Intelligent Data Analysis 23 (2019) 191–214
DOI 10.3233/IDA-173807
IOS Press

Sentiment analysis and opinion mining applied to scientific paper reviews

Brian Keith Norambuena*, Exequiel Fuentes Lettura and Claudio Meneses Villegas
Department of Computing and Systems Engineering, Universidad Católica del Norte, Coquimbo, Chile

Abstract. Sentiment analysis and opinion mining is an area that has experienced considerable growth over the last decade. This area of research attempts to determine the feelings, opinions, emotions, among other things, of people on something or someone. To do this, natural language techniques and machine learning algorithms are used.

This article discusses the problem of extracting sentiment and opinions from a collection of reviews on scientific articles conducted under an international conference on computing in northern Chile.

The first aim of this analysis is to automatically determine the orientation of a review and contrast this with the assessment made by the reviewer of the article. This would allow scientists to characterize and compare reviews crosswise and more objectively support the overall assessment of a scientific article.

A hybrid approach that combines an unsupervised machine learning algorithm with techniques from natural language processing is proposed to analyze reviews. This method uses part-of-speech (POS) tagging to obtain the syntactic structure of a sentence. This syntactic structure, along with the use of dictionaries, allows determining the semantic orientation of the review through a scoring algorithm.

A set of experiments were conducted to evaluate the capability and performance of the proposed approaches relative to a baseline, using standard metrics, such as accuracy, precision, recall, and the F_1-score. The results show improvements in the case of binary, ternary and a 5-point scale classification in relation to classical machine learning algorithms such as SVM and NB, but they also present a challenge to improve the multiclass classification in this domain.

Keywords: Opinion mining, sentiment analysis, paper reviews, hybrid methods

1. Introduction

Opinions are central to almost all human activities because they are a key influence on people's behavior. Each time a decision needs to be made, humans look for others' opinions. In the real world, enterprises and organizations seek to know public opinion about their products and services. In turn, customers want to know others' opinion about a certain product before buying it. In the past, people looked for opinions from their friends and family, while organizations made polls or organized focus groups. Nevertheless, with the sudden growth of social networks such as Twitter and Facebook, individuals and organizations use data provided by these means to support their decision-making process. The field of sentiment analysis, also called opinion mining, emerged in this context.

Sentiment analysis is a relatively recent area in the field of data mining. There are different techniques for extracting, processing, and seeking objective data in texts. Nevertheless, there are subjective components that are also interesting. These components including opinions, sentiments, and emotions, among others, are the focus of sentiment analysis.

*Corresponding author: Brian Keith Norambuena, Department of Computing and Systems Engineering, Universidad Católica del Norte, Coquimbo, Chile. E-mail: brian.keith@ucn.cl.

Sentiment analysis includes a great amount of tasks such as sentiment extraction and classification, subjectivity detection, opinion summary, and opinion spam detection, among others. To do these tasks accurately, it is necessary to face several challenges, particularly the meaning formalization of an opinion. For this purpose, a series of formalisms and math representations to express opinions have been developed.

Sentiment analysis is an area with great development opportunities, particularly due to the huge growth of data available in the web, for example, in blogs, social networks, and forums, among others. One of the applications of opinion mining is product or service assessment by analyzing users' opinions or reviews. This application is highly important for organizations because it allows discovering what people think and say about a certain trademark [20].

An application area where opinion mining techniques have not been applied yet is the reviewing process of scientific articles. In addition, the scientific paper reviewing process is the main quality control mechanism for most scientific communities. This involves reviewing each paper in order to provide suggestions to authors for correcting and improving a paper, whether they think it can be published or must be rejected [5]. As in the sentiment analysis in the industry, there is a suggestion to use opinion mining for analyzing the orientation of scientific paper reviews. This paper shows the application of sentiment analysis on a data set consisting of paper peer reviews.

The domain of scientific paper reviews presents some major challenges, such as:

1. Usually classes are unbalanced, because there is a strong bias towards negative opinions.
2. Different reviews usually vary in terms of the number of assessments.
3. Normally, there is not a clear correlation between the number of positive and negative opinions with the final evaluation made by reviewers.

All these issues make this domain a challenge for opinion mining and sentiment analysis purposes.

Specifically, anonymous reviews taken from an international conference have been used as a data set. This conference is an academic/business event of informatics and computer engineering. Authors submitted their papers through EasyChair. The papers could be written in Spanish, English or Portuguese. A double blind review scheme was used to prevent biases during the evaluation of the different articles. An international reviewing committee was in charge of the evaluation of each paper. The papers were distributed among the reviewers according to their affinity to the corresponding research area. The reviewers evaluated the submitted papers and provided their comments and evaluations in Spanish and in some cases in English.

This paper aims to present the implementation of sentiment analysis methods in the area of scientific paper reviews as a proof of concept for future applications. The used techniques include a Bayesian classifier (NB), a classifier built on the basis of support vector machines (SVM), an unsupervised classifier in the form of a scoring algorithm based on Part-Of-Speech tagging [20] and keyword matching, and finally a hybrid method using both the scoring algorithm and SVM.

The remaining part of this document is organized as follows: Section 2 shows papers related to this study. Section 3 describes the materials and methods used, including a description of data, tools, and processing. Also, the four implemented methods are described in detail: NB, SVM, the scoring algorithm and the hybrid method based on scoring and SVM (called HS-SVM). In particular, this section details the implemented algorithms and optimal parametrization of the methods discussed. In addition, the evaluation and assessment of these classifiers is detailed. Section 4 shows the main results and their discussion. Finally, Section 5 deals with the conclusions and possible future work.

2. Related work

2.1. Sentiment analysis

Opinion mining systems development poses many challenges. First, it is necessary to identify text content. This is not an easy task due to the nature of language, which contains a great deal of semantic subtleties not present in other types of data. Second, sentiments must be classified in one way or another and thus determine their orientation. There are different ways to address this problem [26].

An opinion may be simply defined as a positive or negative sentiment, a viewpoint, an emotion or an appreciation about something or someone. In mathematical terms, an opinion is defined as a quintuple $(e_j, a_{jk}, so_{ijkl}, h_i, t_l)$ where e_j represents an entity and a_{jk} is the k-th characteristic of entity e_{ijkl} is the opinion sentiment value from the viewpoint of an opinion emitter h_i about the aspect a_{jk} of entity e_j in time t_l. This value may be positive, negative or neutral; even a more detailed range may be defined, for example different intensity levels [20].

Apart from sentiment and opinion, subjectivity and emotion are two other important related concepts in the area of opinion mining. A subjective sentence may express a personal sensation, a viewpoint or a belief; however, it does not necessarily involve a sentiment. A good classification of subjectivity may ensure a better sentiment classification [31], and this process can even be considered more complex than distinguishing positive, negative or neutral sentiments. On the other hand, an emotion may be considered as an expression of an individual's own subjective thoughts. Emotions are closely related to sentiments. In fact, the way the strength of an opinion is measured is associated with the intensity of certain emotions such as love, hate, surprise, anger, and sadness, among others.

Then, the objective of opinion mining is discovering all opinions in a quintuple $(e_j, a_{jk}, so_{ijkl}, h_i, t_l)$ in a document D. To attain this objective, it is necessary to do a series of tasks. The first one is extracting all the entities in D, where each entity e_j is unique. The second task is extracting all the aspects of the entities, where each aspect a_{jk} is unique for entity e_j. The third task is extracting an opinion and time t_l in which such opinion was written. The fourth task is determining the orientation so_{ijkl} of the opinion about one aspect. The final task is generating a representation of all the opinions in the form previously defined. The difficulty lies in the fact that not all these tasks can be totally solved [20].

Sentiment classification can be traditionally done in two ways: supervised and unsupervised based on semantics. The success of these techniques depends mainly on the appropriate extraction of the set of characteristics used to detect sentiments. The most used supervised techniques are support vector machines (SVM) and naïve Bayes (NB) classifier [32]. Machine learning solutions involve building classifiers from a collection of documents, where each text can be represented as a bag of words [27,47]. Also, it is common to use some stemming techniques and stop word elimination. In general, classifiers with a good behavior in the domain where they are trained do not show the same behavior in another domain since they are highly dependent on training data used [1]. Most of the literature is dedicated to domain specific solutions, and while there is much work towards cross domain opinion mining most solutions are domain dependant [15]. This article focuses on the domain of scientific paper reviews.

Unsupervised semantics-based methods use dictionaries in which different types of words are classified according to their semantic orientation [46]. Unlike traditional machine learning methods, semantics-based unsupervised methods are more dependent on their domain, although their performance may vary from one domain to another. There are two important sub-categories to mention: dictionary-based and corpus-based. The dictionary-based technique uses a set of initial terms usually manually collected. This set grows by looking up synonyms and antonyms. An example of this type of dictionary is WordNet, which was used for developing SentiWordNet [2]. The main drawback of this type

of approach is its inability to face the specific orientations of a domain and context. The corpus-based technique emerged with the purpose of providing dictionaries for a specific domain. These dictionaries result from a set of opinions seeds growing through the search of words related by means of statistical or semantic techniques such as Latent Semantic Analysis (LSA) or just by the frequency of occurrence of words within the collection of documents used [36].

