IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 25, 2022, accepted March 29, 2022, date of publication April 7, 2022, date of current version April 25, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3165585

Classification of Speech Acts in Public Software Tenders

JORGE HOCHSTETTER[1], **CLAUDIO DÍAZ**[2], **MAURICIO DIÉGUEZ**[1], **JAIME DÍAZ**[1], **AND CARLOS CARES**[1]

[1]Departamento Ciencias de La Computación e Informática, Universidad de La Frontera, Temuco 4811230, Chile
[2]Departamento de Administración y Finanzas, Agrícola San José S.A., Piura 20103, Peru

Corresponding author: Jorge Hochstetter (jorge.hochstetter@ufrontera.cl)

This work was supported in part by DIUFRO Project through de la Universidad de La Frontera, Temuco, Chile, under Grant DI22-0043.

ABSTRACT Governments and private sectors currently procure software solutions for industry through public tender using mass distribution websites. This alternative organizes the demand and produces a large number of software tenders. **Objective**. The present study focuses on analyzing the texts of these documents to characterize them efficiently and explore a particular solution to the general problem known as ‘‘to bid or not to bid.’’ The tool is based on the automatic classification of speech acts, from where we generate different metrics from the Public Call Software Tender (PCST). **Methodology**. Our first approach was to use some analysis techniques suggested for Requirements Specifications. In particular, our interest focused on speech acts and the ontology-based on speech acts for analyzing requirements. These works focus on classifying software requirements in the early stages of the life cycle, which gave us a starting point for our work in PCST. We use our tool to analyze a set of four PCSTs downloaded from the Chilean Government’s public purchases website for the validation stage. The automatic analysis consisted in categorizing and classifying the four PCST downloaded, obtaining the measured values of the variables used by the metrics. **Results**. An initial assessment shows that the results of this application agree with the proposals generated manually by expert analysts. Our proposal saves time and effort when looking for relevant tenders. **Conclusion**. We consider the theory of speech acts, which allows texts to be categorized from a pragmatic point of view. We propose a first version of an automatic text classifier based on characterizing speech acts accompanied by metrics. This tool will allow potential tenderers in a public call for software tenders to decide whether it is worth tendering for the call. Based on these assumptions, we propose to use the identification of speech acts in requirements specifications to calculate a set of metrics that will enable us not only to describe PCST but also to compare them.

INDEX TERMS Public tender, software tenders, speech acts, requirements engineering, automatic classifier.

I. INTRODUCTION

For more than two decades, there has been an economic trend toward service outsourcing [1], [2]; information technology is one of the areas particularly affected by this process [3]. These changes have involved the software development market, divided into services into two levels: local and global [4]. Thus, when a public or private organization seeks to select a provider to develop a specific software product, one of the options considered – mandated if the buyer is a government organization – is to go through a public call for software tenders, referred to in this article as PCST.

This scenario has led to the outsourcing of many software development projects, in which emphasis is laid on establishing methodological conditions and restrictions on the budget, time, technology, and functional framework. These are difficult areas to negotiate because they are established in the tender document, leaving the potential supplier to decide whether or not to present a proposal. Different authors perceive a high level of risk associated with PCST for buyers and tenderers: For the first one, because they may not receive the product they need; and for the tenderers, because they do not have access to the information required to quantify the project properly [5].

Suppose the gap between stakeholders and developers is already enough in conventional projects [6]. In that case,

The associate editor coordinating the review of this manuscript and approving it for publication was Yang Liu.

it is to be assumed that it will not be better in projects whose requirements are not described by professionals close to software development, as happens in a PCST. Under these considerations, it is possible to point out that the software industry generates its software project offers under a scenario of uncertain and incomplete information. Due to the above, we add the fact that the websites where public calls are published can gather, in short periods, an essential set of tenders. We add complexity to the problem since it is necessary to decide which call to apply and which subset of calls to study.

From a scientific perspective then, the focus of the problem is on the methods that are useful for distinguishing those tender documents that contain more or fewer functional conditions of the product, more or fewer methodological conditions, or more or fewer restrictions on the project.

Speech acts [7], studied by pragmalinguistics, are a way of representing the intentionality of the content communicated, i.e., what a speaker wants to represent in the mind of another individual [8], [9]. This is what we hope to learn from PCST documents.

In pragmalinguistics, speech acts, which express the contents of documents, have rigid structures and can be decoded and represented in a formal, conceptual framework (ontology). Others can be inferred by reasoning and are fundamental in communication dynamics [10], distinguishing between acts of commitment, declaration, or questioning. Each of these may have widely differing interpretations in the context of a PCST.

On the other hand, various automatic analysis techniques allow complex tasks, such as searching for grammatical patterns and recognizing conceptual frameworks, concepts, and relations [11]. As well as other, simpler processes like analyzing word frequencies or temporal text drafting modes can help analyze a mass set of documents.

In particular, the automated classification of documents is presented as a process dependent on the context in which the words are used, complicating the task of deciding which category is appropriate for a document by studying the words that comprise it. We must consider different classification algorithms with dissimilar performances for the same problem. It is necessary to test and compare the performance of these algorithms when building the automatic classifier in such a way as to use the one that better solves the classification of the sentences.

So far as we have discovered, there is no generic solution to the problem of automatic classification based on the theory of speech acts [12], [13]. We, therefore, propose to address the problem in the specific domain of PCST. Our first approach was to use some analysis techniques suggested for Requirements Specifications. In particular, our interest focused on the speech acts proposed by [7] and on the ontology-based speech acts used by [14], [15] for analyzing requirements: these works focus on the classification of software requirements in the early stages of the life cycle, and this gave us a starting point for our work in PCST.

