Extended association rules in semantic vector spaces for sentiment classification

Brian Keith Norambuena[1] and Claudio Meneses Villegas[1]

Departamento de Ingeniería de Sistemas y Computación
Universidad Católica del Norte
Av. Angamos 0610, Antofagasta, Chile
{brian.keith, cmeneses}@ucn.cl

Abstract. Sentiment analysis is a field that has experienced considerable growth over the last decade. This area of research attempts to determine the opinions of people on something or someone. This article introduces a novel technique for association rule extraction in text called Extended Association Rules in Semantic Vector Spaces (AR-SVS). This new method is based on the construction of association rules, which are extended through a similarity criteria for terms represented in a semantic vector space. The method was evaluated on a sentiment analysis data set consisting of scientific paper reviews. A quantitative and qualitative analysis is done with respect to the classification performance and the generated rules. The results show that the method is competitive with respect to the baseline provided by NB and SVM.

1 Introduction

The main objective of the present work is to propose a new method for generating association rules with applications in sentiment analysis. This proposal is based on the intuitive idea that two related terms will be close to each other in the vector representation. Given this, if an association rule contains one of the terms, it is possible that the other can also be used in this association rule. The difficulty of this lies in properly defining the proximity criterion. This general idea can be used to build extended association rules including the closest terms. In particular, it is sought to use this idea of extension of the association rules by proximity to classify the polarity of documents.

There are approaches that propose the use of association rules to carry out the task of classification [3]. Associative classification differs from classic association rules in the sense that a restriction is added to the rules in such a way that in the consequent there can only be one attribute (the class).

The rules of associative classification can be obtained using an algorithm similar to Apriori called CBA-RG to generate the rules and another algorithm CBA-CB to construct the classifier. The rules constructed have the class label in the consequent. From the set of generated rules a subset is selected using a heuristic criterion [3].

There are multiple criteria to generate the rules, the most common are support and confidence. Support is the number of instances in the training set which are relevant to the rule. Confidence refers to the conditional probability that the right-hand side of the rule is satisfied if the left-hand side of the rule is satisfied [5].Another useful metric is the Average Deviation Support which measures the discrepancy in the support distribution and allows determining the rules that discriminate the different classes [12].

In general, depending on how the class label is chosen, there are two kinds of approaches of associative classification: those that make predictions through a strategy of maximum likelihood and those that use multiple rules to generate scores. The methods of associative classification to classify texts have been well studied, but the use of association rules for sentiment classification has not been thoroughly explored yet [4].

The rest of this work is organized as follows: the second section formally describes the proposed method and discusses its basic fundamentals. The third section details the materials and methods utilized to evaluate the proposal, including a description of the data and the tools required. The fourth section shows the main results and their discussion. Finally, in the last section the conclusions and possible lines of future work are presented.

2 Proposed method

2.1 Description of the AR-SVS method

This proposal seeks to exploit the capacity of the association rules for detecting interesting patterns. It is sought to generalize classic association rules in such a way that they do not represent associations between words, but between regions of the semantic vector space (as can be observed in Figure 1). In particular, it is expected to obtain associative classification rules using the words that are located close in the semantic vector space.

In order to generate these new association rules, the closest terms to each term of the LHS (left-hand side) and the RHS (right-hand side) of the association rule would be selected. Note that in general there can be several terms in the LHS and the RHS, so that there can be many vicinities to consider in the construction. In the case of the association rules for classification, whose RHS corresponds to the class label, only the closest terms to the elements of the LHS would be considered.

To determine the similarity of the terms, different methods can be utilized. It is recommended to normalize the vectors, because for the specific task of determining whether two words are similar, this has shown to provide better results [11]. The number of closest terms that will be used for the method would be a parameter defined by the final user.

This method allows capturing the semantic associations in the text, and in particular it allows making inferences on what each document really means, since by extending the rules with the closest terms, the method will have more

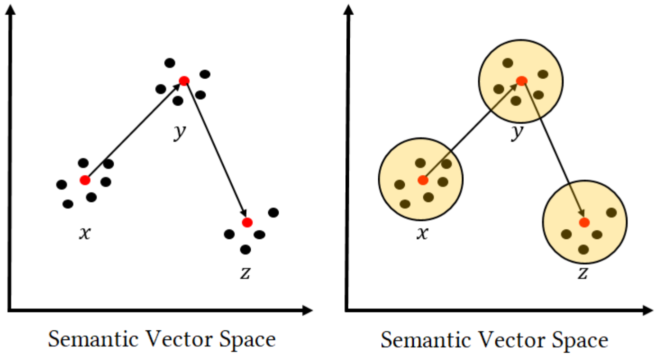

Fig. 1. Association rules in semantic vector spaces.

information at its disposal. This method is considered interesting due to its possible generalizations and its intuitive nature. It is more natural to think in the existence of associated regions inside a semantic vector space than in point associations.