Authors in [22] present a refined characterization of sentiment analysis techniques, including machine learning (supervised and unsupervised algorithms) and lexicon-based approaches (dictionary-based and corpus-based methods). In this review, supervised methods used for sentiment analysis include decision trees, support vector machines, neural networks, and methods based on probability, such as naive Bayes, Bayesian networks and maximum entropy.

A series of related papers is discussed below. Since there are no applications in the same domain, the domain of reviews or entity critique (e.g. films, hotels, products) is used as a reference since they are the closest among possible applications. This study is partially based on the work proposed by the authors in [49], where an opinion classification system of film reviews in Spanish is shown, using dependency parsing and POS tagging.

Table 1 shows results from different studies to determine polarity, starting with the seminal work from Pang et al. [27]. These results are shown with the purpose of providing a reference framework to evaluate results obtained. Table 1 focuses mostly on binary classification. Not all the papers shown in the table will be discussed, unless they are pertinent to our specific work. The strategy used is shown in the Approach column. It can be based on machine learning (ML), lexicon (L) or it may be hybrid (H). The area being worked out is shown in the Domain column. Most work is done on film critiques or Twitter. The values in the Results column are shown in terms of general accuracy, unless otherwise stated. The best results obtained for a certain paper are shown. If work involves doing tests on different data sets or with different class amounts, results will be reported separated by a slash (/) in the same order. The information in the table was obtained from the systematic reviews in [32,40]. The first paper deals with opinion mining as a whole, while the second one focuses on deep learning, a machine learning branch with different applications in opinion mining.

An effective sentiment analysis requires not only considering words individually, but also the linguistic construction of the sentence analyzed since it may totally change the sentiment expressed. The usual way of facing these constructions is by defining a heuristic. Authors in [27] work on film critiques and use a simple heuristic assuming that the negation scope includes words between the negator and the first punctuation after the negative term. Authors in [39] use data generated from the POS tagging process to identify the negation scope.

Apart from linguistic aspects, sentiment analysis must take into account the quality of the text analyzed. Furthermore, people make spelling and grammar mistakes. Some incorrectly written words were found during data processing. To solve these problems, spelling correctors may be used.

An important aspect in opinion mining is detecting sarcasm and irony. This is a complex task in this research field, particularly for the lack of agreement among researchers on how sarcasm and irony must be formally defined [33]. Another aspect is to detect unreliable spam or opinions that may distort analyses [20]. In comparison to other kind of reviews, research paper reviews do not include these aspects; so, they were not included in our analysis.

Research in the opinion mining area has greatly grown in the last decade, though most work focuses on texts written in English. While sentiment analysis in Spanish does not differ in essence with respect to English sentiment analysis, there is a lack of tools and libraries in comparison with English, which makes the implementation of sentiment analysis in Spanish more complex in general. Additionally,

Table 1
Results obtained in related works

Year	Approach	Domain	Result	Authors
2002	ML	Movie reviews	82.9%	[27]
2009	ML	Product reviews in English, Dutch and French	1. 83.30% 2. 69.80% 3. 67.68%	[4]
	L	Movie and product reviews translated to Spanish	71.81%	[6]
2011	L	Movie reviews	76.37%	[39]
	H	Twitter	85.40%	[51]
2012	ML	Forum comments	83.63%	[24]
2013	ML	Movie reviews	1. 85.40% 2. 45.70%	[38]
	H	Tourism product reviews	1. 85.50% 2. 75.50%	[21]
2014	H	Movie reviews	86.21%	[30]
	ML	Twitter	1. 87.61% 2. 70.40%	[43]
2015	ML	Reviews in Japanese	89.30% F_1	[37]
	ML	Movie and product reviews	1. 60.80% 2. 43.50%	[42]
	ML	Movie and product reviews	1. 67.60% 2. 45.30%	[41]
	ML	Movie reviews	86.50%	[19]
	L	Movie, hotel and product reviews	1. 78.50% 2. 80.11% 3. 89.38%	[49]
2016	ML	Movie and product reviews	1. 66.6% 2. 45.2%	[16]
	ML	Movie and product reviews	1. 75.8% 2. 63.6%	[50]
	H	Twitter	95.1%	[12]
	L	Movie reviews	90.1%	[8]

the Spanish language is less structured, compact and technical than English, which makes its semantic analysis difficult. Furthermore, only a small percentage of the research work is based on the Spanish language, with the vast majority of them focused on the English language.

Lexicon and grammar differences between Spanish and English may have an impact on the performance of systems trained for a certain language. Categorizing an opinion as positive, negative or neutral seems a simple task; however, it is really complex, particularly when opinions are written in different languages. Authors in [4] have studied the impact of English, German, and French particularities.

Some opinion mining studies focus on the Spanish language. One of the most relevant is proposed in [7]. It uses a semantics-based model defining a collection of dictionaries to calculate sentiments. Another study recently proposed in [48] describes an opinion mining system that classifies the orientation of Spanish texts taken from Twitter, according to an analysis of natural language, obtaining the syntactic sentence structure.

While works of sentiment analysis centered in movie reviews and product reviews are common in the literature, it must be mentioned that these domains of application are quite different from scientific paper reviews. An important difference is that peer reviews of research articles are an occluded genre (i.e. the documents are not publicly available) [14], contrary to movie reviews and product reviews that are intended for the general public.

Table 2
Attribute description for the paper reviews data set

Attribute	Description	Observations
Timespan	A date associated with the year of conference, which in turn corresponds with the time the review was written.	The data set includes four years of reviews worth of conferences.
Paper ID	This number identifies each individual paper from a given conference.	The data set has 172 different papers.
Review ID	A serial number identifier for each review as a correlative with respect to each individual paper (e.g. the second review of some paper would correspond to the number 2).	The data set has a total of 405 reviews. Most papers have 2 reviews each.
Text	Comments and detailed review of the paper. This is read by the authors and the editing commission of the conference. The editors determine if the paper should be published or not depending on the reviews.	There are 6 instances of empty reviews.
Remarks	Additional comments that can be read only by the editing commission of the conference. This is used in conjunction with the previous attribute to determine if the paper should be published.	This is an optional attribute. Whenever it is possible it is concatenated at the end of the main body of the review.
Language	Language corresponding to the review (it may be English or Spanish).	In this case the majority of the reviews are in Spanish, with only 17 instances of English reviews.
Orientation	Review classification defined by the authors of this study, according to the 5-point scale previously described, obtained through the authors' systematic judgement of each review.	This attribute represents the subjective perception of each review (i.e. how negative or positive the review is perceived when someone reads it).
Evaluation	Review classification as defined by the reviewer, according to the 5-point scale previously described.	This attribute represents the real evaluation given to the paper, as determined by the reviewers.

Another key difference is the vocabulary used, which due to the scientific background of the domain tends to formality. An important difference is that in terms of the use of language the reviewers tend to respect the respective rules of ortography and grammar, which facilitates the analysis in comparison with the other kind of reviews. In general, the main difference is the expected level of formality found throughout the text.

Furthermore, the interpretation of a paper review can be a difficult task because of the conflicting signals contained in the text [13]. Also, reviews contain requests for changes in the form of directions, suggestions, clarification requests and recommendations. Early career researchers tend to be more affected by this, since they lack the experience to adequately interpret the reviewers' comments [25].

Finally, it is important to remark that no publications using scientific paper reviews as a work domain for sentiment analysis have been found in the literature, except for our previous work [17] which in fact is a previous and shorter version of this extended article. So, this proposal for applying sentiment analysis is a novel contribution to this domain.

2.2. Motivation and potential applications

One of the common problems in scientific paper reviews is that the scores provided by reviewers can be inconsistent with what is written in the review. Particularly, there are cases in which reviewers are too strict, leading to the contradiction that, in reading the review, critiques are scarce, thus indicating that a paper was accepted, but in reading the reviewer's result, you may find that it was rejected. The problem can also be the opposite, that is, a reviewer makes substantive critiques while indicating that the paper must be accepted.

Concerning the problem above, consistency evaluation between the written review and the reviewers' score is proposed as a practical application of sentiment classification. For these reasons, the classifier

Table 3
Review data set statistics

Statistic	Number of words	Number of sentences
Min	3	1
Max	530	47
Avg	88.64	8.91
Std. Dev	69.76	7.54

used in this study was trained according to manual data tagging, not the reviewer's original classification. This allows revising the consistency between what the review states and what the reviewer says about the paper acceptance or rejection.

