the *Negative* opinion has a 24%, the *Positive* opinion a 17% and the *Descriptive* a 7%. In the comment 1 the *speech-act Attachment* goes up to 20% and then decreases. *Negative* opinion, *Positive* opinion, and *Descriptive* decrease in comment 1, but then the *Negative* opinion increases, while *Informative* decreases. We can notice that the *speech-acts Positive* opinion, *Descriptive*, and *Attachment* maintain the same trend.

The *speech-acts* in Fig. B.14 have a frequency in comments less than 1%, which is almost the same behaviour in feedback type *Enhancement*, with an exception of the *speech-act Requirement*.

Fig. B.15 shows how most of the *speech-acts* are below 3%, while the plot C for *Enhancement*, most of the *speech-acts* are between 2% and 4%. The only *speech-act* that presents an increment starting in comment 2 is *Assertive*. The last plot B.15 presents a similar behaviour like plot B for *Enhancement*, but showing how the *speech-act Code line* stands out over the other *speech-acts* (similar effect for the *speech-act Requirement* in the opposite plot).

Appendix C. Replication package

In this section we provide a brief description of the replication package we make available as an external resource. The replication package contains the raw data from Apache OpenOffice (AOO) issue tracker, the rules we have developed for identifying the various *speech-acts*, the datasets we produced by applying our approach and using the parameters we defined for training classifiers in Weka.

The replication package is available at the following link: http://se.fbk.eu/technologies/speech-acts-based-analysis.

C.1. Raw data

The package contains all the raw data (first comments in plain text format) which we collected from the AOO issue tracking system (`AOOCommentsOALL.zip`). We also include a SQL dump which can be directly imported into a MySQL database for further structured analysis (`ApacheOpenOffice.sql`). For privacy reasons, we do not include the raw data from SEnerCON.

C.2. Gazetters and JAPE rules

We also provide the rules we developed for identifying the various *speech-acts* in the raw textual data (`GazetteersAndJAPE.zip`).

C.3. Datasets for Weka

Finally, the package contains the datasets which contains the sentences in terms of vectors codified according to the proposed set of parameters. The datasets are in Weka's ARFF file format (files with '.arff' extensions), ready to be directly fed into the Weka tool and train classifier models. These datasets are for both AOO and SEnerCON, as they are vectors of number and do not divulge any sensitive data. We include the data applying SMOTE (files names with '_SMOTE' suffix).

Please cite this article as: I. Morales-Ramirez, F.M. Kifetew and A. Perini, Speech-acts based analysis for requirements discovery from online discussions, Information Systems (2019), https://doi.org/10.1016/j.is.2018.08.003.

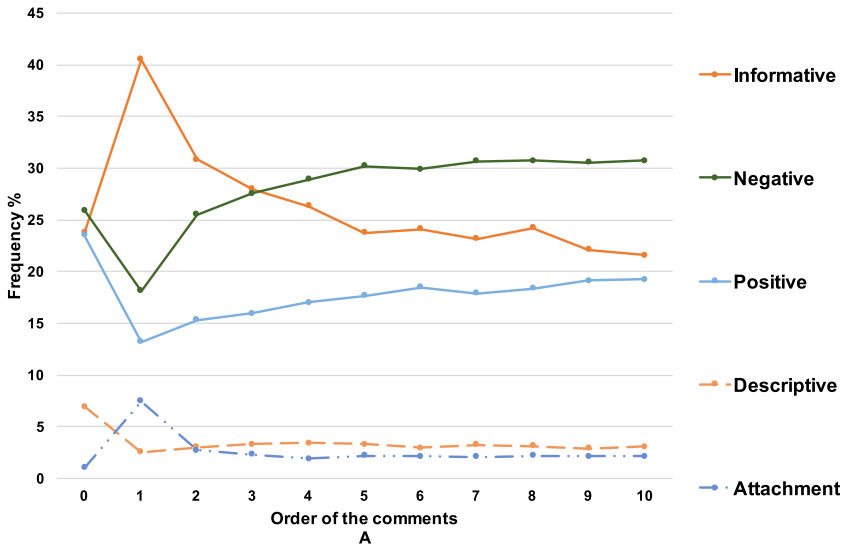

Fig. B.9. Distribution of *speech-acts* by the order of the comments, for feedback type Enhancement.

