

Evaluating semantic representations for extended association rules

Luis Gutiérrez Espinoza and Brian Keith Norambuena*

Department of Computing and Systems Engineering, Universidad Católica del Norte, Chile

Abstract. In this work, we evaluate the impact of changing the semantic text representation on the performance of the AR-SVS (extended association rules in semantic vector spaces) algorithm on the sentiment polarity classification task on a paper reviews dataset. To do this, we use natural language processing techniques in conjunction with machine learning classifiers. In particular, we report the classification performance using the F_1 and accuracy metrics. The semantic representations that we used in our evaluation were chosen based on a systematic literature review, leading to an evaluation of AR-SVS with FastText, GloVe, and LDA2vec representations, with word2vec providing the baseline performance. The results of the experiments indicate that the choice of semantic text representation does not have major effects on the performance of AR-SVS for polarity classification. Furthermore, the results resemble those obtained in the original AR-SVS article, both in quantitative and qualitative terms. Thus, while direct improvements in classification performance were not found, we discuss other aspects and advantages of using different semantic representations.

Keywords: Association rules, semantic vector spaces, word embeddings, sentiment analysis, sentiment classification, paper reviews

1. Introduction

In recent years, research carried out on natural language processing has given rise to numerous applications [23,37,43,50,53] that have allowed advances in human-computer interaction. For example, current technology allows a person to speak to a mobile device, and for it to perform specific actions, thus understanding instructions at a purely objective levels [4,20]. However, when it comes to the problem of natural language processing, there is a lot of content in uttered speech acts that is beyond the merely objective [32,34,35]. Specifically, communication between people involves delivering information that is not directly encoded in the message in question, which implies a capacity for inference on the part of the receiver (i.e., a machine in the case of computing).

The area that deals with the understanding of the subjective aspects of texts by machines is called sentiment analysis [8,50]. In particular, sentiment analysis aims to determine the sentiment polarity, opinion, or points of view, which are latent in text sources, which may correspond to tweets [5,51], product reviews [12,57], opinions about government entities [60], among others.

In the literature, several algorithms have been proposed to perform sentiment classification based on supervised machine learning, one of these algorithms is AR-SVS, which stands for *extended Association Rules in Semantic Vector Spaces* [29]. The AR-SVS algorithm makes use of the concept of extended

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

association rules, which come from the classic association rules of data mining [2,3,31]; and semantic vector spaces [17,18,29,56], which we will refer to as SVS. SVS represent words in a vector space, in such a way that those that have common semantics are close to each other, forming clusters that can represent concepts in a more general way.

The AR-SVS algorithm proved to be competitive in the sentiment classification task [28,29] with respect to the state of the art. However, the performance of the algorithm was measured using only one text representation: *word2vec*. Thus, its performance using different text representations other than *word2vec* is unknown. Therefore, the evaluation of the algorithm using different text representations contributes to the knowledge of a scientifically validated algorithm in an area of research that has experienced a significant increase in interest in recent years.

If the algorithm is significantly sensitive to its text representation, this would mean that it has a high degree of dependency between its performance and the underlying semantic representation. Thus, future efforts should focus on, for example, making use of the text representation that provides the best performance for the algorithm. On the other hand, if the algorithm does not present significant variations when using different text representations, we may assume that it is a robust algorithm with respect to its underlying semantic representation. Thus, later studies could focus on, for example, finding the representation that makes using this method computationally cheaper. In any case, this work presents an exploratory analysis as well as a replication of the original method and results.

2. Background and related work

2.1. Sentiment analysis

Sentiment analysis, also called opinion mining, is a subfield of natural language processing. Sentiment analysis comprises the tasks of detecting, extracting, and classifying opinions, sentiments, and attitudes concerning different topics, expressed as text [31,50]. Sentiment analysis contributes to the observation of a population's attitudes towards political movements, market intelligence, the level of consumer satisfaction with a product or service, box office prediction for feature films, among others [50]. The availability of opinions and evaluations, in general, has seen an increase due to e-commerce and social networks [16,36]. Currently, e-commerce consumers rely primarily on reviews of products posted by previous consumers to buy, while producers and service providers improve their market propositions by obtaining consumer feedback. For example, analysis of product reviews published on Amazon can influence the decision to purchase these [6,10,19].

There are numerous challenges that sentiment analysis must face [8,50]; for example, the same word can have negative connotations in some contexts, and positive in others [39]; on the other hand, there is a great variety of ways in which people express their opinion, this means that small changes in the messages communicated can cause an important difference in the underlying opinions; this can be seen in the phrases "the movie was good" and "the movie was not good" [22]. Furthermore, the opinions expressed are not purely composed of a particular type of assessment, since they may be composed of sentences that show a positive opinion on the subject, and others a negative one. Altogether, these conditions envision the inherent difficulty in the task of sentiment analysis, since it is challenging even for human beings [33].

More specifically, the polarity classification task [50] seeks to determine the sentiment polarity of a document (e.g., positive or negative sentiment). Different sources of texts have been used by studies to determine the polarity of texts, such as tweets, medical texts, and news articles [7,13,40,41,44]. In essence,

polarity classification deals with establishing a positive or negative sentiment orientation. However, there might be different levels of intensity with which sentiments are expressed in a text. For example, the sentences “I don’t like you” and “I hate you” share the same negative semantics, but the latter should be assigned a higher intensity than the first [9]. Several classification methods have been used to carry out the task, however, support vector machines have presented better performance than other classifiers, including decision trees, Bayesian classifiers, and others [50].

Common approaches to solve the polarity classification problem rely on supervised machine learning algorithms to label documents. In particular, support vector machines and simple Bayesian classifiers are both popular approaches [50]. Other supervised machine learning classification methods used in this task are decision trees, neural networks, and maximum entropy classifiers [47].

2.2. Semantic vector spaces

The main idea behind semantic vector spaces [17,18,30] is to represent each document in a collection as a point in space, which is equivalent to a vector in vector space. Thus, the points that are close to each other in this space share similar semantics, and those that are far away tend to be semantically different. The term “semantics” refers to the meaning of a word, a phrase or sentence, or any text in human language, in addition to the study of such meaning [56]. Once the semantic vector space has been built, a recurring task is to check how semantically similar are specific documents with respect to a given query, also called a “pseudo-document”.

Vectors are common structures in the study of artificial intelligence and cognitive science, considering this fact, the most basic form of semantic vector spaces consisted of making use of the frequencies of word occurrences in a body of text to obtain information about its semantics [56]. Thus, in its most basic form, the first step to determine the semantic vector space is to scan the corpus and count the occurrence of some object (a word or pair of words) in a certain situation (document, context, or pattern), and store the result in the corresponding entry in the occurrence matrix, such as in the TF-IDF representation [14]. However, recent advances have created context-based semantic representations such as word2vec [38] and GloVe [49], among others.