In this context, conducting a longitudinal evaluation of the consistency between the review and each reviewer's acceptance is proposed as future work. This evaluation must be done while keeping anonymity and giving each reviewer a numerical identifier so as to avoid revealing their true identity.

This work would allow classifying reviewers between strict (i.e., the score is always more negative than the review's critique) and non-strict (i.e., the score is always more positive than the review's critique). This classification can be applied in such a way that reviewers may be distributed equitably, thus guaranteeing that a good paper will not be rejected because reviewers are too strict and a poor paper will not be accepted because reviewers are not very strict.

The current system is used as a proof of concept, showing that it is possible to use automatic sentiment classification methods to determine review orientations. Certainly, the classification provided by the system is not expected to be consistent with the results given by the reviewers themselves. In fact, this is the consistency to be determined.

3. Materials and methods

3.1. Research data

The data set consists of paper reviews sent to an international conference in Spanish.[1] It has a total of $N = 405$ instances evaluated with a 5-point scale, expressing the reviewer's opinion about the paper ("-2": very negative, "-1": negative, "0": neutral, "1": positive, "2": very positive). The attributes of each instance in the data set are described in Table 2.

Empty reviews and reviews in English are not considered in the analysis. Table 3 shows a basic statistics summary concerning word count and number of sentences for the reviews in the data set.

Figure 1 shows the data distribution in terms of the classifications assigned by the authors when reviewing the content of each review, note that the data set is skewed. Figure 2 shows the data distribution in terms of the classifications assigned by original reviewers. The distribution of the original scores is more uniform in comparison to the revised scores. This difference is assumed to come from a discrepancy between the way the paper is evaluated and the way the review is written by the original reviewer.

The study focuses on classifying reviews according to the scale determined by the authors. Original evaluations will be used as complements for evaluating the consistency between the classification inferred from the text and the one assigned by the reviewer.

[1]The data set used in this study can be found in http://mii.ucn.cl/files/2814/8570/2080/reviews.json.

Table 4
Confusion matrix of evaluation (rows) vs orientation (columns)

Class	−2	−1	0	1	2	Total
−2	30	49	6	0	0	85
−1	3	44	11	0	0	58
0	2	29	23	5	0	59
1	0	10	41	29	1	81
2	0	5	23	57	14	99
Total	35	137	104	91	15	*382*

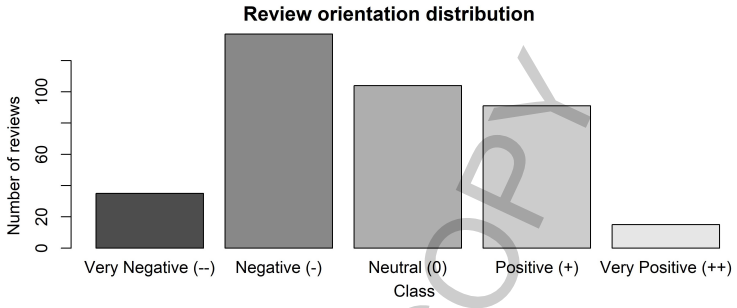

Fig. 1. Distribution of review qualifications (revised score).

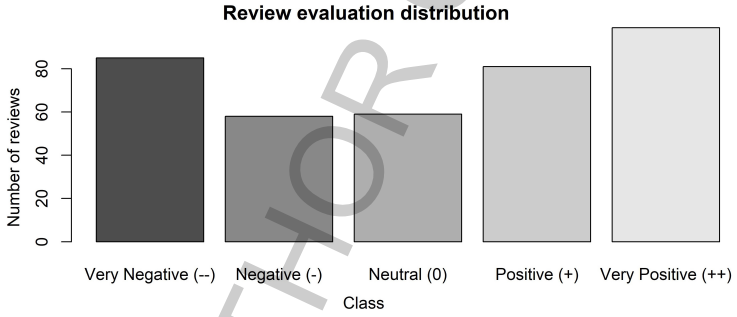

Fig. 2. Distribution of review qualifications (original score).

In Table 4 the relationship between the original evaluation scale and the orientation scale can be observed. There is a slight bias toward negative classes in the orientations when compared to the evaluations. Also, it is clear that the perceived orientation of the review is not completely aligned with the real evaluation. The accuracy of human evaluators is measured through their accuracy on predicting the real evaluation correctly. Considering all the five classes in the problem there is an accuracy rate of 36.65%. On the other hand, the accuracy rate is 65.45% if a ternary classification approach is taken.

The accuracy rates given in the previous paragraph serve as a reference. Indeed they can be seen as a baseline with which to compare the different results obtained with different techniques. In fact, it can be observed that for the five classes case the different methods have a similar behaviour, while there is a clear difference in favor of human classifiers in the ternary case.

3.2. Tools and resources

The following tools were used for developing an opinion classification system and making sentiment analysis:

1. Python programming language, version 2.7.
2. Scikit-learn library, its classifier implementations and evaluation methods [28].
3. *Stanford POS Tagger* library, particularly its model for processing text in Spanish [44]. This model uses the form proposed by the EAGLES group to tag words [18] in each sentence.
4. SentiWordNet 3.0 lexical ontology, containing semantic orientations and synonym sets in English [2]. A Spanish-translated version available in [29] was used. Some words and their translation were added to the original set because it was not complete.
5. Dictionaries specifying the semantics of the words. They were constructed by manually reviewing the data set and finding words that fit in each category. The following dictionaries were considered:
   - Positive words, e.g. "bueno" (good) and "innovador" (innovative).
   - Negative, e.g "malo" (bad/wrong), "deficiente" (deficient).
   - Adversative words, e.g. "pero" (but).
   - Amplifier words, e.g. "muy" (very).
   - Mitigator words, e.g. "poco" (few).
   - Suggestion words, e.g. "sugiero" (to suggest), "corregir" (to correct).
   - Negation words, e.g. "no", "nunca" (never).
6. A list of compound expressions that must be fused before processing the text, such as "sin embargo" ("nevertheless", "nonetheless", "however"), which is taken to be a single token in the form "sin_embargo").

3.3. Methods

Methods used in opinion mining are related to data extraction and preprocessing, natural language processing, and machine learning methods, which play a fundamental role in the task of determining the orientation of an opinion. A learning task may be divided into two broad approaches: supervised learning, in which classes are provided in data, and unsupervised learning, in which classes are unknown and the learning algorithm needs to automatically generate class values. Supervised methods naïve Bayes [3] and Support Vector Machines [20] were used. For the unsupervised learning task, an approach based on part-of-speech tagging and keyword matching was used. Furthermore, a hybrid approach [32] which combines both supervised and unsupervised methods is proposed.

Deep learning methods have not been tested due to the small size of the data set. While deep learning methods perform well in sentiment analysis [40], the number of parameters that must be estimated for deep learning to work well is too big for the amount of data present in this data set. Enlarging the data set is a difficult task since scientific reviews are an occluded genre [14] and as such getting access to more data is not easy. Gathering more reviews has been left for future work, and given this, the application of deep learning methods on this data set has been left for future work.

Figure 3 shows the high level architecture of the implemented system with the purpose of showing the general logic flow. Paper reviews are represented in a structured format using *json*. As part of the preprocessing step the raw data has been checked manually and corrections have been applied where needed. After reading the corrected data, another preprocessing step is needed before constructing the supervised and unsupervised classifiers. All the classifiers generate a report in text format that can be visualized by the final user.

3.3.1. Supervised methods: NB and SVM

NB classifier assumes that all attributes are conditionally independent, but this assumption is not

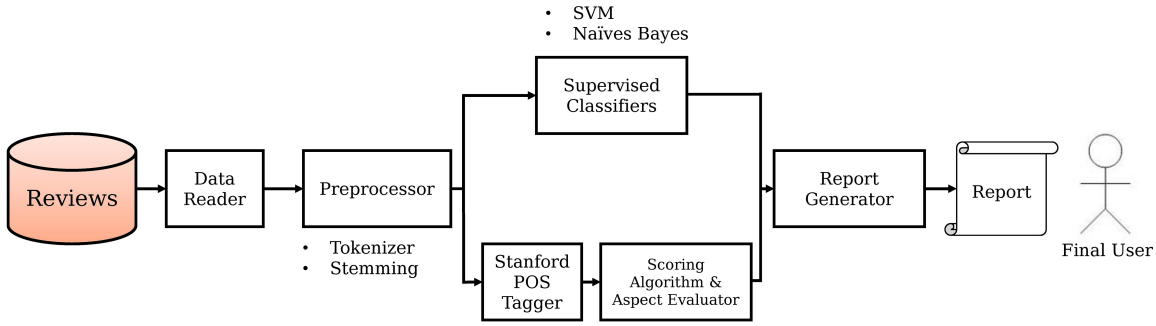

Fig. 3. High level diagram of the implemented methods.

generally achieved in practice. For example, words in a document are not independent among them. Despite this, researchers have shown that this method generates good models [20].