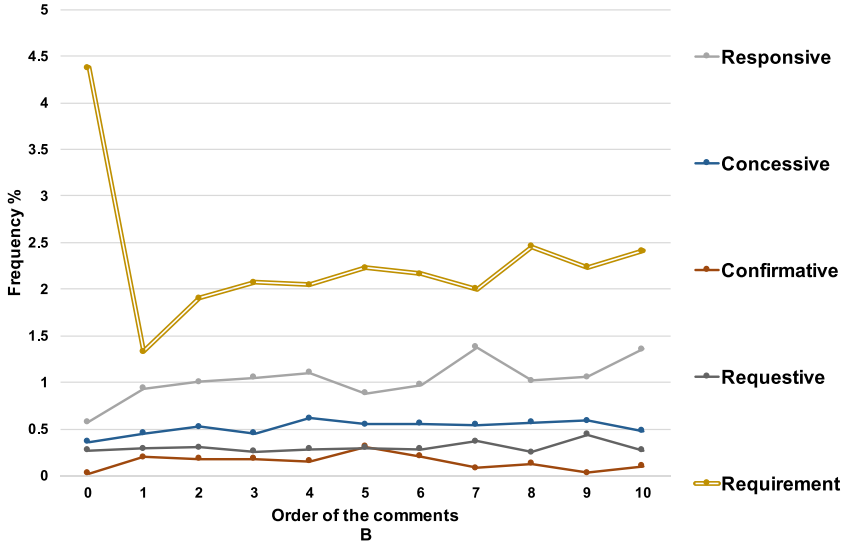

Fig. B.10. Distribution of *speech-acts* by the order of the comments, for feedback type Enhancement.

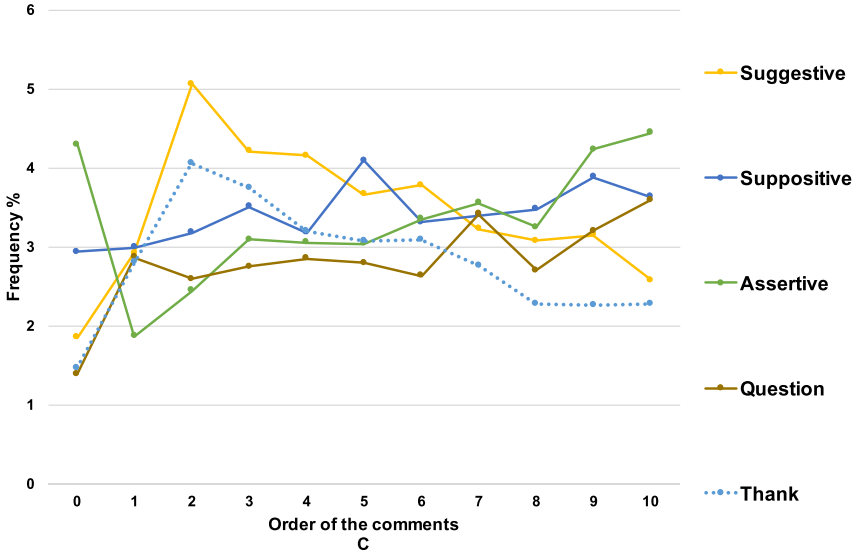

Fig. B.11. Distribution of *speech-acts* by the order of the comments, for feedback type Enhancement.

Please cite this article as: I. Morales-Ramirez, F.M. Kifetew and A. Perini, Speech-acts based analysis for requirements discovery from online discussions, Information Systems (2019), https://doi.org/10.1016/j.is.2018.08.003.