The method has been named AR-SVS (extended Association Rules in Semantic Vector Spaces). Note that the method is composed of several independent components, and the choice of these components is a challenge in itself. Having described the general idea of the method, each one of the steps and the design decisions involved are detailed. The method is described in a general way in Algorithm 1 under the assumptions that the model of vector representation is already trained and the association rules are in the correct format (i.e. classification rules).

Algorithm 1 Algorithm AR-SVS

Input: Set R of association rules, parameter $n \in \mathbb{N}$ that indicates the number of semantically similar terms to utilize

Output: Set R' of extended association rules.

```
function AR-SVS
    Sea $R' = \emptyset$
    $\forall r \in R$:
        $K_r = \{represent(i) \,|\, i \in LHS(r)\}$
        $S_r = \bigcup_{k \in K_r} \{closest(k, n)\}$
        $\forall t \in S_r$:
            $r' = (\{t\} \rightarrow RHS(r))$
            $sup(r') = sup(r)$
            $R' = R' \cup \{r'\}$
    return $R'$
end function
```

In this definition, the functions $LHS(r)$ and $RHS(r)$ obtain the sets of the left-hand side and the right-hand side of the rule r, respectively. The function $represent(i)$ takes the term i and obtains its representation in the vector space. The function $closest(k, n)$ obtains the n terms closest to the term k according to some similarity metric. Finally, $sup(r)$ corresponds to the support of the rule, the generated rules inherit the support of the original rule (if a same rule is generated many times, it inherits the highest corresponding support).

The definition of the method has been done in a general way, allowing freedom to apply the methods considered adequate in each step. The first design decision to consider is the selection of the algorithm to construct the association rules and the selection of evaluation metrics for the rules. Also, it is necessary to select the representation of the terms, the similarity criterion and define the value of n.

One of the limitations of the proposed algorithm is that it only considers the construction of association rules for the classification problem. On the other hand, the proposal described only generates rules of unitary length. This has been done to reduce the method's complexity, because finding all the possible combinations of valid extended association rules would require the definition of a specialized evaluation metric.

2.2 Classification with AR-SVS

Although the algorithm to construct extended association rules by means of semantic vector spaces is intrinsically valuable, it is necessary to remember that the aim of this work requires using the association rules obtained to determine the semantic orientation of a document. In order to do this, it is necessary to have a classification algorithm and a scheme that allows utilizing the association rules to classify.

In particular, an approach based on scores is used to carry out the classification, this is formally described in Algorithm 2. The basic idea of this approach is to construct a scores vector that will represent each document with respect to each class. The vectors built for each document will be utilized as input for some method of traditional classification.

Where the function $zeros(c)$ takes as input the number of classes c and returns a vector v_d initialized in zeros that will be used to store the score associated to each class. The function I is an indicator function that takes the value 1 if the statement in it is true and 0 otherwise. The function $sup(r)$ obtains the support of the rule r. The main part of the algorithm corresponds to the following: for each extended association rules, the support of the rule is added to the vector v_d in the position corresponding to the rules' class. Three variants of the RBS algorithm are defined:

1. **RBS-B**: the input corresponds to the set of basic rules R.
2. **RBS-X**: the input corresponds to the set of extended rules R'.
3. **RBS-BX**: the input corresponds to the union of both sets $R \cup R'$.

Algorithm 2 Rule Based Scoring Algorithm (RBS)

Input: Set of association rules R and the list of documents D.

Output: c-dimensional vector representation of the documents. Where c is the number of classes.

```
function RBS
    $\forall d \in D$:
        $v_d = zeros(c)$
        $\forall r \in R$:
            $v_d[RHS(r)] += I(LHS(r) \subseteq d) \cdot sup(r)$
    return $v_d$
end function
```

To determine the class different approaches can be applied. The simplest way is to assign the class that has the highest score in each document, and in case of a draw, assume neutrality. Another option is training a machine learning classifier that takes as input the scores vectors.