The choice of this conceptual framework was based on two points: first, the theory of speech acts makes it possible to classify the entire content communicated by the stakeholders without the need to adjust the document to a standard structure such as IEEE 830; second, in [15] This theoretical framework is used to structure an ontology to allow complex problems of software requirements classification to be tackled, providing a more comprehensive range of possible results.

In practical terms, the paragraphs of these documents, classified by type of speech act, are our classification objects. Our proposal complements other proposals for word frequency-based analysis of requirements and PCST [16], [17], increasing the types and number of metrics used to improve the characterization of tender documents.

We take PCST in Chile as our case study. Purchases by Chilean State organizations are governed by the Public Purchases Law, 19.886, passed in May 2003 and in force from October 2004. This law obliges the public sector to call for public tenders through the web platform known as Chilecompra (www.mercadopublico.cl) [17]. Thus, software product tenders are published and received on this platform and are still available after the tender closes.

In this way, we structure our work as follows: in section 2, we point out, from the perspective of automatic text analysis, how the area of Linguistics is related to Computer Science through Speech Acts. In section 3, we report the experience in creating the classifier based on speech acts, and then in section 4, we describe the steps of the methodology. Section 5 points out the metrics whose calculation is made possible thanks to the classifier. In section 6, we integrate the classifier with the proposed metrics exemplifying some real cases of tenders and comparing the results with the perception of expert professionals. In section 7, we describe the discussion and limitations. Finally, we present our main conclusions and future work.

II. AUTOMATIC TEXT ANALYSIS

The efforts of Information Science are channeled toward the study of phenomena related to information processing by computers [18]. In the 1980s, information science started to become integrated with linguistics, and the first attempts were made to process grammatical formalisms by computer [19]. In this context, the terms Computational Linguistics and Natural Language Processing appeared; these refer to the same discipline, the ultimate object of which is to analyze and fully understand human language [20].

Computational Linguistics comprises the treatment of both spoken and written language. Processing the spoken language forms part of what is known as speech technologies. These technologies are a set of tools for such varied tasks as the conversion of written texts into an oral equivalent, a transformation of speech into text, automatic transcription of conversations, voice verification in telephone services, and developing systems to allow oral dialogue between people and machines [21], [22].

Computational Linguistics is applied in written language to identify and analyze patterns associated with diverse linguistic variables. Therefore, its field of action is associated principally with concordance programs, statistical methods, searching through lists, and computer programs to facilitate the identification of occurrences with varying linguistic features in a given text [23].

The theories of traditional linguistics and natural language processing are not fully integrated. Although many of their study objects are shared, specific differences exist between their conceptualizations [21], [22]. Nevertheless, proposals exist like the description of speech acts in the communication of software agents in the Albatross language or implementations that seek to corroborate the consistency of some linguistic theories [23], [24].

As will be explained later, it is vitally important to integrate the knowledge developed in computational linguistics with the theory of speech acts at the core of our work. Remarkably, one of the motivational edges of the work is based on the possibility of using computational linguistics as a means to facilitate the automatic recognition of written patterns, which should be related to the types of speech acts. This idea underlies the intention of directly applying a taxonomy typical of traditional linguistics, being considered a classification explicit enough to be identified by a computer through the support of some computational tools.

A. LINGUISTICS AND SOFTWARE ENGINEERIG

One area of information technology as a technological discipline – as opposed to its scientific nature – is Software Engineering (SE) [19]. SE is a discipline that contributes to producing software systems designed to satisfy real needs [25]. One area of this discipline consists of investigations and tasks related to the study and analysis of stakeholders' needs; the name Requirements Engineering (RE) is a sub-discipline of particular importance for the present work [26].

RE refers to goal-related concerns in the real world and how they express functions and restrictions of software systems; it also includes the evolution of these objects, functions, and restrictions over time [27].

It is recognized that RE involves socio-economic, physical, technical, operational, and evolutionary facets [28], [29]. For example, it is the task of the RE to worry about the cultural feasibility of implementing a new software system, to verify that the infrastructure aspects are adequate. This is how basic aspects such as adequate electrical or furniture provision have been included, or the existence and quantification of technological elements related to a specific solution, such as scanners or digital cameras. From an operational point of view, the RE seeks to clarify the administrative environment, its procedures, and how the new software system will become part of said operations. Consequently, given its broad scope, we consider that we are justified in adopting an interdisciplinary approach in this work. In this field, we can observe proposals based on interpretations of units and theories common to traditional linguistics, as many approaches suggest [30]–[35].

Taking advantage of this new relation between SE and linguistics, we have extended the contribution of the theory of speech acts to RE. From an ontological core for the definition of requirements as proposed in [15] to a set of objective measures – jointly known as SE metrics – that can be applied to a group of requirements specifications associated with PCST. Therefore, we use a linguistic approach to progress towards quantitative management of the process of eliciting requirements in software production.

B. SPEECH ACTS IN PCST

One of our motives for analyzing software tenders is finding a methodology to facilitate evaluation and understanding of the technical bases. As these documents are often considered software requirements specifications, we believe that the ontology of [15] offers a reasonable basis for classifying the content under this paradigm. The reasons for this appreciation are first that its conceptualization enables each paragraph present in the technical specifications of the proposed system to be assessed. Secondly, a large proportion of its concepts are closely related to natural language characteristics; therefore, as we explain below, the contents can be classified as speech acts.

Under this new ontology, it is not necessary to expect the structures of the documents to obey ideas external to the context of the stakeholders. In other words, it is much more likely that each requirement contained in the document can be classified in this theoretical framework.