As for SVM, this approach has a sound theoretical basis and has empirically shown to be the most accurate classifier for text documents [20]. The classifier implemented by Pyhton *scikit-learn* library [28], *libsvm* implementation [9] was used. Particularly, a linear kernel was used because it rendered better results than other nuclei available in the library. The optimal classifier parametrization was obtained via empirical tests. The optimal parameter C found corresponds to $C = 1.5$ (values from 0.5 to 3.0 with 0.25 increments were used). Default parameters were used for the other configurable elements of the implementation because they provided good results.

For SVM, an output coding based on error correction codes [10] was used. This method is implemented in *sklearn* libraries and its performance was better than the *one vs. all* approach used by default for the implementation [28], obtaining a 10% improvement in terms of the average metric F_1-score. The selected code size is twice greater than the amount of classes. This parameter was selected via empirical performance evaluation (values from 0.5 to 3.0, with 0.25 increments were tested).

In both cases, the training of the classifiers was done by splitting the data set into a training set and a testing set with a 70% and 30% proportion, respectively.

3.3.2. Unsupervised methods: Part-Of-Speech tagging

Once the text is separated in tokens, the next step is usually made to conduct a morphosyntactic analysis to identify characteristics, for example, its grammatical category. This analysis is known as Part-Of-Speech (POS) tagging.

The method uses a text in a given language as input and, through the application of its internal POS tagging model, assigns a grammatical category to the words in a sentence, for example, verb and adjective, among others. In addition, each category has its own characteristics, for example, in Spanish verbs are characterized by tense and type of subject, which are not applicable to nouns.

The complexity of this task depends on the target language to be analyzed. For example, Spanish is more complex as to verb conjugation and implicit subjects. To apply this technique, preprocessing stemming is omitted because it may prevent obtaining the correct grammar structure.

POS tagging poses two main challenges: The first one is word ambiguity, which depends on the context of the sentence analyzed; the second one is assigning a grammatical category to a word when the system does not know how to do it. To solve both problems, the context around the word in a sentence is typically considered and the most probable is selected. The grammatical category has a relevant characteristic. A word belonging to the same word group can replace a token with the same grammatical category, without affecting the sentence grammatically [34].

Most tools to determine grammatical category only work in English, as a result it becomes necessary to find a POS tagging library that can handle Spanish. The *Stanford Log-linear Part-Of-Speech Tagger* [45] library was used. This library reads a text and assigns a grammatical category to each word. This library is implemented in Java (version 8) and provides models in six different languages, including Spanish.

3.4. Data preprocessing

Before classifying a text, it is necessary to process it. First, punctuation standardization is done, so that writing rules can be respected (for example, "The writing is awful,but the form is correct." would become "The writing is awful, but the form is correct." (now, there is a space after the comma)). Once this is done, the text is tokenized, separating it into sentences (according to the use of periods) and each sentence into words. Depending on each case, different preprocessing is done.

In the case of NB, punctuation marks and Spanish stopwords are eliminated because they do not provide any data for this classifier. A TF-IDF scheme is applied to the input text, this representation being Bayes classifier input. Similarly, in the case of SVM, punctuation marks and Spanish stopwords are eliminated. A TF-IDF scheme is applied to the input text; then, the singular value decomposition (SVD) method is applied, keeping 100 main values, this representation being SVM input. SVD is applied in order to reduce dimensionality, even though SVM is not sensitive to high dimensionalities, this reduction will reduce the computational cost of the method.

In the case of POS Tagging neither punctuation marks nor stopwords are eliminated because they contain useful data for the classifier (for example, negation). The text is then entered into Stanford POS Tagger in order to identify its semantic structure. Finally, a manual review is made to look for words (i.e. iterating over each word in the document) found in certain dictionaries so as to mark these instances with additional tags. This list of tokens and their associated tags corresponds to the unsupervised classifier input.

3.5. Scoring algorithm

To evaluate a review, Algorithm 1 is used over each sentence and then the average of all the sentences in the review are calculated.

The value produced by Algorithm 1 provides the semantic orientation of the review in terms of a continuous numeric scale. This result must be discretized to obtain the classification in the corresponding classes.

The binary classification method (classes "−1" and "1"), ternary classification (classes "−1", "0", and "1"), and 5-point scale multiclass classification (from "−2" to "2") were tested, obtaining different performances in each case due to their increasing complexity.

The algorithm was implemented by following a rule-based scheme, according to the semantic characteristics of words. Particularly, a dictionary-based approach combined with a series of heuristics was used, these heuristics consist of rules that define the effect of each type of word on the semantic orientation of a sentence.

First, each word is analyzed to be tagged according to its semantic characteristics (POS Tagging). In addition, the dictionaries mentioned previously were used to add other tags in each word. The dictionaries are listed below, they were used in order to specify the effect of each word on the semantic orientation of the sentence. Particularly, the general effect on the sentence, according to a series of pre-established rules, is calculated, depending on the word found and its semantic orientation. The strategy used in each case is similar to the one used in [49], though without using dependency parsing.

Algorithm 1 Scoring Algorithm

Require: TokenList, a list of tokens in a sentence; PosBias, an additional weight factor for positive words; NegBias, an additional weight factor for negative words.
Ensure: TotalScore, the semantic orientation value for the sentence.

```
1: function SCORESENTENCE
2:     TotalScore = 0
3:     PreviousTokens(2) = None
4:     Inverted = False
5:     TokenScore = 0
6:     for all (Token token in TokenList) do
7:         Tags = GetTags(Token)
8:         TokenScore = GetSentiWordNetScore(Token, Tags)
9:         if IsPositive(Tags) then
10:            TokenScore = TokenScore * PosBias
11:        else if IsNegative(Tags) then
12:            TokenScore = TokenScore * NegBias
13:        end if
14:        if Token == '?' then
15:            TokenScore = − QMOrientation
16:            Next Token
17:        end if
18:        if IsSuggestion(Tags) then
19:            TokenScore = − SuggestionOrientation
20:        end if
21:        if IsInversion(Tags) then
22:            Inverted = ¬ Inverted
23:        end if
24:        if Inverted then
25:            TokenScore = − TokenScore
26:        end if
27:        if IsVerb(Tags) and ContainsNo(PreviousTokens) then
28:            TotalScore = TotalScore − NegatedVerbOrientation
29:        end if
30:        if IsIncrement(PreviousTokens) then
31:            TokenScore = TokenScore * ModFactor
32:        end if
33:        if IsDecrement(PreviousTokens) then
34:            TokenScore = TokenScore/ModFactor
35:        end if
36:        if IsAdversative(Tags) then
37:            TotalScore = TotalScore * AdversativeWeight
38:        end if
39:        TotalScore = TotalScore + TokenScore
40:        Update PreviousTokens
41:    end forreturn TotalScore
42: end function
```


3.5.1. Word lists

Positive words: It contains the list of words considered positive in the problem domain. Its semantic orientation is obtained from SentiWordNet ontology (values from 0 to 1), specifying that the positive

value is required. In case the word is not in the ontology, a 0.5 default value is assumed (halfway between the minimum value of 0 and the maximum of 1, reflecting a moderately positive word).

For example, in the sentence "el artículo es innovador" ("the article is innovative") the word "innovative" would be in this dictionary, as it is a positive word, and this sentence would have a positive semantic orientation.

Negative words: It contains the list of words considered negative in the problem domain. Its semantic orientation is obtained from SentiWordNet ontology (values from 0 to -1), specifying that the negative value is required. In case the word is not in the ontology, a -0.5 default value is assumed (halfway between the minimum value of -1 and the maximum of 0, reflecting a moderately negative word).

For example, in the sentence "el artículo está mal escrito" ("the article is badly written") the word "badly" would be in this dictionary, as it is a negative word, and this sentence would have a negative semantic orientation.

Intensifiers: It contains the list of words increasing the intensity of the words that follow by a certain predefined factor. The intensification factor is 2.5, a value that was considered empirically appropriate (values from 1.1 to 3.0 were tested, with 0.1 increments, the value 2.5 was chosen taking the value that provided the best average F_1-score based on a sample of 10 runs per value).

For example, in the sentence "el artículo está muy bien escrito" ("the article is very well written") the word "very" would be in this dictionary, as it has the effect of intensifying the effect of the next word. So if the word "well" added 0.5 to the semantic orientation, after using the intensification factor it would now add 1.25. This sentence would in turn have a very positive semantic orientation. This is implemented by using the value *ModFactor* as can be seen in Algorithm 1, in this case, the value would be 2.5.

Mitigators: It contains the list of words that decrease the intensity of the words that follow by a certain predefined factor. The reduction factor is 0.4 (values from 0.1 to 0.9 were tested, with 0.1 increments, the value 0.4 was chosen taking the value that provided the best average F_1-score based on a sample of 10 runs per value).