2.3. Association rules

Association rules are an important type of pattern that can be found within data. The goal of association rule mining is to find all the co-occurrence relationships, called associations, between the various objects. The following is the problem of mining association rules initially proposed by [3].

We consider the problem of analyzing a data set made up of transactions. Let $I = \{i_1, i_2, \dots, i_m\}$ be a set of items. Let $T = \{t_1, t_2, \dots, t_n\}$ be a set of transactions, also called a database, where each transaction t_i is a set of items such that $t_i \subseteq I$. An association rule is an implication of the form

$$X \rightarrow Y, \text{ where } X \subset I, Y \subset I, \text{ y } X \cap Y = \emptyset,$$

where X and Y are called *itemsets*. The left-hand side (LHS) of the rule is called **antecedent** and the right-hand side (RHS) is called **consequent**.

We say that transaction $t_i \in T$ covers the itemset X if transaction t_i contains the itemset X (i.e., $X \subseteq t_i$). The *support count* [55] of X in T , denoted by $|X|$ is the number of transactions in T that cover X . Using this definition we can define importance measures for association rules. In particular, we measure the importance of an association rule based on its support and confidence; as these metrics are the most used in the literature.

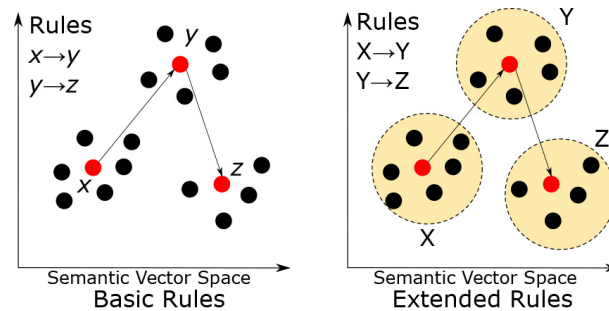


Fig. 1. Association rules in semantic vector spaces (based in Fig. 1 from [29]). The left side shows how basic association rules would be represented in a semantic vector space as simple point (x, y, z) associations ($x \rightarrow y, y \rightarrow z$). The right side shows how extended association rules would encompass a whole region (X, Y, Z) rather than just single points ($X \rightarrow Y, Y \rightarrow Z$).

The support $X \rightarrow Y$ of an association rule is the percentage of transactions in T that contain $X \cup Y$ [48], and can be considered as an estimate of the probability $P(X \cup Y)$. In this way, the support of the association rule determines how applicable the rule is in the transaction set T . Let n be the number of transactions in T . The support of the rule $X \rightarrow Y$ is calculated as $support = \frac{|X \cup Y|}{n}$. Support is a useful metric because if it has a very low value, the rule can only happen by chance and does not represent a pattern itself.

The confidence $X \rightarrow Y$ of an association rule is the percentage of transactions in T that contain X , and that also contain Y [48]; can be considered as an estimator of the conditional probability $P(Y|X)$, and is calculated as $confidence = \frac{|X \cap Y|}{|X|}$. In this way, the confidence of an association rule determines how predictable the rule is. If the confidence of an association rule is very low, the occurrence of Y from X cannot be predicted or inferred with an acceptable determination. An association rule with low confidence has low applicability to the problem domain.

Taking support and confidence into consideration, the task of mining association rules in a set of T transactions is to discover all association rules in T that have support and confidence greater than or equal to a minimum support threshold (denoted by **minsup**) and a minimum confidence threshold (denoted by **minconf**), respectively.

It is worth mentioning that both **minsup** and **minconf** are defined by the user [31]. There are several algorithms to perform association rule mining, in this work we consider the apriori algorithm [2], which allows for the generation of association rules based on the concept of frequent itemsets [31].

2.4. Extended association rules: AR-SVS algorithm

The AR-SVS algorithm (*extended Association Rules – Semantic Vector Space*) [28,29], exploits the semantic similarities of words represented in semantic vector spaces to generate extended association rules that are used by classifiers in sentiment polarity classification tasks. One of the motivations of this algorithm is to represent associations between regions of semantic space, and not just between specific words. Figure 1 presents the intuition behind the AR-SVS algorithm. Note that Fig. 1 uses two dimensions for the semantic vector space as an example, but in practice, semantic vector spaces usually have hundreds of dimensions [17,18,56].

Although an association rule of this type can have multiple words in both its LHS and RHS, in the case of using association rules for classification, the RHS only has the class label, so the construction of the semantically similar regions is carried out considering the terms of the LHS. To determine the similarity of the terms, we use cosine similarity once the vectors have been normalized. This method allows capturing

semantic associations in the text as well as allowing inferences to be made about what each document really means. Intuitively, the neighboring terms of terms that indicate a positive (or negative) polarity in the semantic vector space, have a similar orientation to these.

Algorithm 1: AR-SVS

Input: Set R of association rules, $n \in \mathbb{N}$ terms in the neighborhood.

Output: Set R' of extended association rules.

```

1:  $R' \leftarrow \emptyset$ 
2:  $\forall r \in R$  :
3:    $K_r = \{represent(i) | i \in LHS(r)\}$ 
4:    $S_r = \cup_{k \in K_r} \{closest(k, n)\}$ 
5:    $\forall t \in S_r$  :
6:      $r' = (\{t\} \rightarrow RHS(r))$ 
7:      $support(r') = support(r)$ 
8:      $R' = R' \cup \{r'\}$ 
9: return  $R'$ 

```

The Algorithm 1 shows the AR-SVS pseudocode. The functions $LHS(r)$ and $RHS(r)$ obtain the sets on the left and right sides of the association rule r , respectively. The parameter n corresponds to the number of neighboring terms to the central terms that will be considered in the execution of the algorithm. The function $represent(i)$ obtains the representation in the semantic vector space of the term i . The $closest(k, n)$ function gets the n terms closest to the k term according to the similarity metric. Also, the generated association rules inherit the support from the original rule [29]. Finally, note that the algorithm ends when there are no more rules to extend and there are no additional stopping criteria. Furthermore, each rule has at most a finite number of terms in its LHS and the algorithm extracts the n closest terms for each one of the LHS terms. Thus, the algorithm completes in finite time.

It should be noted that the extended association rules generated by the AR-SVS algorithm are used to determine the semantic orientation of a document through a classifier; For this, it must be provided with a representation of each document, which is generated by a scoring function called RBS (*Rule Based Scoring*), which assigns a score for the class of positive polarity, and another for the negative, in each document.