Most of the definitions associated with ontology [15] are based just on the classification of speech acts [7] and on the concept of quality [36]. As a result, most of the distinctions are not associated with complex relations between the tender document paragraphs. Therefore, it is feasible to recognize most of them from a fragmented document.

Speech acts [7] can be distinguished into various types: directive, assertive, of commitment, expressive, declarative, and representative-declarative, each of which may contain sub-types. These are described and explained in [15] in the context of software requirements specification documents.

Describing a new taxonomy of speech acts was no easy task; our first version did not produce a satisfactory result in the Kappa Agreement statistical test [37], so we had to improve the objectivity of the descriptions of each speech act. The descriptions are shown below present acceptable agreement in the classifications of actual speech acts obtained from the public platform of the Chilean state:

a) Assertive (AS): an expression that transmits what the speaker believes to be accurate or true. The content is not necessarily genuine; the assertive speech act only transmits that the speaker holds it to be confirmed. For example: "In 2018, a remote training phase was carried out for 527 professionals; at the time of their participation in face-to-face training, 90% of them complied with the requirements''.

b) Declarative (DE): is an expression that proposes reality by its declaration. In other words, depending on his/her role, the person making this declaration must successfully alter reality; thus, e.g., by declaring war or declaring him/herself a candidate for office, the speaker converts this declaration into a part of reality. For example: ''The Municipality reserves the right to request new system modules from the winning tenderer during the contract's validity.''

c) Representative-declarative (RD): is an expression that recognizes the truth of a proposition made real by a declaration. The speaker recognizes the conditions described by his/her influence on the content. For example: ''The Municipality reserves certain rights concerning the information made available during software development.''

d) Of commitment (CO): is an expression indicating to the hearer the speaker's intention of acting; if the content describes conditions, the commitment speech act indicates that the speaker will carry out the necessary actions to create situations in which these conditions will be met. For example: ''Once the tender is awarded, the documentation of the software that will interact with the system will be handed over.''

e) Expressive (EX): transmits a state of mind sincerely that the speaker holds concerning a condition. An expressive speech act expresses something equivalent to assessing the condition favoring or rejecting something more or less firmly. For example: ''The website should be programmed in PHP rather than ASP.''

f) Directive (DI): This expression describes the conditions that the speaker wishes to see converted into reality. In contrast with the assertive speech act, the speaker considers that the conditions are not met but that the hearer could comply with his/her wishes in an undefined future moment. Moreover, these speech acts are acts of imposition, intentionally influencing the hearer's behavior [38].

g) Furthermore, in order to offer a minimum conceptual framework, we have added the following complementary definition in addition to those given above:

h) Quality: is an essential entity capable of perception and measurement inherent in other entities (e.g., red may be inherent in various entities or objects).

i) The selection of document paragraphs is an important aspect. For this purpose, we define a tender document paragraph as the text between two punctuation marks which may be the start of a paragraph or a complete stop.

j) For example, under this criterion, the following section contains two tender document paragraphs: ''The system must allow reports to be visualized in the screen for the report results to be reviewed before printing. It must be possible for reports to be issued for periods chosen by the user when necessary''.

k) This selection is not linguistic and must follow the criterion proposed in [15]. Conjunctive, disjunctive, and sequential connectors should also be included. However, these do not follow simple criteria, and we do not have lists or structures which would facilitate objective recognition. Nevertheless, if the level of agreement between assessors is high, a satisfactory classification can be obtained by this method, despite the problem of how to divide the tender document paragraphs. In other words, under this evaluation, different assessors must classify the speech acts in the text of a public call for tenders in the same way. The fact that components of various speech acts coexist in a single tender document paragraph does not contradict the theory of speech acts. To be more precise, although, in a different field, the classification of types of speech acts has been described as one of the non-exclusive distinctions, so its borders are not always well defined [38]. In our case also, this appears to be accurate; for example, if a tender document paragraph says ''details of the modules are not specified, since the tenderers must have such modules functioning in other municipalities,'' we can classify the content as a directive speech act.

l) Finally, we must recognize that, despite everything stated in this section, in the trials carried out in our investigation, we had to accept that not all the contents of the technical bases of PCST refer to technical bases. Due to human error, they are often mixed with the administrative bases; we have classified these declarations as NA.

III. AUTOMATIC CLASSIFIER

The early stage of the investigation was based on a proposed set of conceptual definitions of speech acts for software requirements engineering. In addition to these definitions, we proposed the set of metrics described in Section 3; we show how they were calculated and conjecture on the impact they could have on the descriptions contained in and the decisions taken on PCST.

With the proposed metrics, a PCST can be scored and a quality measurement delivered to support the tenderers in deciding whether to apply. To feed the proposed forms, the tender document paragraphs contained in a PCST must be identified and classified under the criteria of the speech acts described in this article. If carried out manually, however, this process may be tiresome and subjective, potentially introducing errors into the result. To reduce the degree of error, we propose the possibility of automatically carrying out the process, from selecting the PCST on the public market website to estimating the degree of completeness of the tender document, as shown in Figure 1.

As PCST analysis is the process that feeds the proposed metrics and is also one of the tasks which contain the most significant probability of increasing errors in estimation, we observed a need to develop an efficient method for classifying these documents. We, therefore, propose that this process should be automated.