For example, in the sentence "el artículo tiene pocos errores" ("the article has a few mistakes") the word "few" would be in this dictionary, as it has the effect of mitigating the effect of the next word. So if the word "error" subtracted 0.5 to the semantic orientation, after using the mitigation factor it would now subtract 0.2. This sentence would in turn have a slightly negative semantic orientation. This is implemented by using the value *ModFactor* as can be seen in Algorithm 1, in this case the value would be 2.5 (note that $1/2.5 = 0.4$).

Negation words: It contains the list of words that reverse the orientation of the words that follow (the semantic orientation value is multiplied by -1).

For example, in the sentence "el artículo no es bueno" ("the article is not good") the word "no" would be in this dictionary, as it has the effect of reversing the orientation. So if the word "good" added 0.5 to the semantic orientation, after using the negation factor it would now subtract 0.5. This sentence would in turn have a negative semantic orientation. The negation is implemented through the boolean value *Inverted* in Algorithm 1.

Adversative words: It contains the list of adversative words. These reduce the intensity of previous words, but they intensify the ones that follow. There are two types of adversative clauses (restrictive and exclusive) [49]. While there exist other types of conjunctions (e.g. coordinate, copulative or disjunctive), for simplicity only adversative conjunctions were considered and for the purposes of this study, only the restrictive case was considered. The reduction factor is 0.5 (value empirically found; values from 0.1 to 0.9 were tested, with 0.1 increments).

For example, in the sentence "la estructura está bien, pero tiene problemas de contenido" ("the structure is good, but the content has problems") the word "pero" would be in this dictionary, as it is an adversative clause. So if the word "good" added 0.5 to the semantic orientation and the word "problems" subtracted 0.5 to the semantic orientation, then after considering the adversative clause the word "good" would add 0.25, and then the whole sentence would have a semantic orientation of -0.25 instead of 0. This sentence would in turn have a slightly negative semantic orientation. This is implemented through the use of the factor *AdversativeWeight* in Algorithm 1, assigned to 0.5 in this case.

Suggestion words: It contains the list of words referring to a suggestion (for example, modal verbs like "should" and "could" and other verbs like "improve", "correct"). Modal verbs are very important due to the fact that they are emotional words giving either positive or negative polarity in the sentence. However, for this particular domain, these words are considered to have an always negative orientation that must be subtracted from the sentence score, however, they have a lesser impact in comparison to regular negative words.

Usually, reviews that suggest direct rejection tend to use discourse units with the function of negative evaluation, while reviews that suggest a major revision of the article use discourse units with the function of recommendation [35]. Based on this, the score of a recommendation, while slightly negative in the sense that it implies that the paper must be improved, has a lower impact than a direct negative evaluation. The suitable empirical value was found to be -0.025 (value empirically found; it was tested from -0.5 to 0.0, with 0.025 increments).

For example, in the sentence "sugiero mejorar la estructura" ("I suggest improving the structure") the word "suggest" would be in this dictionary, as it implies a suggestion and something that must be improved. So this sentence would now have a semantic orientation of -0.025, which would correspond to a slightly negative semantic orientation. This is implemented by the value *SuggestionOrientation* which is assigned to 0.025 in Algorithm 1.

3.5.2. Heuristics

1. If a **question mark** is found in the review, it causes a slight predefined negative impact, regardless of the context, which must be subtracted from the sentence score. Its impact is -0.1 over the score, a value empirically found (values from -0.5 to 0.0 were tested, with 0.05 increments). Note that exclamation marks were not considered due to the fact that they are not frequently used (only 6 reviews contain exclamation marks compared to 87 reviews containing question marks). While it is clear that they would have an effect in the semantic orientation similar to intensifiers, in this particular domain they're not frequently used, at least according to this data set. The absence of exclamation marks might be due to the fact that the reviewers tend to a neutral formality in their writing style. This is reflected in Algorithm 1 by assigning the value *QMOrientation* to 0.05.
2. If a **negation adverb** is found ("not") and it is followed by a verb, it has a strong predefined negative impact. The scope of the negation is considered to be up to three tokens after the adverb, based on the conservative heuristic used by Fernández Anta et al. [11]. To detect these patterns, POS tags are used. This action is done regardless of the context and must always be subtracted from the sentence score (for example, "does not show", "does not present", "does not focus on the topic", and "do not contribute", among others). Its impact is -0.5 over the score, a value also empirically found (values from -1.0 to -0.1 were tested, with 0.1 increments). Note that this is not the same as regular negation, since the other words in themselves do not represent a positive or negative opinion, however, the combination of the word "not" and these verbs indicate that the paper lacks something, and therefore it reflects a negative opinion from the reviewer. This is

implemented through the use of the function *ContainsNo*(*PreviousTokens*) in Algorithm 1 that takes the previous three tokens and determines if the current verb is in the scope of a negation, in case this is true then the fixed value of *NegatedVerbOrientation* is subtracted from the total score (0.5 in this case).

3. **Bias parameters** were included to strengthen positive and negative words. Since most reviews are likely to be critiques, it may be useful to include a bias towards positive opinion to compensate for the natural negativity. Movie reviews present similar behaviour, and bias parameters have been found to be useful [49]. Given this, bias of 10% was included, favoring positive words and multiplying its score by a 1.1 factor. This value was empirically found, other values that were tested are 5%, 15%, and 20%. In Algorithm 1 this heuristic is reflected through *PosBias* and *NegBias*, in this case $PosBias = 1.1$ and $NegBias = 1.0$, this means that negative words are not affected but positive words are stronger.
4. Finally, in case the word is not included in a dictionary (the list of words, not the ontology), it is assumed that it does not have any effect in this domain. So, its score is assigned to 0, under the assumption that it will have no effect.

The list of previous heuristics could be refined. Nevertheless, the results obtained with them are satisfactory since the result improved compared to the baseline performance obtained for our classifiers without using heuristics.

3.5.3. Classification with the scoring algorithm

Algorithm 1 produces continuous values that can be positive or negative. Nevertheless, the objective is to obtain the semantic orientation in terms of the classes defined above. Thus, Algorithms 2–4 must be used for binary, ternary and the five-point classification. For this purpose, the parameter values (*DoublePositiveThreshold*, *DoubleNegativeThreshold*, *NegativeThreshold* y *PositiveThreshold*) were obtained by applying Monte Carlo simulation, testing a series of value ranges between -1.0 and 1.0 and using the combination with the best performance.

Algorithm 2 Score-based Binary Classification

Require: Score, value given by the scoring algorithm (Algorithm 1).
Ensure: Class: positive or negative.

```
1: function BINARYSCORECLASSIFICATION
2:     if Score ⩾ 0 then return "1"
3:     elsereturn "−1"
4:     end if
5: end function
```


3.6. Hybrid method: HS-SVM

Another method based on the scoring algorithm and support vector machines is proposed for classification in this domain. The method has been named Hybrid Scoring Support Vector Machine (HS-SVM), in reference to the fact that it is a hybrid method that uses the scoring algorithm proposed in the previous section. This is a hybrid method of sentiment analysis since it combines a supervised classifier (SVM) and an unsupervised classifier (Scoring algorithm) to obtain the final class. The preprocessing steps for this new method are the same ones used for the original classifiers. Figure 4 shows the proposed method's components and flow.

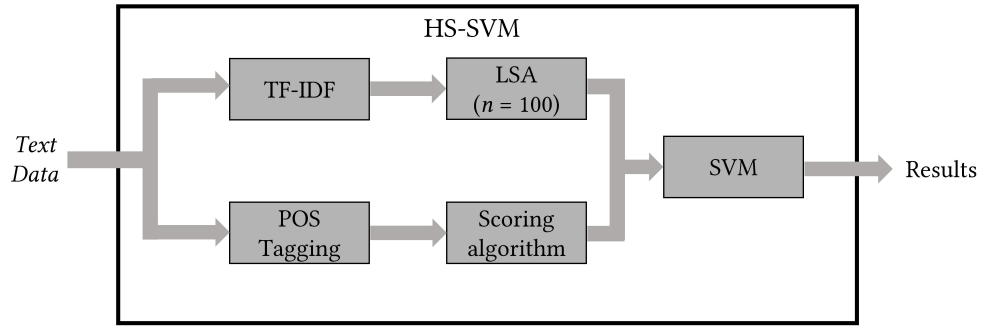

Fig. 4. Hybrid Scoring Support Vector Machine components and flow.