3 Methodology

The method has been evaluated on the dataset of reviews of scientific articles. The dataset has a total of 405 reviews, from these elements the reviews written in English (17 instances) and the empty reviews (6 instances) are discarded, leaving a total of 382 reviews in Spanish. In this work the scale "orientation" has been used, because the evaluation does not always coincide with the semantic orientation of the text [2].

The evaluation of the methods utilizes a holdout approach with a proportion of 70% for the training set and 30% for the tests set, carrying out 10 replications for each method. The averages of accuracy, precision, recall and F1 with its standard deviation are reported for each class. Regarding the preprocessing, first a tokenization of the input is carried out. Then, a stopwords filter [1] is applied. Afterwards, stemming is applied by means of the Porter algorithm [8].

For the rules, a modified variant of the algorithm Apriori is used that considers the minimum support with respect to each class (instead of the total of the dataset) and the average deviation support ($ADSup$). The representation of the text is done using *word2vec* trained on the data set. For the construction of the set S_r the similarity of the cosine between the normalized vectors is considered. The threshold value n is empirically obtained by evaluating qualitatively the similarity of the obtained terms.

The vectors of documents constructed for each variant of RBS are classified using three different approaches: choosing the class with the maximum score, training a Naïve Bayes classifier, and training a support vector machine. Naïve Bayes and SVM with LSA vectors as input are used as a comparison baseline. These two latter methods have been selected due to their wide use in the literature of sentiment analysis [9].The implementation was carried out in Python using the library *sklearn* [7].

For the baseline of NB and SVM, once the preprocessing is completed, a representation is obtained using TF-IDF. The final representation is obtained by applying LSA (utilizing the $n = 100$ more significant components). For the method AR-SVS the *word2vec* representation is used [6], implemented through the library *gensim* [10]. The representation has been trained on the data set, pre-trained vectors have not been utilized.

Finally, for the binary classification, the thresholds that have been used for the Apriori algorithm is a support of 25% with respect to class and an ADSup of 15%. On the other hand, for ternary classification a minimum support of 10% has been used and an ADSup of 40%. These values have been found empirically, evaluating the average accuracy of 10 replicates for different values from 5% to 70% with increments of 5% for both parameters.

4 Results and discussion

4.1 Classification

The results for binary classification are shown in Table 1. The best results are obtained with the RBS-B method, followed by the SVM base. This is the only instance of the method that exceeds the baseline. Although the other variants fail to overcome the performance of NB or SVM in all of the metrics, these present a competitive behavior in accuracy and recall. A larger difference is observed in the results of precision and F_1.

Table 1. Summary of results obtained for binary classification.

		Binary Classification			
Classifier	Representation	*Accuracy*	*Precision*	*Recall*	F_1
NB	TF-IDF-LSA	0.68 ± 0.05	0.67 ± 0.06	0.68 ± 0.05	0.64 ± 0.06
	RBS-B	0.63 ± 0.03	0.62 ± 0.04	0.63 ± 0.03	0.61 ± 0.04
	RBS-X	0.64 ± 0.04	0.64 ± 0.03	0.64 ± 0.04	0.63 ± 0.04
	RBS-BX	0.63 ± 0.03	0.63 ± 0.03	0.63 ± 0.02	0.63 ± 0.03
SVM	TF-IDF-LSA	0.7 ± 0.05	0.7 ± 0.05	0.7 ± 0.05	**0.69 ± 0.06**
	RBS-B	**0.72 ± 0.06**	**0.72 ± 0.07**	**0.72 ± 0.06**	**0.69 ± 0.06**
	RBS-X	0.62 ± 0.05	0.46 ± 0.15	0.62 ± 0.05	0.52 ± 0.10
	RBS-BX	0.67 ± 0.06	0.65 ± 0.14	0.66 ± 0.06	0.65 ± 0.11
MAX	RBS-B	0.65 ± 0.05	0.59 ± 0.16	0.65 ± 0.05	0.52 ± 0.07
	RBS-X	0.65 ± 0.06	0.64 ± 0.11	0.65 ± 0.06	0.54 ± 0.08
	RBS-BX	0.64 ± 0.05	0.53 ± 0.17	0.64 ± 0.05	0.51 ± 0.07

It can also be observed that the use of the extended rules (the X and BX variants) does not produce improvements in the classification results. However, it must be highlighted that even using only the new rules it is possible to classify the documents in a competitive way. The use of both rules (basic and extended) does not produce a consistent effect in the different evaluation metrics and the differences are not significant anyway.