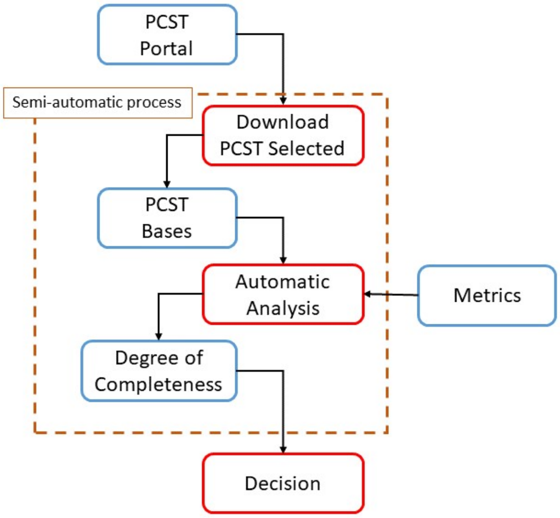


FIGURE 1. **Proposed automatic process.**

We propose to create the first version of an automatic text classifier that can identify and classify the various tender document paragraphs contained in a PCST under the criteria of the speech acts described above, determining the number of each type. These data can be used to feed the metrics and thus obtain a "quality score" for the PCST. As [12] says, automatic text classification can be applied by identifying different speech acts, although the quality of the results will vary according to the classification conditions and the algorithm used.

According to [39], automatic document classification is the automatic assignment of a set of documents to various pre-existing categories. In this case, however, different sections of a document will be classified based on the speech acts described above.

IV. METHODOLOGY

We follow Sánchez's [39] recommendations for defining an automatic text classifier. The steps that we followed in constructing the classifier were adapted from [39] and consisted of: (i) document construction to be classified, (ii) Preparation of the entry data, (iii) Classification, and (iv) results from comparison.

A. DOCUMENTS CONSTRUCTION TO BE CLASSIFIED

The first step was to create the test document, which contained a subset of PCST document paragraphs, labeled and categorized, extracted from the Chilecompra public tender's page. Labeling and categorization were carried out manually in a review by two experts. The document is in ARFF format to allow processing with the "Weka" software application for data analysis and mining [40], [41] version 3.6.11.

We categorized 1,550 tender documents paragraphs to design and construct the classifier, taken from a set of 15 PCSTs from Municipalities. 4 attributes described the requirements. These attributes are described in Table 1.

TABLE 1. **Attributes of tender document paragraphs.**

Attribute	Description
id_doc	Identifier of each tender
id_parr	Identifier of each paragraph in a tender
text	The body of the tender document paragraph
cat_ach	The category of each tender document paragraph

B. PREPARATION OF THE ENTRY DATA

The next step was to pre-process the data, eliminating words that did not provide relevant information for the classification. This step reduced the number of words to be reviewed, diminished the classification time, and increased the success rate of the classifier.

The attributes id_doc and id_parrafo are eliminated from the classification process. The first contains a number that identifies the PCST document paragraph from the rest; it is not associated with the content of the PCST document paragraph but rather with the order in which the document is searched. Likewise, the attribute id_parrafo is not associated with the subject of the tender document paragraph but with the number of tender document paragraphs in a PCST.

The tender document paragraphs contained in the set were then pre-processed by applying a filter to the words of each paragraph. In this particular case, an unsupervised filter was applied to the attributes, called StringToWordVector, the main object of which is to turn the words into character vectors to order the articles' content to allow them to be processed subsequently by the classifier.

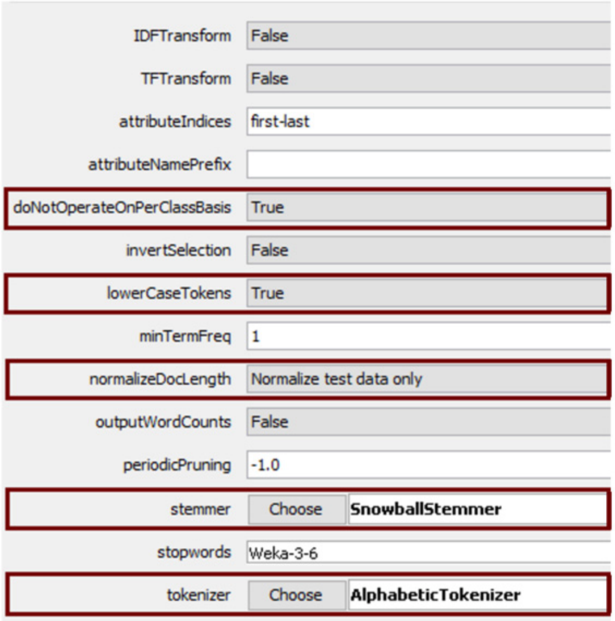
Depending on the configuration of the filter, words or characters that contributed nothing to the classification could be eliminated (e.g., common words like articles or pronouns, spaces, punctuation marks, among others); these elements are known as StopWords.

The configuration used for the exercise resolution is shown in Figure 2.

With the configuration defined for the filter, 1,077 attributes were obtained from the 1,546 documents; these attributes were used to start constructing the classifier.

C. CLASSIFICATION

We define the process of categorizing each tender document paragraph by the type of speech act defined. According to [42], creating an automatic text classification system consists of discovering variables that will be useful for distinguishing the texts that belong to pre-existing classes. According to this definition, these classifiers must be trained on a set of previously classified documents. This definition implies that the work methodology must include two groups of documents to work on: the training set and the test set, which must not contain the same tender document paragraphs.



IDFTransform	False
TFTransform	False
attributeIndices	first-last
attributeNamePrefix	
doNotOperateOnPerClassBasis	True
invertSelection	False
lowerCaseTokens	True
minTermFreq	1
normalizeDocLength	Normalize test data only
outputWordCounts	False
periodicPruning	-1.0
stemmer	Choose SnowballStemmer
stopwords	Weka-3-6
tokenizer	Choose AlphabeticTokenizer

FIGURE 2. Filter settings StringToWordVector.