Algorithm 3 Score-based Ternary Classification

Require: Score, value given by the scoring algorithm (Algorithm 1).
Ensure: Class: positive, negative or neutral.

```
function MULTICLASSSCORECLASSIFICATION
    if Score > PositiveThreshold then return "1"
    else if Score < NegativeThreshold then return "−1"
    elsereturn "0"
    end if
end function
```

Algorithm 4 Score-based 5-point Scale Classification

Require: Score, value given by the scoring algorithm (Algorithm 1).
Ensure: Class: very positive, positive, negative, very negative or neutral.

```
function FULLSCALESCORECLASSIFICATION
    if Score > DoublePositiveThreshold then return "2"
    else if Score > PositiveThreshold then return "1"
    else if Score < DoubleNegativeThreshold then return "−2"
    else if Score < NegativeThreshold then return "−1"
    elsereturn "0"
    end if
end function
```


The score works as a new feature for the SVM's input data. The SVM is then trained with this additional feature. This proposed approach has the advantage of having the information provided by the scoring algorithm and its associated components and the flexibility of the SVM. However, it has a higher computational cost since it requires the usage of the scoring algorithm and training the SVM classifier. Nevertheless, since the data set for this application is sufficiently small, this drawback has no significant effect.

3.7. Aspect evaluator

Reviewer comments can have different functions, and they can be more directed towards the technical content, the general readibility or the structural aspect of the paper itself [14]. So while there are many aspects that could be evaluated, for example the opinion of the reviewer on the validity of the claims in

the article or the discussion itself, it is simpler to evaluate textual aspects such as the format or writing rather than the content itself, since the latter requires certain knowledge of the domain of the reviewed article. Given this, a list of five important aspects considered when reviewing a paper was constructed. The evaluated aspects are listed below:

1. References
2. Format
3. Structure
4. Writing

Evaluation consists in looking for references to these aspects (or their synonyms) in a sentence. A score is assigned to each sentence using Algorithm 1. The search of synonyms is done by using SentiWordNet synonym sets or synsets [2].

Algorithm 5 Aspect Evaluator Algorithm

Require: TokenList, a list of tokens in a sentence.
Ensure: AspectScore, an array with 4 positions as inputs for the basic aspects defined as: Writing, Format, References and Structure.

```
1:  function AspectScoreSentence
2:      AspectScores = Initialize()
3:      CumulativeScore = 0
4:      for all (Token token in TokenList) do
5:          Tags = GetTags(Token)
6:          TokenScore = EvaluateToken(Token, Tags)
7:          CumulativeScore = CumulativeScore + TokenScore
8:          if IsAdjective(Tags) or IsVerb(Tags) or IsNoun(Tags) then
9:              for all Aspect in BaseAspects do
10:                 if IsSynonymous(Token, Aspect) then
11:                     AspectScores[Aspect] = AspectScores[Aspect] + CumulativeScore
12:                 end if
13:             end for
14:         end if
15:         if IsAdversative(Tags) then
16:             CumulativeScore = 0
17:         end if
18:     end forreturn AspectScore
19: end function
```

A vector containing the scores of each aspect is initialized in zero. As the algorithm evaluates the sentence tokens, POS tags are used to check if the current token is an adjective, a verb or a noun. These three tags were considered because an adjective and a verb may implicitly correspond to one aspect (e.g., "do not refer" or "well written"). If they correspond to one of these tags, they are checked to see if they agree with one of the aspects defined in the list. If all previous conditions apply, the current sentence score is added to the score of the associated aspect.

If an adversative clause is found, the current accumulated score is saved and a new accumulator is initialized because the use of these expressions marks the beginning of a different semantic orientation and the accumulation of previous values may affect the accuracy of results. The algorithm then continues its calculations using the new accumulator. Once the algorithm finishes the analysis of the sentence, the final score is the sum of the old accumulator and the new accumulator.

Table 5
Classification results for orientation in the binary case

Average summary	Accuracy	Precision	Recall	F1
NB	0.68 ± 0.05	0.67 ± 0.06	0.68 ± 0.05	0.64 ± 0.06
SVM	0.7 ± 0.05	0.7 ± 0.05	0.7 ± 0.05	0.69 ± 0.06
Score	0.81	0.81	0.81	0.81
HS-SVM	0.79 ± 0.05	0.8 ± 0.05	0.79 ± 0.05	0.79 ± 0.05

In the final implementation, the scoring and aspect evaluation algorithms were considered as one function, for the sake of simplicity.

4. Results and discussion

This section shows the results obtained with the implemented methods. First, the results from the orientation classification task are discussed, followed by the results of the evaluation classification task. Then, the results obtained from the aspect evaluator are provided.

To evaluate the classifier standard machine learning and pattern recognition metrics for classification problems are applied. In particular, we use *accuracy*, *precision*, *recall* and the F_1*-score*. These evaluation metrics have been selected because they are the most commonly applied metrics in the state-of-the-art and related work.

Evaluation metrics are provided as an average over each class, along with the corresponding standard deviation considering 10 replications, except in the case of the scoring algorithm, which is evaluated over all the data set and always provides the same result since it is deterministic (results only depend on parameters).

4.1. Orientation classification

The results provided here originate from using the methods to classify the orientation of each review (i.e. the perceived evaluation). Table 5 shows the classification results for binary classification, Table 6 shows the results for ternary classification and finally Table 7 shows the results for the 5-point scale classification.

In the binary case, performance is similar regarding the results from other studies (as shown in Table 1). The best average performance is obtained with the scoring algorithm, followed by HS-SVM, pure SVM and NB.

The amount of data available for the binary classification case is smaller than the amount of data for the multiclass case because the neutral reviews of the data set are not used. One of the main problems in comparison with other studies is the scarce amount of data available. A much better performance may be expected with a greater amount of instances.

In the case of ternary classification, average performance decreases in all metrics. This performance reduction is due to the greater classification complexity inherent to a problem with more classes. If the classifier were to work as a random selection it would only have 33.3% probability of predicting correctly. So, in comparison to that baseline, the classifiers still have a good quality. However, it is interesting to note that in this case, the best results are obtained with the HS-SVM classifier, which now surpasses the scoring algorithm itself.

In the case of the 5-point scale classification, the scoring algorithm is slightly better than the supervised methods and the HS-SVM approach surpasses all the other methods in this case, just as it did in

Table 6
Classification results for orientation in the ternary case

Average summary	Accuracy	Precision	Recall	F1
NB	0.46 ± 0.03	0.42 ± 0.05	0.46 ± 0.03	0.41 ± 0.05
SVM	0.48 ± 0.05	0.46 ± 0.06	0.48 ± 0.06	0.46 ± 0.06
Score	0.51	0.58	0.51	0.52
HS-SVM	0.56 ± 0.04	0.54 ± 0.04	0.56 ± 0.04	0.54 ± 0.04

Table 7
Classification results for orientation in the 5-point scale case

Average summary	Accuracy	Precision	Recall	F1
NB	0.35 ± 0.03	0.3 ± 0.04	0.35 ± 0.03	0.3 ± 0.04
SVM	0.4 ± 0.03	0.38 ± 0.04	0.41 ± 0.03	0.37 ± 0.03
Score	0.41	0.5	0.41	0.4
HS-SVM	0.46 ± 0.05	0.45 ± 0.06	0.46 ± 0.05	0.43 ± 0.05

the ternary case. According to these results, the use of this hybrid approach has better classification performance in the multiclass case, while in the binary case it is only slightly behind the scoring algorithm. In this sense, this method is considered to be more robust in relation to an increase in the number of classes.

There were problems with classifying very negative reviews with the scoring algorithm (and in general), in particular, if the lower threshold for the scoring algorithm classification is increased, examples of a very negative class can be correctly classified; however, some negative examples will also be incorrectly classified.

One of the main issues that may affect classification results for the supervised case is that these classifiers do not take into account text structure. They only consider the appearance of words according to the TF-IDF scheme described in the data preprocessing section.

The poor performance of SVM on this multiclass data set may be due to the fact that this classifier is highly sensitive to class imbalance [23]. And as Fig. 1 shows, this data set is highly skewed. So, in a sense, the obtained results by SVM on that data set could not be reliable.

Better results could be obtained with the scoring algorithm by improving the heuristics used or applying parsing dependency [49]. Nevertheless, results are considered satisfactory, since in all the metrics this method surpasses the other approaches.

The performance improvement with respect to the pure SVM approach is consistent in all the cases. The method works by adding more information to SVM, basically facilitating the classification process. SVM is helped by the heuristics defined for the scoring algorithm.

This method could also be combined with the results obtained for the aspects of each review. In this approach, the use of the scoring algorithm and aspect evaluation could be considered as an additional preprocessing stage. This stage would have the function of calculating additional text characteristics to facilitate the classification process by supervised methods.

This combined approach may be used for generalizations in other opinion mining cases. It would be interesting to evaluate if similar improvements may be made in other domains. Certainly, it would be necessary to adapt and modify scoring algorithms and aspect evaluation, and probably obtain a new set of optimal parameters.

Adding a hierarchical classification approach may improve results, by first filtering neutral reviews, then applying binary classification, and later applying an approach on positive and negative sets to separate very negative/positive examples from those only negative/positive.