In summary, once the association rules are generated using the Apriori algorithm, they are extended using the AR-SVS algorithm. Then, using the extended association rules, a useful representation for a classifier is generated for each document, through the RBS function. The pseudocode for this function is presented in Algorithm 2.

Algorithm 2: RBS (*Rule Based Scoring*)

Input: Set R of association rules, list of documents D .

Output: c -dimensional vectorial representation of the documents, where c is the number of classes.

```

1:  $\forall d \in D$  :
2:    $v_d = zeros(c)$ 
3:    $\forall r \in R$  :
4:      $v_d[RHS(r)]_+ = I(LHS(r) \subseteq d) * sup(r)$ 
5: return  $v_d$ 

```

3. Materials and methods

3.1. Problem definition and objectives

Having provided an overview of the related work and preliminary concepts, we now state the problem and objective of this work. The general objective of this work is to evaluate the performance of the AR-SVS algorithm [29] in the task of sentiment polarity classification. In particular, we focus on the domain of scientific paper reviews [1,25–27], as our work is an extension of the original works in which the AR-SVS algorithm was presented [28,29] we also use the same dataset for comparison purposes.

The core idea of the AR-SVS method is that the itemsets of an association rule can be modified by replacing elements with other similar elements from the underlying vector space without changing the validity of the rule, rather than simply applying set operations to extend the itemsets. Our work seeks to provide an in-depth evaluation of the existing AR-SVS model with respect to the SVS component of this algorithm. A similar approach has been explored by Ozaki [45], who also uses vectorial representations to extend association rules. However, being a relatively new algorithm and concept, there are still multiple aspects of extended association rules based on vectorial representations that must be explored, such as how influential the underlying vector representations are to the final performance of the methods. Thus, we sought to evaluate the effects of the choice of semantic representation of text on classification performance.

The work of [29] presented the AR-SVS algorithm, which proposed the extension of association rules for the task of sentiment analysis, and more precisely, sentiment classification. The extension of the association rules is conducted through the construction of regions in a vector space containing word representations as points, using some vector similarity measure for the determination of the neighborhoods, and a natural number that establishes the number of close vectors to be considered as part of the same region. Once the extended association rules are obtained, they are used to generate a score vector for each document, using a separate scoring algorithm. Subsequently, these score vectors and their respective class labels are fed to supervised machine learning classifiers, such as Naïve Bayes [52] or Support Vector Machines [24], to perform sentiment classification over unseen documents.

As the classification procedure relies on the word representations to determine the regions that will generate the extended association rules, the technique to generate these word representations (i.e., word embeddings) is crucial. The works of [29], generated the word embeddings using word2vec [38], and the impact on the performance of the classification task caused by the use of that specific representation was unknown. In this line, the purpose of our work is to determine how the adoption of a specific model to generate word embeddings (i.e. how the word vectors are distributed in the vector space) affects the global performance of the classification task. It is known that word embeddings capture both semantic and syntactic properties of words within a document [11], so the distribution of words in the semantic vector space changes according to the way a model sets the vectors, capturing these patterns in a different fashion for each model.

To carry out this analysis, we selected three different models to generate word embeddings, in addition to the original word2vec, from a state-of-the-art study in this subject [17] and performed the classification task, reporting the classification metrics. The AR-SVS algorithm and the word embeddings used in the experiments are detailed in future sections.

3.2. Paper reviews data set

Various data sets have been used in the task of classifying sentiments, such as *tweets* [46], movie reviews [47], and consumer feedback [15].

Table 1

Highlighted articles from the systematic literature review of [17]. We use these articles to select the semantic vector space representations used in our experiments

Year	Title	Authors
2007	Investigating Unstructured Texts with Latent Semantic Analysis	Fridolin Wild, Christina Stahl.
2013	Efficient estimation of word representations in vector space	Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean.
2014	GloVe: Global Vectors for Word Representation	Jeffrey Pennington, Richard Socher, Christopher D. Manning.
2016	Bag of Tricks for Efficient Text Classification	Armand Joulin, Edouard Grave, Piotr Bojanowski, Tomas Mikolov.
2016	Mixing Dirichlet Topic Models and Word Embeddings to Make lda2vec	Christopher Moody.
2016	Compressing Word Embeddings	Martin Andrews.

In carrying out this work, we perform a sentiment classification task using the data set presented in [25], which consists of peer reviews of scientific articles, mostly in Spanish, together with the latent appreciation they possess, which can range from very negative to very positive. In particular, a data set consisting of reviews of scientific articles was used. The data set consists of a total of 405 reviews, most of them written in Spanish. 17 reviews were excluded for being written in English, and 6 for being empty, leaving a total of 382. It should be noted that this data set is identical to the one used in the original article by AR-SVS [28].

Each review has the attributes *Evaluation* and *Orientation*. *Evaluation* refers to the actual evaluation of a reviewer against a scientific article. *Orientation* represents the subjective perception of each review by the authors of the original work; that is, how negative or positive the review is perceived when it is read by someone else. Both attributes are measured on an integer scale ranging from -2 to 2 , inclusive, where the most negative and positive values represent a negative or positive polarity, respectively. Since *Evaluation* does not always match the semantic orientation of texts [28], the *Orientation* attribute is used to determine the class labels for the dataset. A negative class label is assigned if the review has an *Orientation* of -1 or -2 , and positive if it has the values 1 or 2 . Reviews with an *Orientation* equal to 0 were discarded.

3.3. Semantic vector spaces

Based on the systematic literature review done by [17], Table 1 shows the Word Embedding representations to use in our evaluation. In general, dense vector representations of words have been widely adopted with satisfactory results in general natural language processing tasks and other domains with good results. In particular, word embeddings based on neural models have been reported as superior to those generated by matrix factorization [17]. In particular, multiple embedding methods are variants of a specific group of neural models: word2vec. Studies have been conducted on the impact of including other techniques together with dense vector representations to perform natural language processing tasks, reporting good results in general. Despite the good performance of dense vector representations, they also have some drawbacks, such as the lack of interpretability of the real values that make up the embedded vectors. For the purposes of this particular work, the text representations proposed in [49] – GloVe, in [21] – FastText, and in [42] – LDA2vec have been selected.

These representations were selected due to the advantages that are detailed in their respective articles, such as: for FastText, we have the most efficient training out of the reviewed semantic representations; for GloVe we have the usage and inclusion of global word information a text; for LDA2vec we have the inclusion of additional information beyond the occurrence of words when generating the representations.

3.4. Parametrization

3.4.1. Text representation

The parameterizations of the text representations were made as follows.