These sets were obtained by applying two options in the Weka software:

Cross-Validation: this option consists of separating different sets – called ''Folds'' – using a determined number of attributes. The number of sets can be varied, changing the number of Folds used. The default value used by Weka is 10 Folds.

Percentage-Split: this method consists in separating the cases into two groups. The number of cases per group is determined as a percentage of the total cases. The default value used by Weka is 66%.

In constructing the classifier, we worked with both methods, testing both the Cross-Validation and the Percentage Split with different values to determine their effectiveness.

On this basis, we then evaluated different classification algorithms. Based on the characteristics of the document and of the classification techniques, as indicated by [39], [43]–[45], we decided to use the Naive Bayes Multinomial, Random Forest, JRIP, and Support Vector Machines methods. We worked with the same test options when separating the training and test sets, conducting various experiments to find the best configuration of the different classification algorithms.

D. RESULTS COMPARISON

We compared the results obtained from the classification of the tender document paragraphs by the different classification methods. This comparison was based on two main aspects: the number of document paragraphs correctly classified and the confusion matrix that resulted from each experiment.

TABLE 2. Results of classification algorithms.

	Cross Validation		Percentage Split	
Algorithms	Folds	% Correct	%	% Correct
Naive Bayes Multinomial	30	82.1475	71	82.5893
Random Forest	20	80.8538	75	79.403
Support Vector Machine	30	80.207	36	77.452
JRIP	30	81.3066	67	79.798

The results obtained with both Cross-Validation and the Percentage Split with the various algorithms are described in Table 2.

Table 2 deduces that the percentages of correctly classified tender document paragraphs are very similar between the different classification algorithms and the two methods of separating test and training sets. However, with a Percentage Split of 71%, the Naive Bayes Multinomial algorithm was the most successful in the classification, achieving 82% of correctly classified tender document paragraphs. After analyzing the confusion matrix, we concluded that the majority of tender document paragraphs were concentrated in just one of the five categories (directive speech act), which accounts for more than 70% of the results. This result implies that this category of speech act strongly impacts the documents analyzed.

With the taxonomy of speech acts described in Section 1, we established a basis for characterizing the contents communicated by the stakeholders in the PCST. The document paragraphs in PCST can be classified in this way by dividing the text into fragments following our criteria of sentences between two full stops.

It should be noted that in our trials, various fragments resisted classification. We attribute this problem to two factors. It is impossible to classify document items expressed in graphic formats – such as tables or diagrams – with a standard text-processing algorithm. The second is that tender documents may contain errors. In some documents, paragraphs defining technical bases (which may be considered requirements specifications) are mixed with others referring to administrative bases, which have no direct influence on the product to be constructed. For this reason, we added a general metric to cover this possibility. Nevertheless, we considered that the level of agreement achieved in our classifications was such that the application would be sufficiently consistent with producing metrics values from which documents of different natures, and even different structures, could be compared.

V. METRICS FOR PCST IN SPEECH ACTS

Based on the characteristics and information they provide, we classified the metrics into two groups. The object of the first group is to describe the profile of each PCST; these metrics indicate proportions between the number of document paragraphs corresponding to the different speech acts identified. The second group generally refers to the quality of concentration of the different types of document paragraphs in a PCST; this enables us to determine how well balanced the document is in relation to the speech acts identified. For convenience, we defined all the metrics with values between zero and one. The proposed metrics are:

A. DESCRIBABLE PORTION

Describes the fraction of a document represented by the total number of speech acts that can be classified. We propose to calculate this as the sum of speech acts (#AS + #DE + #RD + #DI + #CO + #EX) divided by the total number of separate document paragraphs. A low proportion may indicate that the method cannot describe the PCST in question properly.

B. DIRECTIVE PORTION

Represents directive speech acts as a proportion of the total number of document paragraphs in the PCST. If the value of the Directive Portion is close to zero, this means that the system's expectations tendered for are not well explained. If it is close to one, the specification probably does not give sufficient weight to other concepts, e.g., quality descriptions. For our classification, the equation defining the Directive Portion is:

$$DirectivePortion = \frac{\#DI}{\#\text{Total speech acts}}$$

C. PREFERENCES PORTION

The final metric that we have defined in this first group refers to expressions of preference in the evaluation contained in expressive speech acts. Although [15] say that these speech acts must be related to other concepts in the document to be considered elements of the specification, it is highly probable that, given the formality of the PCST, any expressive act contained will refer to concepts that are important for the specification. Thus we will assume that an expressive speech act in a PCST refers to an evaluation or preference between options given in the PCST. This portion is calculated as follows:

$$PreferencesPortion = \frac{\#EX}{\#EX + \#DI}$$

EX represents expressive speech acts. The higher the value of this metric, the greater the number of document paragraphs that propose ways of assessing an offer through the preferences affecting each buyer's expectations. Likewise, if a large proportion of expressive speech acts for the requirements to be satisfied – indicated by the Preferences metric – the tenderers can compare different systems that satisfy the stakeholders' needs and evaluate which best meets the buyer's preferences.

The second group of metrics is associated with the orderliness of the documents containing the technical bases. We believe that an orderly document is preferable to a disorderly one; we thus consider the premise that an orderly document groups the paragraphs associated with one or more of the speech acts described above. Once the document paragraphs have been classified and numbered in PCST, the dispersion of each type of speech act can be assessed. The dispersion of a group of these concepts can also be assessed if we want to consider all the concepts associated with the domain assumptions. Moreover, there are several ways of measuring statistical dispersion; our conception is closest to definitions of mean absolute deviation, variance, or standard deviation.