Table 8
Classification results for evaluation in the binary case

	Binary			
Average summary	Accuracy	Precision	Recall	F1
NB	0.56 ± 0.04	0.58 ± 0.04	0.56 ± 0.04	0.56 ± 0.04
SVM	0.67 ± 0.04	0.67 ± 0.04	0.67 ± 0.03	0.67 ± 0.04
Score	0.7	0.73	0.7	0.69
HS-SVM	0.71 ± 0.04	0.72 ± 0.04	0.71 ± 0.04	0.71 ± 0.04

Table 9
Classification results for evaluation in the ternary case

Average summary	Accuracy	Precision	Recall	F1
NB	0.46 ± 0.04	0.45 ± 0.04	0.46 ± 0.04	0.44 ± 0.04
SVM	0.56 ± 0.04	0.53 ± 0.05	0.56 ± 0.04	0.53 ± 0.03
Score	0.46	0.62	0.46	0.5
HS-SVM	0.59 ± 0.02	0.56 ± 0.03	0.59 ± 0.02	0.57 ± 0.02

Table 10
Classification results for evaluation in the 5-point scale case

Average summary	Accuracy	Precision	Recall	F1
NB	0.23 ± 0.02	0.27 ± 0.04	0.23 ± 0.02	0.24 ± 0.03
SVM	0.33 ± 0.05	0.35 ± 0.04	0.33 ± 0.05	0.33 ± 0.04
Score	0.27	0.55	0.27	0.24
HS-SVM	0.37 ± 0.06	0.38 ± 0.06	0.37 ± 0.06	0.36 ± 0.06

4.2. Evaluation classification

The results provided here are obtained from executing the methods to classify the evaluation of each review (i.e. the original score given by the reviewers). Table 8 shows the classification results for binary case, Table 9 shows the results for the ternary case and Table 10 shows the results for the 5-point scale classification.

In general, maximum possible performance decreases. Although the obtained results are still acceptable since they are better than a random selection, they show that properly classifying the instances is more complex if the original scores provided by each reviewer are used instead of the orientation scores. This discrepancy results from the fact that reviewers do not usually provide scores agreeing with what is actually written in the review.

It is important to note that the parametrization of the scoring algorithm was not adjusted, retaining the original one designed for orientation classification. While this reduces classification accuracy and all associated metrics, this method is still competitive with the baseline methods (NB and SVM), and even those are still surpassed by the scoring algorithm classification in the binary case.

On the other hand, HS-SVM obtains the best results in comparison to the other methods. This stems from the flexibility provided by its SVM component, while at the same time benefiting from all the information provided by the scoring method. In general, according to the results of these experiments, HS-SVM surpasses the other methods, both in the evaluation classification task and in the orientation classification task.

4.3. Aspect evaluation

Table 11 summarizes the results for each aspect and it also shows the distribution of the aspects with

Table 11
Summary of results for aspect evaluation

Aspects	References	Format	Structure	Writing	Average aspects
1 (>0)	105	31	69	94	74.75
0 ($=0$)	192	336	276	246	250
-1 (<0)	85	15	87	42	57.25
Average	0.006	0.019	-0.020	0.077	0.021

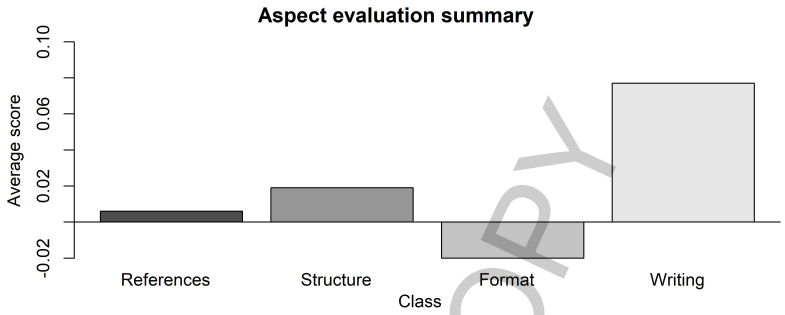

Fig. 5. Aspect average bar graph.

respect to its orientation (positive, negative or neutral). The results shown correspond to the average values, rounded to the third significant decimal.

Correctness evaluation becomes more complex because there is no previous tagging of these scores. Based on the results obtained, there are more positive than negative evaluations in almost all the defined aspects. The average of the values obtained is positive in all aspects, except for the one concerning structure. However, the majority of the reviews is considered neutral towards most of these aspects. The neutral ones are usually the result of not mentioning this aspect in the review. Considering this, it must be noted that references themselves are the most mentioned aspect according to these results.

A manual data review shows that the behavior observed may be due to the fact that one of the main problems arises in aspects referring to the structure of the papers evaluated. In addition, several reviews consider writing and discourse as good, even if the content or other aspects are negatively characterized. A graph with average aspect values is shown in Fig. 5 to illustrate the differences between the aspects.

The format aspect is the least mentioned one in comparison with the other ones when considering the number of zero scores. The aspect most commented by reviewers is references. This makes sense because it is in agreement with the logical demands of a scientific paper, where the validity of the content is generally more important than the format itself.

On average, the results obtained for the set of papers is positive; however, the approach used is far from being optimal because there is no mechanism to automatically obtain the paper aspects. So, there certainly are interesting elements which were not considered. Nonetheless, the aspects defined may include the main evaluation criteria when reviewing a paper, without considering the content and its contribution.

There is a possibility to enlarge the classifiers implemented. Particularly, the scores of each aspect could be used as additional input for the classifier. Although they are calculated by following the same scheme as the general score, these could provide more information to the classifier, as an extension to what was done in the HS-SVM method.

Finally, the methods implemented here could be applied in similar sentiment analysis domains, such as other kind of reviews (e.g., movies, hotels or products). However, this would entail adapting some

of the dictionaries used in Algorithm 1. For example, positive and negative words may vary from one domain to another, but adversative clauses should remain the same.

5. Conclusions

This article has studied the application of sentiment analysis techniques in the domain of paper reviews. Specifically, it has applied supervised methods (NB and SVM), an unsupervised method (the scoring algorithm) and a hybrid approach (HS-SVM) in the classification of 382 (non-empty Spanish) reviews of research papers of an international conference.

The best performance is obtained with binary classification, corresponding to the simplest version of the problem studied. Performance gradually decreases as more classes are added (such as the neutral one or those corresponding to extreme values). In this sense, the HS-SVM method is more robust than the others in relation to the number of classes.

One of the most interesting results is improvement obtained by the combination of the scoring algorithm and SVM. Basically, the score gives additional information to the SVM to facilitate the classification. Future work could deal with the extension and generalization of this method, also including the scores obtained for the aspects so as to further improve performance. By adding new semantic information (e.g. the score) to traditional machine learning methods, an improvement is expected to be obtained in the results of sentiment classification as compared with a pure method.

In the future, the algorithm performance to obtain the scores of each aspect must be evaluated. Its results were analyzed by observing those obtained in each review and the general average, but there is no specific metric as in the other methods evaluated. To better evaluate these results, it is necessary to have the tags for each aspect. These should be manually obtained in analyzing each review, although the weakness of this study is its subjectivity. So, automatic forms of generating tags for each aspect could be explored.

With respect to possible modifications of the models, one of the factors that could be considered in future work is individual reviewer bias (i.e. the reviewer may have a tendency to evaluate the papers lower or higher than the mean). In order to account for this bias, the current model would need to be modified. Also, another aspect that could be studied is an adequate handling of multi-lingual reviews, as well as the search of an appropriate parametrization in this case.

Concerning the experimental results, it is necessary to enlarge the list of features with more lexico-grammatical features, so that classifiers perform better and improved classification results are acquired. Also, expanding the data set with more reviews would be useful in future research, since the current data set is too small to apply some techniques that require more data to perform well.

As to the applicability of the proposal, future work could deal with the longitudinal evaluation of consistency between the review and the acceptance or rejection of the paper by each reviewer. This may allow a better evaluation of papers since it would be possible to recognize whether a reviewer is strict or not. Finally, since there are no other papers using scientific paper reviews as an application domain, the proposal in this study is a contribution and innovation for the field of sentiment analysis and opinion mining.

References

[1] A. Aue and M. Gamon, Customizing sentiment classifiers to new domains: A case study, in: *Proceedings of Recent Advances in Natural Language Processing (RANLP)*, Vol. 1, 2005, pp. 2–1.