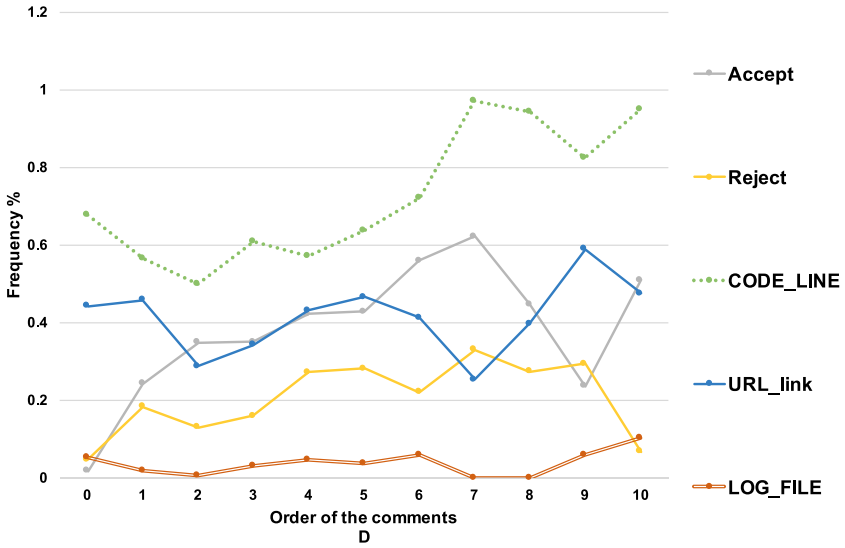

Fig. B.12. Distribution of *speech-acts* by the order of the comments, for feedback type Enhancement.

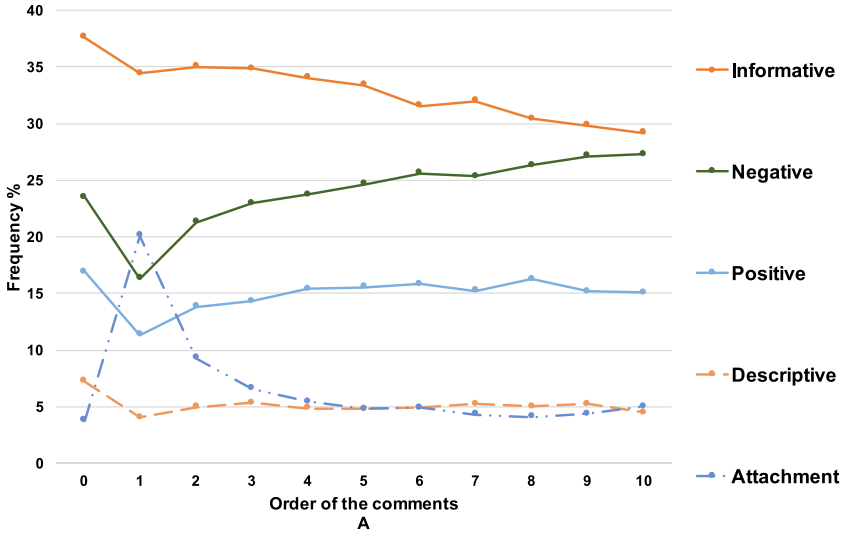

Fig. B.13. Distribution of *speech-acts* by the order of the comments, for feedback type Other.

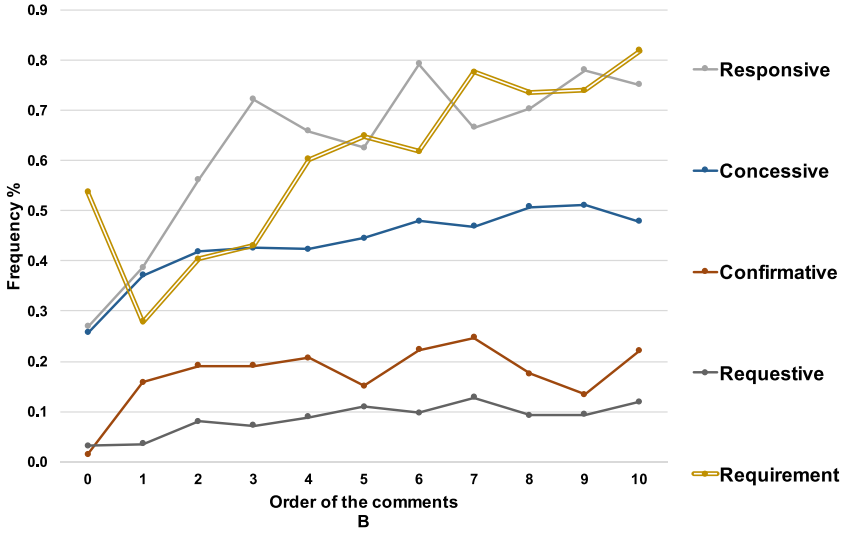

Fig. B.14. Distribution of *speech-acts* by the order of the comments, for feedback type Other.