Subsequently, we face the need to define a metric that will allow us to compare the order of documents with different numbers of paragraphs. In this situation, we observe that, due to the finite size of the technical bases and that two paragraphs cannot occupy the same position, the dispersion is limited by a minimum dispersion (Dmin) and a maximum dispersion (Dmax). i.e.:

$$D_{min} \leq D_{real} \leq D_{max}$$

From which we can derive:

$$0 \leq \frac{D_{real} - D_{min}}{D_{max} - D_{min}} \leq 1$$

D. CONCENTRATION OF DOCUMENT PARAGRAPHS

From the latter expression we can obtain a metric that indicates the orderliness of a document, which is also a useful measure of comparison between different documents. We call this metric Concentration of Homogeneous Document Paragraphs (CDP) and define it as follows:

$$CDP = \frac{D_{real} - D_{min}}{D_{max} - D_{min}}$$

Thus, a CDP close to one indicates that the document is disorderly concerning the evaluated concept or group of concepts. It should be noted here that while an orderly document is preferred, this is no guarantee of its quality. This metric is an indicator of just one aspect of the many that influence the perception of the quality of a document.

Once we understand the mechanics of this metric, we can measure the orderliness of the directive speech acts in a document. To do this, we must search all the positions of the document paragraphs containing the three types of directive speech acts. If they are orderly (i.e., their CDP is close to zero), the document probably contains a section or various sections close together that refers to what the stakeholders hope to obtain, regardless of the content of their titles. Likewise, the CDP of the document paragraphs related to these domain assumptions can also be assessed. In this way, we can identify the positions of all the document paragraphs referring to assertive, declarative, and declarative-representative speech acts. Then – just as occurred with document paragraphs representing directive speech acts – if these are orderly, the document probably has one or several sections referring to the medium or domain in which the system required by the stakeholders will function, independent of the titles or internal divisions of the document.

VI. EXAMPLE APPLICATION OF THE PROPOSAL

In order to test the application of the proposed classifier and its integration with the metrics described, we carried out an

TABLE 3. Experts' judgements of PCST selected.

Algorithms	PCST1	PCST2	PCST3	PCST4
Clarity of the functions	High	Medium	Medium	Low
Directive Portion	0.89	0.80	0.83	0.55
Preferences Portion	0.01	0.01	0.00	0.00

experiment in which we sought to determine the feasibility of applying this automatic PCST selection method. We used the method to analyze a set of four PCSTs downloaded from the Chilean Government's public purchases website. These PCSTs were assessed by two analysts who work with Requirements Engineering; they reported their general perceptions of how the required functions of each PCST were presented. Three classification levels were defined: low, medium, and high.

The automatic analysis consisted in categorizing and classifying the four PCST downloaded, obtaining the measured values of the variables used by the metrics. The Directive Portion and Preferences Portion metrics were then calculated for each PCST. The results of the two analyses are shown in Table 3.

Given the exploratory nature of this proposal, which focuses on laying the foundations for quantitative analysis, the result from Table 3 must be understood just as support. The idea is that speech acts are a feasible approach to assessing PCST and that a classifier is a valuable tool for automating this assessment.

From the results presented in the table, it can be seen that the metrics indicate that the first two PCSTs have the highest values, which agrees with the assessment of the expert analysts. From this, and without generalizing a tendency, we can say that the results provided by the automatic process agree with the analysts' opinions. They propose that the first two PCSTs are the best options to choose to apply.

VII. DISCUSSION AND LIMITATIONS

Governments and private sectors currently procure software solutions for industry through public calls for tender using mass distribution websites. This organizes the demand and produces a large number of software tenders. The present study focuses on analyzing the texts of these documents to characterize them efficiently and thus explore a particular solution to the general problem known as ''to bid or not to bid.'' We consider the theory of speech acts, which allows texts to be categorized from a pragmatic point of view. We propose our first version of an automatic text classifier based on the characterization of speech acts accompanied by metrics. This tool will allow potential tenderers in a public call for software tenders to decide whether it is worth tendering for the call.

This experiment is an initial proposal, and towards formalization, we need to apply it to a more significant number of PCSTs to establish statistically consistent results. We also propose to carry out quantitative investigations to establish the relation between the quality items of a PCST (consistence, risk, attractiveness, completeness) and the values of metrics based on speech acts. The level of correlation found will give us a better understanding of the nature of these documents and thus enable purchasers to draft documents that will be easier to select and assess.

VIII. CONCLUSION AND FUTURE WORK

In this article, we have indicated the importance of PCST documents and the problem facing providers when they need to concentrate their efforts on the calls in which they have the most excellent chances of success. The basic idea of this evaluation is not only to create a quality indicator for PCST but to make progress in identifying relations between certain ''types'' of PCST and recognizing that specific software provider profiles are better suited to certain types of PCST. Based on these assumptions, we propose to use the identification of speech acts in requirements specifications to calculate a set of metrics that will enable us not only to describe PCST but also to compare them.

Furthermore, we present a set of descriptions that we used to classify the speech acts identified automatically. We generated the conjecture that the proposed metrics could be obtained automatically from PCST written in natural language. We use an example to show that it is feasible to apply the proposed model to real cases; an initial assessment shows that the results of this application agree with the proposals generated manually by expert analysts.

The lines of future work derived from this proposal will first address automatic recognition of the speech acts identified. We think it is also feasible to increase their exactness in an additional refinement following the distinctions [44], which distinguish between project requirements, system requirements, and process requirements – all elements present in PCST.

Linked with the above, we consider that the empirical aspect is essential. In this respect, our next step will be to assess a more representative set of PCST using the proposed metrics and to correlate these values with the providers' profiles. This will generate the tools to make a prognosis of certain types of PCST that will probably be awarded to certain providers in the software industry.