[2] S. Baccianella, A. Esuli and F. Sebastiani, Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining, in: *LREC*, Vol. 10, 2010, pp. 2200–2204.
[3] C. Bishop, *Pattern Recognition and Machine Learning*, Springer-Verlag New York, 2006.
[4] E. Boiy and M.-F. Moens, A machine learning approach to sentiment analysis in multilingual web texts, *Inf. Retr.* **12**(5) (Oct. 2009), 526–558.
[5] L. Bornmann, Scientific peer review, *Annual Review of Information Science and Technology* **45**(1) (2011), 197–245.
[6] J. Brooke, M. Tofiloski and M. Taboada, Cross-linguistic sentiment analysis: From english to spanish, in: *RANLP*, 2009, pp. 50–54.
[7] J. Brooke, M. Tofiloski and M. Taboada, Cross-linguistic sentiment analysis: From english to spanish, in: G. Angelova, K. Bontcheva, R. Mitkov, N. Nicolov and N. Nikolov, eds, *RANLP*, RANLP 2009 Organising Committee/ACL, 2009, pp. 50–54.
[8] E. Cambria, Affective computing and sentiment analysis, *IEEE Intelligent Systems* **31**(2) (2016), 102–107.
[9] C.-C. Chang and C.-J. Lin, LIBSVM: A library for support vector machines, *ACM Transactions on Intelligent Systems and Technology* **2**(27) (2011), 1–27. Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.
[10] T.G. Dietterichv and G. Bakiri, Solving multiclass learning problems via error-correcting output codes, Journal of Artificial Intelligence Research, 1995, 263–286.
[11] A. Fernández Anta, P. Morere, L.F. Chiroque and A. Santos, Techniques for sentiment analysis and topic detection of spanish tweets: preliminary report, 2012.
[12] M. Ghiassi, J. Skinner and D. Zimbra, Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network, *Expert Systems with Applications* **40**(16) (2013), 6266–6282.
[13] H. Gosden, Thank you for your critical comments and helpful suggestions: Compliance and conflict in authors' replies to referees' comments in peer reviews of scientific research papers, *Ibérica: Revista de la Asociación Europea de Lenguas para Fines Específicos (AELFE)* (3) (2001), 3–17.
[14] H. Gosden, 'Why not give us the full story?' Functions of referees' comments in peer reviews of scientific research papers, *Journal of English for Academic Purposes* **2**(2) (2003), 87–101.
[15] O.L. Hasna, F.C. Măcicăşan, M. Dînşoreanu and R. Potolea, Modeling sentiment polarity with meta-features to achieve domain-independence, in: *International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management*, Springer, 2014, pp. 212–227.
[16] A. Joulin, E. Grave, P. Bojanowski and T. Mikolov, Bag of tricks for efficient text classification, *arXiv preprint arXiv:1607.01759*, 2016.
[17] B. Keith, E. Fuentes and C. Meneses, A hybrid approach for sentiment analysis applied to paper reviews, 2017.
[18] G. Leech, R. Barnett and P. Kahrel, Guidelines for the standardization of syntactic annotation of corpora, *EAGLES Document EAG-TCWG-SASG/1.8*, 1996.
[19] C. Li, B. Xu, G. Wu, S. He, G. Tian and Y. Zhou, Parallel recursive deep model for sentiment analysis, in: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, 2015, pp. 15–26.
[20] B. Liu, *Web data mining: exploring hyperlinks, contents, and usage data*, Springer Science & Business Media, 2011.
[21] E. Marrese-Taylor, J.D. Velásquez, F. Bravo-Marquez and Y. Matsuo, Identifying customer preferences about tourism products using an aspect-based opinion mining approach, *Procedia Computer Science* **22** (2013), 182–191.
[22] W. Medhat, A. Hassan and H. Korashy, Sentiment analysis algorithms and applications: A survey, *Ain Shams Engineering Journal* **5**(4) (2014), 1093–1113.
[23] A. Mountassir, H. Benbrahim and I. Berrada, An empirical study to address the problem of unbalanced data sets in sentiment classification, in: *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE, 2012, pp. 3298–3303.
[24] J. Ortigosa-Hernández, J.D. Rodríguez, L. Alzate, M. Lucania, I. Inza and J.A. Lozano, Approaching sentiment analysis by using semi-supervised learning of multi-dimensional classifiers, *Neurocomputing* **92** (2012), 98–115.
[25] B. Paltridge, Referees' comments on submissions to peer-reviewed journals: When is a suggestion not a suggestion? *Studies in Higher Education* **40**(1) (2015), 106–122.
[26] B. Pang and L. Lee, Opinion mining and sentiment analysis, *Found. Trends Inf. Retr.* **2**(1-2) (Jan. 2008), 1–135.
[27] B. Pang, L. Lee and S. Vaithyanathan, Thumbs up?: Sentiment classification using machine learning techniques, in: *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing – Volume 10*, EMNLP '02, Stroudsburg, PA, USA, 2002, pp. 79–86. Association for Computational Linguistics.
[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* **12** (2011), 2825–2830.
[29] E. Peinado, Sentiwordnet-bd. https://github.com/rmaestre/Sentiwordnet-BC, 2013.
[30] S. Poria, E. Cambria, G. Winterstein and G.-B. Huang, Sentic patterns: Dependency-based rules for concept-level sentiment analysis, *Knowledge-Based Systems* **69** (2014), 45–63.

[31] S. Raaijmakers and W. Kraaij, A shallow approach to subjectivity classification, in: *2nd International Conference on Weblogs and Social Media, ICWSM 2008, 30 March–2 April 2008, Seattle, WA, USA*, Association for the Advancement of Artificial Intelligence AAAI, 2008, pp. 216–217.
[32] K. Ravi and V. Ravi, A survey on opinion mining and sentiment analysis: Tasks, approaches and applications, *Knowledge-Based Systems* **89** (2015), 14–46.
[33] A. Reyes and P. Rosso, Making objective decisions from subjective data: Detecting irony in customer reviews, *Decis. Support Syst.* **53**(4) (Nov. 2012), 754–760.
[34] M. Rodrigues and A. da Silva Teixeira, *Advanced Applications of Natural Language Processing for Performing Information Extraction*. SpringerBriefs in Electrical and Computer Engineering. Springer International Publishing, 2015.
[35] B. Samraj, Discourse structure and variation in manuscript reviews: Implications for genre categorization, *English for Specific Purposes* **42** (2016), 76–88.
[36] J. Serrano-Guerrero, J.A. Olivas, F.P. Romero and E. Herrera-Viedma, Sentiment analysis: A review and comparative analysis of web services, *Information Sciences* **311** (2015), 18–38.
[37] H. Shi, W. Zhan and X. Li, A supervised fine-grained sentiment analysis system for online reviews, ntelligent Automation & Soft Computing, (ahead-of-print), 2015, 1–17.
[38] R. Socher, A. Perelygin, J.Y. Wu, J. Chuang, C.D. Manning, A.Y. Ng, C. Potts et al., Recursive deep models for semantic compositionality over a sentiment treebank, in: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Vol. 1631, Citeseer, 2013, p. 1642.
[39] M. Taboada, J. Brooke, M. Tofiloski, K. Voll and M. Stede, Lexicon-based methods for sentiment analysis, *Comput. Linguist.* **37**(2) (June 2011), 267–307.
[40] D. Tang, B. Qin and T. Liu, Deep learning for sentiment analysis: successful approaches and future challenges, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **5**(6) (2015), 292–303.
[41] D. Tang, B. Qin and T. Liu, Document modeling with gated recurrent neural network for sentiment classification, in: *EMNLP*, 2015, pp. 1422–1432.
[42] D. Tang, B. Qin and T. Liu, Learning semantic representations of users and products for document level sentiment classification, in: *ACL (1)*, 2015, pp. 1014–1023.
[43] D. Tang, F. Wei, B. Qin, T. Liu and M. Zhou, Coooolll: A deep learning system for twitter sentiment classification, in: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 2014, pp. 208–212.
[44] K. Toutanova, D. Klein, C.D. Manning and Y. Singer, Feature-rich part-of-speech tagging with a cyclic dependency network, in: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, Association for Computational Linguistics, 2003, pp. 173–180.
[45] K. Toutanova and C.D. Manning, Enriching the knowledge sources used in a maximum entropy part-of-speech tagger, in: *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora: Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*, Association for Computational Linguistics, 2000, pp. 63–70.
[46] P.D. Turney. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews, in: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, Stroudsburg, PA, USA, 2002, pp. 417–424. Association for Computational Linguistics.
[47] P.D. Turney, P. Pantel et al., From frequency to meaning: Vector space models of semantics, *Journal of Artificial Intelligence Research* **37**(1) (2010), 141–188.
[48] D. Vilares, M.A. Alonso and C. Gómez-Rodríguez, A linguistic approach for determining the topics of spanish twitter messages, *J. Inf. Sci.* **41**(2) (Apr. 2015), 127–145.
[49] D. Vilares, M.A. Alonso and C. Gomez-Rodriguez, A syntactic approach for opinion mining on spanish reviews, *Natural Language Engineering* **21**(1) (2015), 139–163.
[50] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola and E. Hovy, Hierarchical attention networks for document classification, in: *Proceedings of NAACL-HLT*, 2016, pp. 1480–1489.
[51] L. Zhang, R. Ghosh, M. Dekhil, M. Hsu and B. Liu, Combining lexicon-based and learning-based methods for twitter sentiment analysis, HP Laboratories, Technical Report HPL-2011, 2011, pp. 89.