- (1) **word2vec**: Being the text representation of the baseline, its parameterization, including the number of hidden neurons and the size of the context window, was established according to the work where the AR-SVS algorithm was originally proposed [28].
- (2) **FastText**: The hyperparameters of the model were configured as proposed in the experiments of their original work [21], since the authors considered their results to be competitive with state-of-the-art methods.
- (3) **GloVe**: The hyperparameters of the model were selected according to the experiments of the original work, including the dimensionality of the vectors and parameters related to the construction of the co-occurrence matrix that this model uses [49].
- (4) **LDA2vec**: The two main elements of LDA2vec are the LDA model used to extract topics from the totality of documents, and the CBOW model to generate the text representations, which in combination with the topic vectors make up the final model. The hyperparameters of the two main elements were configured according to the best results obtained in the original article [42].

3.4.2. Apriori and AR-SVS algorithms

For the generation of base association rules and their extension in the semantic vector space, the following parameters were used:

- (1) **Apriori Algorithm**: The parameterization of the hyperparameters of the Apriori algorithm, such as *support* and *ADSup* [58,59], was also carried out according to the original work of AR-SVS [28], where the values of the parameters for the case of binary classification.
- (2) **AR-SVS Algorithm**: The only hyperparameter of the AR-SVS algorithm is the number of closest terms considered to establish regions in the semantic vector space, the number of terms was established according to the original work, with a value of two [28].

3.5. Evaluation metrics

Generally, sentiment rating performance is evaluated using four metrics: accuracy, precision, recall, and F_1 . These indicators are calculated using a confusion matrix, which is a table that shows the performance of an algorithm, in terms of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Metrics are defined according to the following equations:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN},$$

$$Precision = \frac{TP}{TP + FP},$$

$$Recall = \frac{TP}{TP + FN},$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}.$$

Accuracy is the ratio of all true predicted instances to all predicted instances. An accuracy of 100% indicates that the predicted instances are the same as the actual instances. Precision is the ratio of true

positives to all positively predicted instances. Recall is the ratio of predicted true positives to all true positive instances. The F_1 score is the harmonic mean of precision and recall [54], which allows a better calculation of the performance of the classifier in the face of very different precision and recall values.

3.6. Classifiers and evaluation

Both Naive Bayes and Support Vector Machines, were parameterized with the values determined in the original work of AR-SVS [28], in order to replicate the baseline classification results as closely as possible.

For each of the four text representations (the original word2vec, GloVe [49], LDA2vec [42], and FastText [21]), we performed ten experimental runs, which consisted of dividing the data set using the holdout technique at a ratio of 70:30 for the training set and test set, respectively. Then two classifiers (SVM and NB) were trained, using the training set, to later report the metrics F_1 and *accuracy* when performing the classification on the test set. Each experiment consisted of fifteen iterations as described above, with new training and test sets generated randomly but maintaining the training and test set ratio. As the final result of each execution, the average of each metric considered for the two classifiers is reported, taking into account the fifteen iterations of the experiment. In this way, ten performance averages are obtained for SVM and NB considering F_1 and accuracy.

The final report consists of the average of the averages mentioned above, in addition to their standard deviation, for each possible combination between classifier type, metric, text representation, and set of association rules used in the classification, whether they are the basal rules. (RBS-B), extended (RBS-X), or the union between both (RBS-BX).

We executed ten experiments to take advantage of the variability of the semantic vector spaces generated by representations that would allow their viable reconstruction in each execution; despite this, the analysis of the results suggests that the variability in the learning of the semantic vector space does not significantly affect the performance of the AR-SVS method in the sentiment polarity classification task.

4. Results and discussion

In this section, the final results of the work and their respective discussion are presented.

4.1. Quantitative results

Table 2 shows the performance metrics of the NB classifier under the different configurations of the AR-SVS algorithm. In general, the classification results align with previous research with this data set [26] and with previous evaluations of the AR-SVS approach [28,29]. We also note that due to the complexities of the paper reviews data set that we used, the reported accuracy values are not as high as in other sentiment classification tasks which are usually simpler [26].

Regarding accuracy, both FastText and GloVe report a significant drop in performance using RBS-X, while LDA2vec does not show considerable differences within the same category of rules. The values of accuracy when applying RBS-BX are maintained with differences that are within the standard deviations of the values (i.e., marginal differences), except in the case of FastText, where there is a decrease that escapes 0.01 range of the standard deviation.

As with accuracy, the results with F_1 show a decrease in the metric in the RBS-X case for the FastText and GloVe representations, while LDA2vec remains within the standard deviations with a marginal

Table 2

Table showing the results obtained with the NB classifier. The first three rows show accuracy results for the set of base rules, the set of extended rules, and the union of both sets. The last three rows show the results in terms of the F_1 metric for each set of rules

	Naive Bayes	word2vec	FastText	GloVe	LDA2vec
Accuracy	RBS-B	0.63 ± 0.01	0.63 ± 0.01	0.64 ± 0.01	0.63 ± 0.02
	RBS-X	0.66 ± 0.02	0.57 ± 0.01	0.57 ± 0.02	0.63 ± 0.01
	RBS-BX	0.65 ± 0.01	0.62 ± 0.01	0.65 ± 0.01	0.63 ± 0.01
F_1	RBS-B	0.62 ± 0.02	0.62 ± 0.01	0.64 ± 0.03	0.61 ± 0.02
	RBS-X	0.64 ± 0.03	0.53 ± 0.02	0.53 ± 0.02	0.60 ± 0.02
	RBS-BX	0.64 ± 0.01	0.61 ± 0.01	0.64 ± 0.01	0.62 ± 0.01

Table 3

Table showing the results obtained with the SVM classifier. The first three rows show accuracy results for the set of base rules, the set of extended rules, and the union of both sets. The last three rows show the results in terms of the F_1 metric for each set of rules

	Support vector machine	word2vec	FastText	GloVe	LDA2vec
Accuracy	RBS-B	0.65 ± 0.02	0.65 ± 0.02	0.65 ± 0.02	0.65 ± 0.02
	RBS-X	0.62 ± 0.02	0.61 ± 0.01	0.62 ± 0.01	0.63 ± 0.02
	RBS-BX	0.63 ± 0.01	0.64 ± 0.01	0.65 ± 0.01	0.63 ± 0.01
F_1	RBS-B	0.57 ± 0.03	0.57 ± 0.03	0.59 ± 0.03	0.58 ± 0.04
	RBS-X	0.51 ± 0.03	0.47 ± 0.01	0.47 ± 0.01	0.49 ± 0.02
	RBS-BX	0.52 ± 0.02	0.54 ± 0.02	0.57 ± 0.03	0.54 ± 0.02

difference. Referring to RBS-BX, both GloVe and LDA2vec show marginal differences in performance, while FastText again shows a difference of 0.01 considering the standard deviations.