REFERENCES

[1] R. McIvor, *Global Services Outsourcing*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[2] I. Oshri, J. Kotlarsky, and L. Willcocks, *The Handbook of Global Outsourcing and Offshoring: The Definitive Guide to Strategy and Operations*. Basingstoke, U.K.: Palgrave Macmillan, 2015.

[3] M. R. G. Ramírez, J. L. G. Gascó, and J. L. Taverner, ''Outsourcing de sistemas de información: Situación actual, evolución y tendencias,'' *Investigaciones Europeas de Dirección y Economía de la Empresa*, vol. 21, no. 2, pp. 93–99, 2015.

[4] A. Castillo, J. Cavieres, and N. Selamé, ''El trabajo asalariado flexible: La subcontratación y la organización colectiva en Chile,'' *Panorámica del Trabajo en el Chile Neoliberal*, vol. 156, p. 156, 2015.

[5] K. Breiner, M. Gillmann, A. Kalenborn, and C. Müller, ''Requirements engineering in the bidding stage of software projects—A research preview,'' in *Requirements Engineering: Foundation for Software Quality*. Cham, Switzerland: Springer, 2015, pp. 270–276.

[6] M. Suhaib, "Conflicts identification among stakeholders in goal oriented requirements engineering process," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 12, pp. 2278–3075, Oct. 2019.

[7] J. R. Searle, *Speech Acts: An Essay in the Philosophy of Language*. Cambridge, U.K.: Cambridge Univ. Press, 1969.

[8] M. A. Palacio and L. Gustilo, "A pragmatic analysis of discourse particles in Filipino computer mediated communication," *Online Submission*, vol. 16, no. 3, pp. 1–19, 2016.

[9] G. Leech, *Principles of Pragmatics*. Evanston, IL, USA: Routledge, 2016.

[10] A. T. Valls, "Aportaciones de la Pragmática y del Análisis del Discurso a la enseñanza del Español como Lengua Extranjera," *Aportaciones la pragmática y del análisis del discurso a la enseñanza del español como Leng. Extranj.*, pp. 159–177, 2014.

[11] S. Sayago, "El análisis del discurso como técnica de investigación cualitativa y cuantitativa en las ciencias sociales," *Cinta de Moebio*, no. 49, pp. 1–10, 2014.

[12] E. Gustafson, "Classification of speech acts in discussion threads," Longman, Essex, U.K., Tech. Rep., 2011.

[13] W. Andewi and W. Waziana, "An analysis of teacher's speech acts in teaching and learning process," *Teknosastik*, vol. 17, no. 1, pp. 29–34, 2019.

[14] I. J. Jureta, J. Mylopoulos, and S. Faulkner, "A core ontology for requirements," *Appl. Ontol.*, vol. 4, nos. 3–4, pp. 169–244, 2009.

[15] I. J. Jureta, J. Mylopoulos, and S. Faulkner, "Revisiting the core ontology and problem in requirements engineering," in *Proc. 16th IEEE Int. Requirements Eng. Conf.*, Sep. 2008, pp. 71–80.

[16] R. McIvor, *Global Services Outsourcing*. Cambridge, U.K.: Cambridge Univ. Press, 2010, doi: 10.1017/CBO9780511844911.

[17] J. Hochstetter, M. García, and C. Cares, "Socio-technical factors in electronic software biddings," *J. Theor. Appl. Electron. commerce Res.*, vol. 14, no. 3, pp. 34–60, Sep. 2019.

[18] J. Hochstetter, C. Vairetti, C. Cares, M. G. Ojeda, and S. Maldonado, "A transparency maturity model for government software tenders," *IEEE Access*, vol. 9, pp. 45668–45682, 2021, doi: 10.1109/ACCESS.2021.3067217.

[19] R. Rubio and J. Bernal, "Introducción a la lingüística computacional," Instituto Caro y Cuervo y Ediciones de la U. Dirección, Bogotá, Colombia, Tech. Rep., 2016.

[20] A. A. T. Orellana, "Evaluación de técnicas de procesamiento de lenguaje natural para textos cortos en lenguaje español," Bachelor's thesis, Universidad del Azuay, Cuenca, Ecuador, 2018.

[21] A. I. C. García, "Lingüística computacional," Universidad de Granada, Granada, Spain, Tech. Rep., 2020.

[22] M. C. Campos, "Los corpus del español clásico y moderno: Entre la filología y la lingüística computacional," *RLA. Rev. lingüística teórica y Apl.*, vol. 57, no. 2, pp. 41–64, 2019.

[23] J. M. Mendel, "A comparison of three approaches for estimating (synthesizing) an interval type-2 fuzzy set model of a linguistic term for computing with words," *Granular Comput.*, vol. 1, no. 1, pp. 59–69, 2016.

[24] O. Sohaib, M. Naderpour, W. Hussain, and L. Martinez, "Cloud computing model selection for e-commerce enterprises using a new 2-tuple fuzzy linguistic decision-making method," *Comput. Ind. Eng.*, vol. 132, pp. 47–58, Jun. 2019.

[25] I. Sommerville, *Software Engineering*, 7th ed. London, U.K.: Pearson, 2010.

[26] B. Geurts, "Communication as commitment sharing: Speech acts, implicatures, common ground," *Theor. Linguistics*, vol. 45, nos. 1–2, pp. 1–30, Jun. 2019.

[27] E. M. Schön, J. Thomaschewski, and M. J. Escalona, "Agile requirements engineering: A systematic literature review," *Comput. Stand. Interface*, vol. 49, pp. 79–91, Jan. 2017.