Please cite this article as: I. Morales-Ramirez, F.M. Kifetew and A. Perini, Speech-acts based analysis for requirements discovery from online discussions, Information Systems (2019), https://doi.org/10.1016/j.is.2018.08.003.

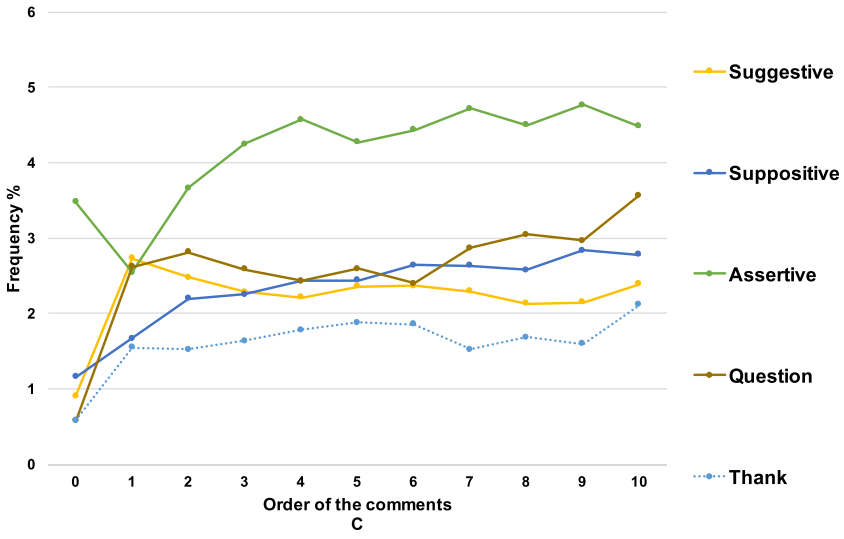

Fig. B.15. Distribution of *speech-acts* by the order of the comments, for feedback type Other.

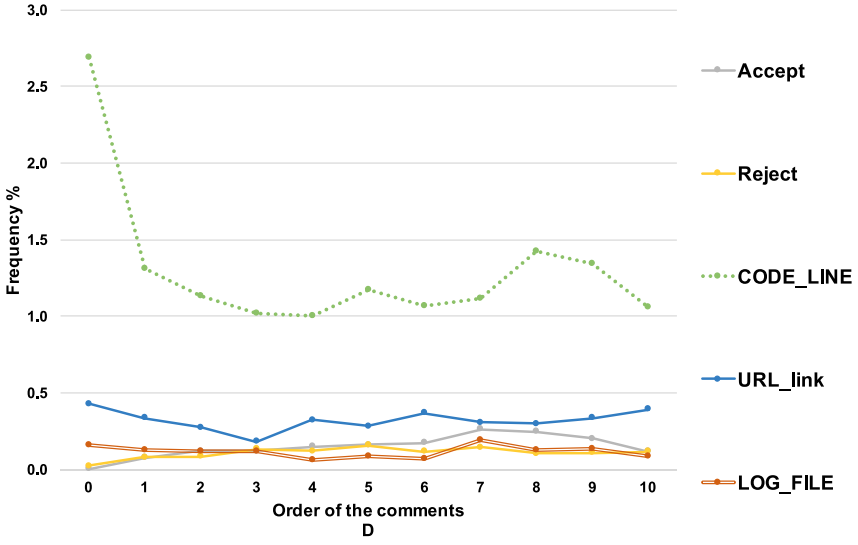

Fig. B.16. Distribution of *speech-acts* by the order of the comments, for feedback type Other.