Table 3 shows the performance metrics obtained with the SVM classifier. Regarding accuracy, a stable classification performance is observed, with only marginal differences in RBS-X and RBS-BX using FastText, GloVe, and LDA2vec.

The reported values of F_1 show marginal differences for the RBS-X and RBS-BX sets, in the same way as accuracy, despite the fact that the average values of the metric for the different text representations have greater variations than the accuracy averages for this same classifier.

We now discuss the main results, considering that the purpose of this work is to evaluate the performance of the classifiers using the text representations other than word2vec and the association rules in the semantic vector space.

In particular, Support Vector Machines maintain an accuracy superior to Naïve Bayes and closer to the original baseline with word2vec. When looking at the case of F_1 , we find that the opposite occurs, as NB presents a superior performance compared to SVM, also being comparable to the performance of the original baseline. The decrease in the values of F_1 by SVM versus NB can be justified through the lack of optimization of SVM hyperparameters, as in these experiments it was parameterized according to the original work [28,29]. It was assumed that the same optimal would be valid for these cases, but this assumption may not necessarily be true.

It should be noted that all the metrics reported for the RBS-B set, both for Naïve Bayes and for Support Vector Machines, are generated by performing classification on a set of association rules produced by the Apriori algorithm, without the use of the AR-SVS algorithm, and therefore without the construction of a semantic vector space. The differences in these values are due to the variability of the association rules produced by the Apriori algorithm due to the different training sets used.

In all the cases of the text representations, the trend in the values of the original work is maintained according to the different association rules [28], where there is a general drop in the metrics compared to the RBS-X set, with an increase when using the RSB-BX set, but not exceeding the RBS-B values.

Table 4

Sample of extended association rules obtained with word2vec. The text in these rules was obtained by doing literal translations of the original text in Spanish

word2vec		
Example	Original rules	Extended rules
Rule WV1	("Development") \Rightarrow ("−1")	("Lack") \Rightarrow ("−1") ("Model") \Rightarrow ("−1")
Rule WV2	("Clear") \Rightarrow ("−1")	("To Evaluate") \Rightarrow ("−1")
Rule WV3	("Study") \Rightarrow ("−1")	("Conclusion") \Rightarrow ("−1") ("To Present") \Rightarrow ("−1")
Rule WV4	("Result") \Rightarrow ("1")	("Article") \Rightarrow ("1")

Table 5

Sample of extended association rules obtained with FastText. The text in these rules was obtained by doing literal translations of the original text in Spanish

FastText		
Example	Original rule	Extended rules
Rule FT1	("Clear") \Rightarrow ("−1")	("Clarity") \Rightarrow ("−1") ("To Clarify") \Rightarrow ("−1")
Rule FT2	("Result") \Rightarrow ("1")	("Resolved") \Rightarrow ("1") ("Understanding") \Rightarrow ("1")
Rule FT3	("Problem") \Rightarrow ("−1")	("Problematic") \Rightarrow ("−1") ("To Deepen") \Rightarrow ("−1")
Rule FT4	("Conclusion") \Rightarrow ("−1")	("Confusion") \Rightarrow ("−1") ("Inclusion") \Rightarrow ("−1")

4.2. Qualitative results

Along with the quantitative results and analyzes, qualitative results and analyzes are also presented. Table 5, Table 6, Table 7, and Table 4, exhibit examples of extended association rules using semantic vector spaces generated by each text representation, in addition to the association rule that originates them. Figure 2 illustrates a directed graph with the relationships found between the terms of the sample association rules for the text representations, for compatibility reasons the accents have been omitted. Each node of the graph corresponds to a term present in the antecedent of some association rule, while the arcs indicate the text representation whose semantic vector space generated the relationship between the terms. The arc labels were set as FT for FastText, GV for GloVe, L2V for LDA2vec, and W2V for word2vec.

Note that the effectiveness of the general AR-SVS model was evaluated in [28,29] with the same data set. Thus, we do not seek to evaluate the effectiveness of the model, but rather focus on the qualitative differences that changing the semantic vector space produces on the extended rules.

Next, an analysis and comparison between the extended association rules and their root, according to text representation, is presented. The terms were rebuilt from their roots. In general, domain knowledge is required to properly analyze and understand the relationships described by the graph of Fig. 2. Furthermore, domain knowledge could be used to interpret the rules obtained by the different text representations with the AR-SVS algorithm.

In the case of the WV1 rule, a negative class label is set to the term "Development", with association rules extended to the terms "Lack" and "Model". This may be due to the semantic characteristics that word2vec can extract, since "Development" and "Lack", due to their semantics and the possible contexts of the documents, could indicate a sustained tendency of reviewers to target development of the articles as

Table 6

Sample of extended association rules obtained with GloVe. The text in these rules was obtained by doing literal translations of the original text in Spanish

GloVe		
Example	Original rule	Extended rules
Rule GV1	("To Use") \Rightarrow ("1")	("Tools") \Rightarrow ("1")
Rule GV2	("Conclusion") \Rightarrow ("-1")	("Irrelevant") \Rightarrow ("-1")
Rule GV3	("Problem") \Rightarrow ("-1")	("Normal") \Rightarrow ("-1")
Rule GV4	("Study") \Rightarrow ("-1")	("Case") \Rightarrow ("-1")

Table 7

Sample of extended association rules obtained with LDA2vec. The text in these rules was obtained by doing literal translations of the original text in Spanish

LDA2vec		
Example	Original rule	Extended rules
Rule LV1	("Clear") \Rightarrow ("1")	("To Use") \Rightarrow ("1")
Rule LV2	("Development") \Rightarrow ("1")	("Work") \Rightarrow ("1")
Rule LV3	("To Apply") \Rightarrow ("1")	("Result") \Rightarrow ("1")
Rule LV4	("Conclusion") \Rightarrow ("-1")	("Example") \Rightarrow ("-1")

deficient, while the term "Model" can also be considered within the development of a work. Comparing WV1 with LV2, the latter determines the term "Development" with a positive class label, extending this rule to the term "Work". The difference in polarity between the WV1 and LV2 rules could be due to the consideration of additional information by the LDA2vec model, which also includes information about the documents at a global level through the LDA component of the model.

The FT1 rule, generated with FastText, sets a negative class label for the term "Clear", with association rules extended to "Clarity" and "To Clarify". This can be interpreted in the same sense as WV1, where the adjective "Clear" and its variants are used to indicate a lack of clarity of the articles to make themselves understood to the readers. The WV2 rule has the same negative label and extends the relationship to the term "To Evaluate", which could be for the same reasons discussed in this paragraph. On the other hand, the LV1 rule has the term "Clear" with a positive polarity, extending to the term "To Use", similar to the case of LV2, this can be attributed to the consideration of the totality of documents to look for relationships between individual terms. The term "To Use" has a positive class label, and is extended to "Tools" with the GV1 rule. The interpretation of this rule is more evident, since a good use of tools is associated with a positive aspect.