[28] P. A. Laplante, *Requirements Engineering for Software and Systems*, 3rd ed. New York, NY, USA: Auerbach, 2017, doi: 10.1201/9781315303710.

[29] A. Bennaceur, T. T. Tun, Y. Yu, and B. Nuseibeh, "Requirements engineering," in *Handbook of Software Engineering*. Cham, Switzerland: Springer, 2019, pp. 51–92.

[30] C. Rolland and C. Proix, "A natural language approach for requirements engineering," in *Seminal Contributions to Information Systems Engineering*. Berlin, Germany: Springer, 1992, pp. 257–277.

[31] C. M. Zapata, A. F. Jaramillo, and F. I. Arango, "Una propuesta para mejorar la completitud de requisitos utilizando un enfoque lingüístico," *Rev. Científica Ing. y Desarro.*, vol. 19, no. 19, pp. 1–16, 2011.

[32] C. Cares, "A semiophere-based approach for modelling pre-semantic engineering communication," Universidad de La Frontera, Temuco, Chile, Tech. Rep., 2008.

[33] F. Mokammel, E. Coatanéa, J. Coatanéa, V. Nenchev, E. Blanco, and M. Pietola, "Automatic requirements extraction, analysis, and graph representation using an approach derived from computational linguistics," *Syst. Eng.*, vol. 21, no. 6, pp. 555–575, Nov. 2018.

[34] L. Zhao, W. Alhoshan, A. Ferrari, K. J. Letsholo, M. A. Ajagbe, E.-V. Chioasca, and R. T. Batista-Navarro, "Natural language processing (NLP) for requirements engineering: A systematic mapping study," 2020, *arXiv:2004.01099*.

[35] A. Ferrari and A. Esuli, "An NLP approach for cross-domain ambiguity detection in requirements engineering," *Autom. Softw. Eng.*, vol. 26, no. 3, pp. 559–598, 2019.

[36] M. Hu and Y. Liu, "E-maintenance platform design for public infrastructure maintenance based on IFC ontology and semantic web services," *Concurrency Comput., Pract. Exper.*, vol. 32, no. 6, p. e5204, 2020.

[37] A. J. Viera and J. M. Garrett, "Understanding interobserver agreement: The Kappa statistic," *Family Med.*, vol. 37, no. 5, pp. 360–363, 2005.

[38] C. Otaola, "El verbo y las expresiones directivas en el discurso político," in *Estudios Ofrecidos al Profesor José Jesús de Bustos Tovar*, vol. 2. Madrid, Spain: Editorial Complutense, 2003, pp. 1033–1053.

[39] R. S. Jiménez, "La documentación en el proceso de evaluación de Sistemas de Clasificación Automática," Universidad Complutense de Madrid, Madrid, Spain, Tech. Rep., 2007, pp. 25–44, vol. 30.

[40] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *ACM SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, 2009.

[41] P. Kotak and H. Modi, "Enhancing the data mining tool WEKA," in *Proc. 5th Int. Conf. Comput., Commun. Secur. (ICCCS)*, Oct. 2020, pp. 1–6.

[42] R. Venegas, "Clasificación de textos académicos en función de su contenido léxico-semántico," *Revista Signos*, vol. 40, no. 63, pp. 239–271, 2007.

[43] M. F. B. Othman and T. M. S. Yau, "Comparison of different classification techniques using WEKA for breast cancer," in *Proc. 3rd Kuala Lumpur Int. Conf. Biomed. Eng.*, 2007, pp. 520–523.

[44] P. M. Barnaghi, V. A. Sahzabi, and A. A. Bakar, "A comparative study for various methods of classification," in *Proc. Int. Conf. Inf. Comput. Netw. (ICICN)*, vol. 27, 2012, pp. 875–881.

[45] M. N. Amin and M. A. Habib, "Comparison of different classification techniques using WEKA for hematological data," *Amer. J. Eng. Res.*, vol. 4, no. 3, pp. 55–61, 2015.

JORGE HOCHSTETTER received the master's degree in educative informatics in information systems engineering from La Frontera University, Temuco, Chile, in 2010, and the Ph.D. degree from Alicante University, Spain, in 2021. He is currently an Academician with La Frontera University. His research interests include requirements engineering, software development tenders, and transparency in electronic procedures.

CLAUDIO DÍAZ received the degree in industrial engineering from the University of La Frontera. Between 2012 and 2014, he worked analyzing biddings of software with the University of La Frontera and Federico Santa María Technical University. He is currently working in private companies.

MAURICIO DIÉGUEZ received the Doctorate degree in informatics from the University of Alicante. He is currently working as a full-time Teacher and the Director of the Department of Computer Science and Informatics, Universidad de La Frontera. His research interests include information security management, electronic government, and engineering education.

JAIME DÍAZ received the Ph.D. degree in engineering informatics and the master's degree in computer science. He is currently working as a full-time Teacher with the Universidad de la Frontera, Chile, in business processes and user experience advisor. He is a Technological Projects Evaluator with CORFO (a Chilean Government Organization). His research interests include HCI, e-commerce, and education.

CARLOS CARES received the degree in engineering from the Universidad de Concepción, Chile, in 1989, the Master of Engineering degree from Universidad Federico Santa María, Chile, in 1996, and the Ph.D. degree from the Technical University of Catalonia, Spain, in 2012. He studied civil-informatics engineering at the Universidad de Concepción, obtaining his professional qualification, in 1989. He is currently the Head of the Software Engineering Studies Center, University of La Frontera, Temuco, Chile. His experience is in information security, e-health, and e-government applications. His research interests include requirements engineering and software engineering for intelligent systems.