References

[1] M. Kim, T. Zimmermann, R. DeLine, A. Begel, Data scientists in software teams: state of the art and challenges, IEEE Trans. Softw. Eng. (2017) 17, http://dx.doi.org/10.1109/TSE.2017.2754374.
[2] W. Martin, F. Sarro, Y. Jia, Y. Zhang, M. Harman, A survey of app store analysis for software engineering, IEEE Trans. Softw. Eng. 43 (9) (2017) 817–847, http://dx.doi.org/10.1109/TSE.2016.2630689.
[3] M. Kim, T. Zimmermann, R. DeLine, A. Begel, The emerging role of data scientists on software development teams, in: Proceedings of the 38th International Conference on Software Engineering, ICSE, ACM, 2016, pp. 96–107, http://dx.doi.org/10.1145/2884781.2884783.
[4] E.C. Groen, N. Seyff, R. Ali, F. Dalpiaz, J. Dörr, E. Guzman, M. Hosseini, J. Marco, M. Oriol, A. Perini, M.J.C. Stade, The crowd in requirements engineering: the landscape and challenges, IEEE Softw. 34 (2) (2017) 44–52, http://dx.doi.org/10.1109/MS.2017.33.
[5] I. Morales-Ramirez, A. Perini, R.S.S. Guizzardi, An ontology of online user feedback in software engineering, J. Appl. Ontol. 10 (3–4) (2015) 297–330, http://dx.doi.org/10.3233/AO-150150.
[6] C.D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S.J. Bethard, D. McClosky, The stanford CoreNLP natural language processing toolkit, in: Association for Computational Linguistics (ACL) System Demonstrations, 2014, pp. 55–60, http://dx.doi.org/10.3115/v1/P14-5010.
[7] W. Maalej, H. Nabil, Bug report, feature request, or simply praise? On automatically classifying app reviews, in: Proceedings of the 23rd International Requirements Engineering Conference, RE, IEEE, 2015, pp. 116–125, http://dx.doi.org/10.1109/RE.2015.7320414.
[8] S. Panichella, A. Di Sorbo, E. Guzman, C.A. Visaggio, G. Canfora, H.C. Gall, How can I improve my app? Classifying user reviews for software maintenance and evolution, in: Proceedings of the International Conference on Software Maintenance and Evolution, ICSME, IEEE, 2015, pp. 281–290, http://dx.doi.org/10.1109/ICSM.2015.7332474.
[9] L.V.G. Carreño, K. Winbladh, Analysis of user comments: an approach for software requirements evolution, in: D. Notkin, B.H.C. Cheng, K. Pohl (Eds.), Proceedings of the 35th International Conference on Software Engineering, ICSE, IEEE, 2013, pp. 582–591, http://dx.doi.org/10.1109/ICSE.2013.6606604.
[10] L. Villarroel, G. Bavota, B. Russo, R. Oliveto, M.D. Penta, Release planning of mobile apps based on user reviews, in: Proceedings of the 38th International Conference on Software Engineering, ICSE, ACM, 2016, pp. 14–24, http://dx.doi.org/10.1145/2884781.2884818.
[11] A. Ciurumelea, A. Schaufelbuhl, S. Panichella, H.C. Gall, Analyzing reviews and code of mobile apps for better release planning, in: Proceedings of the 24th International Conference on Software Analysis, Evolution and Reengineering, SANER, IEEE, 2017, pp. 91–102, http://dx.doi.org/10.1109/SANER.2017.7884612.
[12] Y. Tian, P. Achananuparp, I.N. Lubis, D. Lo, E. Lim, What does software engineering community microblog about? in: Proceedings of the 9th Working Conference on Mining Software Repositories, MSR, IEEE, 2012, pp. 247–250, http://dx.doi.org/10.1109/MSR.2012.6224287.
[13] L. Singer, F.M.F. Filho, M.D. Storey, Software engineering at the speed of light: how developers stay current using twitter, in: Proceedings of the 36th International Conference on Software Engineering, ICSE, ACM, 2014, pp. 211–221, http://dx.doi.org/10.1145/2568225.2568305.

[14] E. Guzman, R. Alkadhij, N. Seyff, A needle in a haystack: What do twitter users say about software? in: Proceedings of the 24th International Conference on Requirements Engineering, RE, IEEE, 2016, pp. 96–105, http://dx.doi.org/10.1109/RE.2016.67.

[15] E. Guzman, R. Alkadhi, N. Seyff, An exploratory study of twitter messages about software applications, J. Requir. Eng. 22 (3) (2017) 387–412, http://dx.doi.org/10.1007/s00766-017-0274-x.

[16] J.R. Searle, Speech Acts: An Essay in the Philosophy of Language, vol. 626, Cambridge University Press, 1969.