The LV3 rule has a positive class label, and relates the term "Apply" by extending it to "Result". This can be explained due to the positive appreciation of the reviewers for articles that may show an important application of the results of the same. The term "Result" is extended to "Understanding" and "Resolved" with a positive class label via the FT2 rule; on the other hand, it is also extended to "Article" by the WV4 rule with a positive polarity label. The polarity agreement between all the rules that include the term "Result" could be due to the fact that the text representations, despite being generated differently, rescue the same semantic values when the uses of certain terms appear in specific contexts, which is supported by the validity hypotheses of a semantic vector space.

The FT3 rule has a negative class label and extends the term "Problem" to the terms "Problematic" (as a noun) and "To Deepen", which could evidence recurring indications by reviewers about a lack of depth in the problems or justification of the articles. The GV3 rule extends the term "Problem" to the

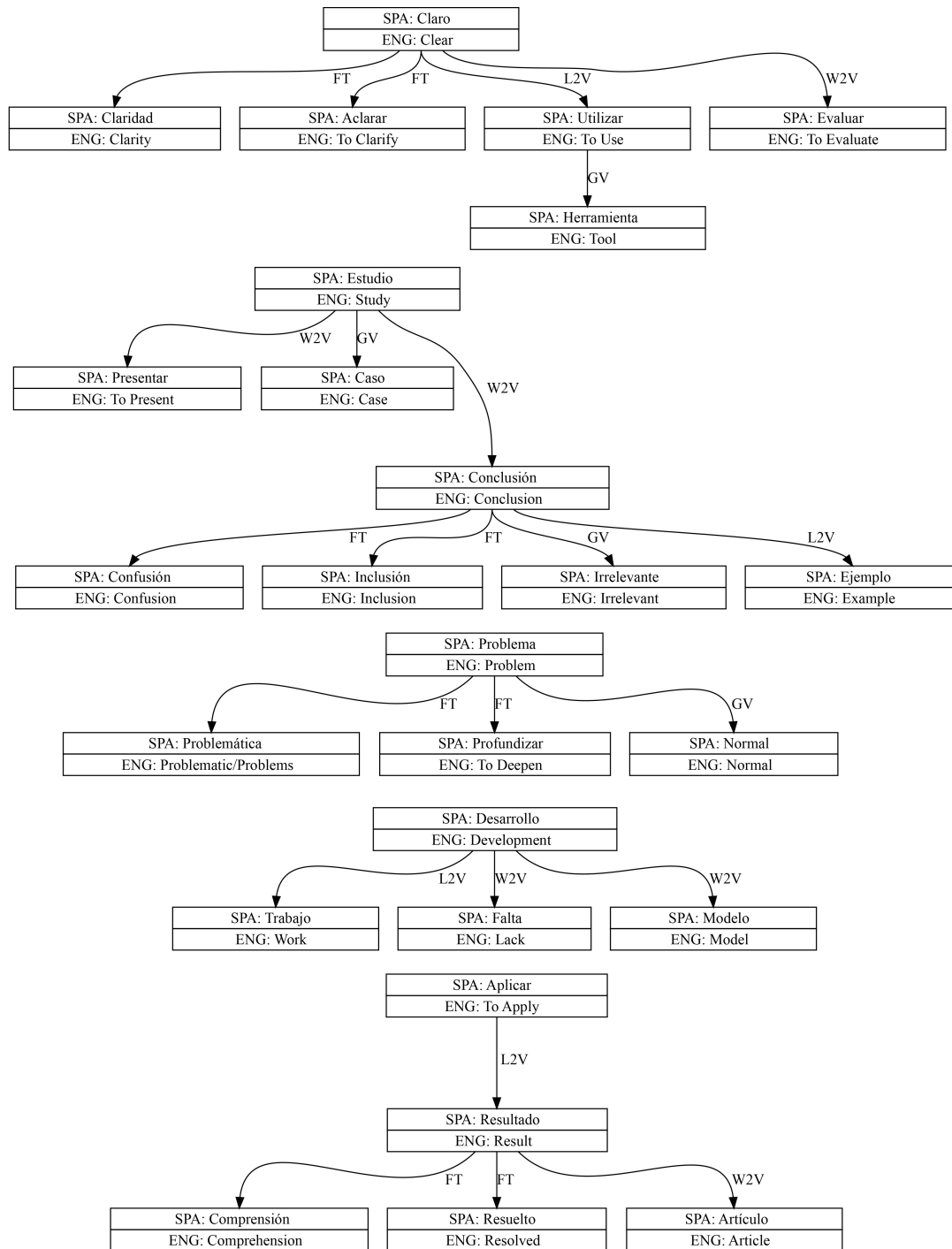


Fig. 2. Graph showing the relationships between words obtained from the association rules with different semantic representations. A directed edge between node X and node Y with label L means that there exists an association rule of the form $X \rightarrow Y$ when using the representation L . In our experiments L can be word2vec (W2V), FastText (FT), GloVe (GV), and LDA2vec (L2V).

term “Normal”, keeping the negative label; this can be explained by the recurrent appearance of problems in scientific articles, which implies that it is normal for problems to arise.

The WV3 rule relates the term “Study” by extending it to the terms “To Present”, and “Conclusion” with a negative class label. Keeping the same polarity, the GV4 rule extends “Study” to “Case”. For both WV3 and GV4, the polarity of the rules can be justified with the term “Study”, with the lack of rigor of the studies conducted in some articles, as well as some incomplete or irrelevant case studies for the central themes of each article, all this according to the judgment of the reviewers. The term “Conclusion” is extended by all the new representations used in conjunction with the AR-SVS algorithm; the FT4 rule extends “Conclusion” to “Confusion” and “Inclusion”, the GV2 rule to “Irrelevant”, and the LV4 rule to “Example”, all with a negative class label. This could be explained by the fact that, generally, the observations concerning the conclusions are made most of the time to point out their defects. It should be mentioned that this is a pattern found for all text representations.

5. Conclusions

The main purpose of this work was to carry out an exploratory analysis on the sentiment polarity classification performance of association rules produced by the AR-SVS algorithm with different semantic vector space, generated by different text representations (word2vec, FastText, LDA2Vec, and GloVe).

In terms of quantitative results, the use of the alternative text representations LDA2vec, FastText, and GloVe does not provide significant improvements in any of the relevant classification metrics compared to the baseline provided by word2vec. In qualitative terms, the semantic vector spaces created by the different text representations are useful to generate extended association rules that share close semantic relationships between the terms that involve them. These extended association rules are even similar between different text representations, which is consistent with the fact that the quantitative results do not show significant differences between them.

Finally, regarding limitations of this work and potential future work, we highlight hyperparameter optimization and the use of a different rule mining algorithm. Further hyperparameter optimization would be possible for the classification algorithms. Moreover, the hyperparameter tuning task could also be extended to the embedding models themselves. Furthermore, the generation of the non-extended association rules was carried out using the Apriori algorithm. The use of a different rule mining algorithm to generate the base association rules and its impact on the subsequent generation of extended association rules is proposed as another potential line of research.

Acknowledgments

This work was partially funded by the National Agency for Research and Development (ANID)/Scholarship Program/DOCTORADO EXTRANJERO BECAS CHILE/2019 – 72200105.

References

- [1] R.S. Acosta, C.M. Villegas and B.K. Norambuena, Heurísticas para data augmentation en nlp: Aplicación a revisiones de artículos científicos, *RISTI-Revista Ibérica de Sistemas e Tecnologías de Informaç ao* **2019**(34) (2019), 44–53.
- [2] R. Agarwal, R. Srikant et al., Fast algorithms for mining association rules, in: *Proc. of the 20th VLDB Conference*, 1994, pp. 487–499.

- [3] R. Agrawal, T. Imieliński and A. Swami, Mining association rules between sets of items in large databases, in: *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, SIGMOD '93, New York, NY, USA, Association for Computing Machinery, 1993, pp. 207–216.
- [4] A. Alateeq, M. Roantree and C. Gurrin, Voxento: A prototype voice-controlled interactive search engine for lifelogs, in: *Proceedings of the Third Annual Workshop on Lifelog Search Challenge*, 2020, pp. 77–81.
- [5] T. Alegre Sepúlveda and B. Keith Norambuena, Twitter sentiment analysis for the estimation of voting intention in the 2017 Chilean elections, *Intelligent Data Analysis* **24**(5) (2020), 1141–1160.
- [6] S. AlZu'bi, A. Alsmadi, S. AlQatawneh, M. Al-Ayyoub, B. Hawashin and Y. Jararweh, A brief analysis of Amazon online reviews, in: *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, IEEE, 2019, pp. 555–560.
- [7] L. Barbosa and J. Feng, Robust sentiment detection on Twitter from biased and noisy data, in: *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, Association for Computational Linguistics, 2010, pp. 36–44.
- [8] M. Birjali, M. Kasri and A. Beni-Hssane, A comprehensive survey on sentiment analysis: Approaches, challenges and trends, *Knowledge-Based Systems*, 2021, 107134.
- [9] S. ChandraKala and C. Sindhu, Opinion mining and sentiment classification a survey, *ICTACT Journal on Soft Computing* **3**(1) (2012), 420–425.
- [10] P.-Y. Chen, S. Dhanasobhon and M.D. Smith, All reviews are not created equal: The disaggregate impact of reviews and reviewers at Amazon.com, *Com (May 2008)*, 2008.
- [11] Y. Chen, B. Perozzi, R. Al-Rfou and S. Skiena, The expressive power of word embeddings, *arXiv preprint arXiv:1301.3226*, 2013.
- [12] E. Costa, R. Ferreira, P. Brito, I.I. Bittencourt, O. Holanda, A. Machado and T. Marinho, A framework for building web mining applications in the world of blogs: A case study in product sentiment analysis, *Expert Systems with Applications* **39**(5) (2012), 4813–4834.
- [13] S. Das, A. Martínez-Jerez and P. Tufano, einformation: A clinical study of investor discussion and sentiment, *Financial Management* **34**(3) (2005), 103–137.
- [14] F. Debole and F. Sebastiani, Supervised term weighting for automated text categorization, in: *Text Mining and Its Applications*, Springer, 2004, pp. 81–97.
- [15] M. Gamon, Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis, in: *Proceedings of the 20th International Conference on Computational Linguistics*, Association for Computational Linguistics, 2004, pp. 841.
- [16] S. Guo, M. Wang and J. Leskovec, The role of social networks in online shopping: information passing, price of trust, and consumer choice, in: *Proceedings of the 12th ACM Conference on Electronic Commerce*, 2011, pp. 157–166.
- [17] L. Gutiérrez and B. Keith, A systematic literature review on word embeddings, in: *International Conference on Software Process Improvement*, Springer, 2018, pp. 132–141.
- [18] I. Habernal and T. Brychcín, Semantic spaces for sentiment analysis, in: *International Conference on Text, Speech and Dialogue*, Springer, 2013, pp. 484–491.
- [19] S. Jang, A. Prasad and B.T. Ratchford, How consumers use product reviews in the purchase decision process, *Marketing Letters* **23**(3) (2012), 825–838.
- [20] H.J. Jeong, S. Ye, J. Lim, I. You, W. Hyun and H.K. Song, A remote computer control system using speech recognition technologies of mobile devices, in: *2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, 2013, pp. 595–600.
- [21] A. Joulin, E. Grave, P. Bojanowski and T. Mikolov, Bag of tricks for efficient text classification, *arXiv preprint arXiv:1607.01759*, 2016.
- [22] N. Kamoen, M.B. Mos and W.F. Dekker, A hotel that is not bad isn't good. The effects of valence framing and expectation in online reviews on text, reviewer and product appreciation, *Journal of Pragmatics* **75** (2015), 28–43.
- [23] Y. Kang, Z. Cai, C.-W. Tan, Q. Huang and H. Liu, Natural language processing (nlp) in management research: A literature review, *Journal of Management Analytics* **7**(2) (2020), 139–172.
- [24] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya and K.R. Murthy, A fast iterative nearest point algorithm for support vector machine classifier design, *IEEE Transactions on Neural Networks* **11**(1) (2000), 124–136.
- [25] B. Keith, E. Fuentes and C. Meneses, A hybrid approach for sentiment analysis applied to paper reviews, in: *Proceedings of ACM SIGKDD Conference, Halifax, Nova Scotia, Canada*, 2017, p. 10.
- [26] B. Keith, E. Fuentes and C. Meneses, Sentiment analysis and opinion mining applied to scientific paper reviews, *Intelligent Data Analysis* **23**(1) (2019), 191–214.
- [27] B. Keith and C. Meneses, Barycentric coordinates for ordinal sentiment classification, in: *Proceedings of the 23rd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2017.
- [28] B. Keith and C. Meneses, Extended association rules in semantic vector spaces for sentiment classification, in: *World Conference on Information Systems and Technologies*, Springer, 2018, pp. 649–658.