[17] C. Caleb, D. Schrock, P. Dauterman, Speech act analysis within social network sites' status messages, in: 59th International Communication Association Conference, vol. 20, 2009.

[18] D. Feng, E. Shaw, J. Kim, E.H. Hovy, An intelligent discussion-bot for answering student queries in threaded discussions, in: International Conference on Intelligent User Interfaces, ACM, 2006, pp. 171–177, http://dx.doi.org/10.1145/1111449.1111488.

[19] J. Kim, G. Chern, D. Feng, E. Shaw, E. Hovy, Mining and assessing discussions on the web through speech act analysis, in: Proceedings of the Workshop on Web Content Mining with Human Language Technologies, 2006, URL https://pdfs.semanticscholar.org/1fdb/5205e6e2da07e3d81f6ad3177ca299eb74f7.pdf.

[20] I. Morales-Ramirez, A. Perini, Discovering speech acts in online discussions: a tool-supported method, in: Joint Proceedings of the CAiSE 2014 Forum, in: CEUR Workshop Proceedings, vol. 1164, CEUR-WS.org, 2014, pp. 137–144, URL http://ceur-ws.org/Vol-1164/PaperDemo01.pdf.

[21] I. Morales-Ramirez, F.M. Kifetew, A. Perini, Analysis of online discussions in support of requirements discovery, in: Proceedings of the 29th International Conference on Advanced Information Systems Engineering, CAiSE, Springer, 2017, pp. 159–174, http://dx.doi.org/10.1007/978-3-319-59536-8_11.

[22] I. Morales-Ramirez, A. Perini, M. Ceccato, Towards supporting the analysis of online discussions in OSS communities: a speech-act based approach, in: Information Systems Engineering in Complex Environments - CAiSE Forum Selected Extended Papers, in: LNBIP, vol. 204, Springer, 2014, pp. 215–232, http://dx.doi.org/10.1007/978-3-319-19270-3_14.

[23] A. Stolcke, N. Coccaro, R. Bates, P. Taylor, C. Van Ess-Dykema, K. Ries, E. Shriberg, D. Jurafsky, R. Martin, M. Meteer, Dialogue act modeling for automatic tagging and recognition of conversational speech, J. Comput. Linguist. 26 (3) (2000) 339–373, http://dx.doi.org/10.1162/089120100561737.

[24] N. Novielli, C. Strapparava, Dialogue act classification exploiting lexical semantics, in: Conversational Agents and Natural Language Interaction: Techniques and Effective Practices, IGI Global, 2011, pp. 80–106, http://dx.doi.org/10.4018/978-1-60960-617-6.ch004.

[25] K. Bach, R.M. Harnish, Linguistic Communication and Speech Acts, MIT Press, Cambridge, MA, 1979.

[26] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, N. Aswani, I. Roberts, G. Gorrell, A. Funk, A. Roberts, D. Damljanovic, T. Heitz, M.A. Greenwood, H. Saggion, J. Petrak, Y. Li, W. Peters, Text Processing with GATE (Version 6), 2011.

[27] S. Marciuska, C. Gencel, P. Abrahamsson, Automated feature identification in web applications, in: International Conference on Software Quality, Springer, 2014, pp. 100–114.

[28] E. Guzman, M. Ibrahim, M. Glinz, Prioritizing user feedback from twitter: a survey report, in: Proceedings of the 4th International Workshop on CrowdSourcing in Software Engineering, CSI-SE@ICSE, IEEE/ACM, 2017, pp. 21–24, http://dx.doi.org/10.1109/CSI-SE.2017.4.

[29] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, Smote: synthetic minority over-sampling technique, J. Artificial Intelligence Res. 16 (1) (2002) 321–357, URL https://www.jair.org/media/953/live-953-2037-jair.pdf.

[30] M. Friendly, Visualizing Categorical Data, first ed., SAS Publishing, 2001.

[31] D. Meyer, A. Zeileis, K. Hornik, Visualizing independence using extended association plots, in: Proceedings of 3rd International Workshop on Distributed Statistical Computing, DSC, 2003, URL https://www.r-project.org/conferences/DSC-2003/Proceedings/MeyerEtAl.pdf.

[32] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell, A. Wesslén, Experimentation in Software Engineering: An Introduction, Kluwer Academic Publishers, Norwell, MA, USA, 2000.