- [29] B. Keith and C. Meneses, An extension to association rules using a similarity-based approach in semantic vector spaces, *Intelligent Data Analysis* **23**(3) (2019), 587–607.
- [30] B. Keith Norambuena and C. Meneses Villegas, An extension to association rules using a similarity-based approach in semantic vector spaces, *Intelligent Data Analysis* **23**(3) (2019), 587–607.
- [31] B. Liu, *Web data mining: exploring hyperlinks, contents, and usage data*, Springer Science & Business Media, 2007.
- [32] B. Liu, Opinion mining and sentiment analysis, in: *Web Data Mining*, Springer, 2011, pp. 459–526.
- [33] B. Liu, Sentiment analysis and opinion mining, *Synthesis Lectures on Human Language Technologies* **5**(1) (2012), 1–167.
- [34] Z. Lu, L. Cao, Y. Zhang, C.-C. Chiu and J. Fan, Speech sentiment analysis via pre-trained features from end-to-end asr models, in: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 7149–7153.
- [35] S. Maghilnan and M.R. Kumar, Sentiment analysis on speaker specific speech data, in: *2017 International Conference on Intelligent Computing and Control (I2C2)*, IEEE, 2017, pp. 1–5.
- [36] F.J. Mata and A. Quesada, Web 2.0, social networks and e-commerce as marketing tools, *Journal of Theoretical and Applied Electronic Commerce Research* **9**(1) (2014), 56–69.
- [37] S.M. Mathews, Explainable artificial intelligence applications in nlp, biomedical, and malware classification: A literature review, in: K. Arai, R. Bhatia and S. Kapoor, eds, *Intelligent Computing*, Cham, Springer International Publishing, 2019, pp. 1269–1292.
- [38] T. Mikolov, K. Chen, G. Corrado and J. Dean, Efficient estimation of word representations in vector space, *arXiv preprint arXiv:1301.3781*, 2013.
- [39] S.M. Mohammad, Sentiment analysis: Detecting valence, emotions, and other affectual states from text, in: *Emotion Measurement*, Elsevier, 2016, pp. 201–237.
- [40] A. Montejo-Ráez, E. Martínez-Cámara, M.T. Martín-Valdivia and L.A. Urena-Lopez, Random walk weighting over sentiwordnet for sentiment polarity detection on twitter, in: *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, Association for Computational Linguistics, 2012, pp. 3–10.
- [41] A. Montejo-Ráez, E. Martínez-Cámara, M.T. Martín-Valdivia and L.A. Urena-López, Ranked wordnet graph for sentiment polarity classification in twitter, *Computer Speech & Language* **28**(1) (2014), 93–107.
- [42] C.E. Moody, Mixing dirichlet topic models and word embeddings to make lda2vec, *arXiv preprint arXiv:1605.02019*, 2016.
- [43] F. Nazir, W.H. Butt, M.W. Anwar and M.A. Khan Khattak, The applications of natural language processing (nlp) for software requirement engineering – a systematic literature review, in: K. Kim and N. Joukov, eds, *Information Science and Applications 2017*, Singapore, Springer Singapore, 2017, pp. 485–493.
- [44] Y. Niu, X. Zhu, J. Li and G. Hirst, Analysis of polarity information in medical text, in: *AMIA Annual Symposium Proceedings*, American Medical Informatics Association, Vol. 2005, 2005, p. 570.
- [45] T. Ozaki, Evaluation measures for extended association rules based on distributed representations, in: *Workshops of the International Conference on Advanced Information Networking and Applications*, Springer, 2019, pp. 305–313.
- [46] A. Pak and P. Paroubek, Twitter as a corpus for sentiment analysis and opinion mining, in: *LREc*, Vol. 10, 2010, pp. 1320–1326.
- [47] 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*, Association for Computational Linguistics, 2002, pp. 79–86.
- [48] J.S. Park, M.-S. Chen and P.S. Yu, Efficient parallel data mining for association rules, in: *Proceedings of the Fourth International Conference on Information and Knowledge Management*, 1995, pp. 31–36.
- [49] J. Pennington, R. Socher and C. Manning, Glove: Global vectors for word representation, in: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.
- [50] K. Ravi and V. Ravi, A survey on opinion mining and sentiment analysis: tasks, approaches and applications, *Knowledge-Based Systems* **89** (2015), 14–46.
- [51] P. Ray and A. Chakrabarti, Twitter sentiment analysis for product review using lexicon method, in: *2017 International Conference on Data Management, Analytics and Innovation (ICDMAI)*, IEEE, 2017, pp. 211–216.
- [52] I. Rish et al., An empirical study of the naive bayes classifier, in: *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*, number 22 in 3, 2001, pp. 41–46.
- [53] M.J.F. Rodrigues and A.J. da Silva Teixeira, *Advanced Applications of Natural Language Processing for Performing Information Extraction*, Springer, 2015.
- [54] M. Sadegh, R. Ibrahim and Z.A. Othman, Opinion mining and sentiment analysis: A survey, *International Journal of Computers & Technology* **2**(3) (2012), 171–178.
- [55] R. Srikant and R. Agrawal, Mining sequential patterns: Generalizations and performance improvements, in: *International Conference on Extending Database Technology*, Springer, 1996, pp. 1–17.
- [56] P.D. Turney and P. Pantel, From frequency to meaning: Vector space models of semantics, *Journal of artificial intelligence research* **37** (2010), 141–188.

- [57] V. Vyas and V. Uma, Approaches to sentiment analysis on product reviews, in: *Sentiment Analysis and Knowledge Discovery in Contemporary Business*, IGI Global, 2019, pp. 15–30.
- [58] M. Yuan, Y. Ouyang and Z. Xiong, A text categorization method using extended vector space model by frequent term sets, *Journal of Information Science and Engineering* **29**(1) (2013), 99–114.
- [59] M. Yuan, Y. Ouyang, Z. Xiong and H. Sheng, Sentiment classification of web review using association rules, in: A.A. Ozok and P. Zaphiris, eds, *Online Communities and Social Computing*, Berlin, Heidelberg, Springer Berlin Heidelberg, 2013, pp. 442–450.
- [60] S.M. Zavattaro, P.E. French and S.D. Mohanty, A sentiment analysis of us local government tweets: The connection between tone and citizen involvement, *Government Information Quarterly* **32**(3) (2015), 333–341.
- [61] V. Srividhya and R. Anitha, Evaluating preprocessing techniques in text categorization, *International Journal of Computer Science and Application* **47**(11) (2010), 49–51.
- [62] G. Li and F. Liu, A clustering-based approach on sentiment analysis, in: *2010 IEEE International Conference on Intelligent Systems and Knowledge Engineering*, IEEE, 2010, pp. 331–337.
- [63] W. Medhat, A. Hassan and H. Korashy, Sentiment analysis algorithms and applications: A survey, *Ain Shams Engineering Journal* **5**(4) (2014), 1093–1113.