[33] App Annie, The App Annie Platform, 2018, https://www.appannie.com/en/.

[34] N. Chen, J. Lin, S.C.H. Hoi, X. Xiao, B. Zhang, AR-miner: mining informative reviews for developers from mobile app marketplace, in: Proceedings of the 36th International Conference on Software Engineering, ICSE, ACM, 2014, pp. 767–778, http://dx.doi.org/10.1145/2568225.2568263.

[35] N. Jha, A. Mahmoud, Mining user requirements from application store reviews using frame semantics, in: Proceedings of the 23rd International Working Conference on Requirements Engineering: Foundation for Software Quality, REFSQ, 2017, pp. 273–287, http://dx.doi.org/10.1007/978-3-319-54045-0_20.

[36] A. Di Sorbo, S. Panichella, C.V. Alexandru, J. Shimagaki, C.A. Visaggio, G. Canfora, H.C. Gall, What would users change in my app? Summarizing app reviews for recommending software changes, in: Proceedings of the 24th ACM SIGSOFT International Symposium FSE, ACM, 2016, pp. 499–510, http://dx.doi.org/10.1145/2950290.2950299.

[37] S. Keertipati, B.T.R. Savarimuthu, S.A. Licorish, Approaches for prioritizing feature improvements extracted from app reviews, in: Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineeringon Evaluation and Assessment in Software Engineering, ACM, 2016, pp. 33:1–33:6, http://dx.doi.org/10.1145/2915970.2916003.

[38] C. Gao, B. Wang, P. He, J. Zhu, Y. Zhou, M.R. Lyu, Paid: prioritizing app issues for developers by tracking user reviews over versions, in: Proceeding of the 26th International Symposium on Software Reliability Engineering, ISSRE, IEEE, 2015, pp. 35–45, http://dx.doi.org/10.1109/ISSRE.2015.7381797.

[39] P.M. Vu, T.T. Nguyen, H.V. Pham, T.T. Nguyen, Mining user opinions in mobile app reviews: A keyword-based approach (T), in: Proceedings of the 30th International Conference on Automated Software Engineering, ASE, IEEE/ACM, 2015, pp. 749–759, http://dx.doi.org/10.1109/ASE.2015.85.

[40] E. Guzman, O. Aly, B. Bruegge, Retrieving diverse opinions from app reviews, in: International Symposium on Empirical Software Engineering and Measurement, ESEM, 2015, pp. 1–10, http://dx.doi.org/10.1109/ESEM.2015.7321214.

[41] G. Williams, A. Mahmoud, Mining twitter feeds for software user requirements, in: Proceedings of the 25th International Conference on Requirements Engineering, RE, IEEE, 2017, pp. 1–10, http://dx.doi.org/10.1109/RE.2017.14.

[42] G.M. Kanchev, P.K. Murukannaiah, A.K. Chopra, P. Sawyer, Canary: extracting requirements-related information from online discussions, in: Proceedings of the 25th International Conference on Requirements Engineering, RE, IEEE, 2017, pp. 31–40, http://dx.doi.org/10.1109/RE.2017.83.

[43] A.D. Sorbo, S. Panichella, C.A. Visaggio, M.D. Penta, G. Canfora, H.C. Gall, DECA: development emails content analyzer, in: Proceedings of the 38th International Conference on Software Engineering, ICSE, ACM, 2016, pp. 641–644, http://dx.doi.org/10.1145/2889160.2889170.

[44] I. Morales-Ramirez, D. Muñante, F.M. Kifetew, A. Perini, A. Susi, A. Siena, Exploiting user feedback in tool-supported multi-criteria requirements prioritization, in: Proceedings of the 25th International Conference on Requirements Engineering, RE, IEEE, 2017, pp. 424–429, http://dx.doi.org/10.1109/RE.2017.41.

[45] J. Cowie, W. Lehnert, Information extraction, Commun. ACM 39 (1) (1996) 80–91, http://dx.doi.org/10.1145/234173.234209.

Please cite this article as: I. Morales-Ramirez, F.M. Kifetew and A. Perini, Speech-acts based analysis for requirements discovery from online discussions, Information Systems (2019), https://doi.org/10.1016/j.is.2018.08.003.