The results in binary classification show that the representation generated by the RBS algorithm can be used to classify the documents in an adequate way. However, it is necessary to observe that its good performance depends on the set of rules used as an input, because the three variants B, X and BX have shown different behaviors on this data set.

The results for ternary classification are shown in Table 2. The behavior of the RBS method is in general similar to the binary case. Again, the good performance of the RBS-B variant in all the metrics can be noted. The results of the RBS variants show a more competitive behavior. It must be noted that the RBS-BX method obtains similar results to the SVM base. Although RBS-B outperforms this method, in this case, adding the extended rules led to a slightly decreased performance.

Table 2. Summary of results obtained for ternary classification.

		Ternary Classification			
Classifier	Representation	*Accuracy*	*Precision*	*Recall*	F_1
NB	TF-IDF-LSA	0.46 ± 0.03	0.42 ± 0.05	0.46 ± 0.03	0.41 ± 0.05
	RBS-B	0.41 ± 0.05	0.42 ± 0.07	0.41 ± 0.05	0.37 ± 0.04
	RBS-X	0.47 ± 0.04	0.38 ± 0.05	0.47 ± 0.04	0.41 ± 0.05
	RBS-BX	0.42 ± 0.05	0.36 ± 0.05	0.42 ± 0.05	0.37 ± 0.04
SVM	TF-IDF-LSA	0.48 ± 0.05	0.46 ± 0.06	0.48 ± 0.06	0.46 ± 0.06
	RBS-B	**0.49 ± 0.05**	0.48 ± 0.05	0.49 ± 0.05	**0.47 ± 0.06**
	RBS-X	0.45 ± 0.06	0.29 ± 0.08	0.45 ± 0.06	0.35 ± 0.07
	RBS-BX	0.48 ± 0.05	0.47 ± 0.04	0.48 ± 0.05	0.46 ± 0.05
MAX	RBS-B	**0.49 ± 0.05**	**0.52 ± 0.1**	**0.5 ± 0.05**	0.41 ± 0.07
	RBS-X	0.45 ± 0.04	0.44 ± 0.12	0.45 ± 0.04	0.36 ± 0.06
	RBS-BX	0.47 ± 0.05	0.45 ± 0.13	0.48 ± 0.05	0.35 ± 0.07

The results in ternary classification corroborate the observed in the binary case with regards to the usefulness of the RBS representation. As before, it can be observed that the use of basic rules allows obtaining a better classification performance. Although unlike the binary case, the variants RBS-X and RBS-BX present a more competitive behavior.

4.2 Generated rules

Having analyzed the main results in terms of classification, the rules generated by the AR-SVS method are now discussed. Table 3 shows some of the original rules and the extended rules generated by the method for the binary classification for exemplification purposes.

Table 3 shows four association rules obtained through Apriori and the extended rules constructed using AR-SVS. It must be noted that in many cases the terms are repeated (e.g., the word “uso” (use) appears both in the positive and negative cases), furthermore, it is possible that the extended rules contain the same terms as the original rules (e.g. in the last rule, the term “deber” (must) was determined similar to “uso” (use) and “trabajo” (work). That is, there are terms that are related both by semantic similarity and co-ocurrences evaluated by the algorithm Apriori, even allowing for the generation of cyclical relationships.

Table 3. Exaxmples of the obtained rules.

Parametrization		
A minimum support of 20% is used and a minimum ADSup of 5% for this example. The two most similar terms are obtained ($n = 2$).		
Examples	**Original rule**	**Extended rules**
Regla 1	(‘Interés’) ⟹ (’1’)	(‘Aplicación’) ⟹ (’1’) (‘Artículo’) ⟹ (’1’)
Regla 2	(‘Faltar’) ⟹ (’-1’)	(‘Uso’,) ⟹ (’-1’) (‘Aspecto’,) ⟹ (’-1’)
Regla 3	(‘Ser’, ‘Mejor’) ⟹ (’1’)	(‘Uso’) ⟹ (’1’) (‘Hacer’) ⟹ (’1’) (‘Aspecto’) ⟹ (’1’) (‘Embargo’) ⟹ (’1’)
Regla 4	(‘Deber’, ‘Trabajo’, ‘Ser’) ⟹ (’-1’)	(‘Uso’) ⟹ (’-1’) (‘Trabajo’) ⟹ (’-1’) (‘Uso’) ⟹ (’-1’) (‘Paper’) ⟹ (’-1’) (‘Uso’) ⟹ (’-1’) (‘Trabajo’) ⟹ (’-1’)

The graph of Figure 2 shows the relationships among the different terms of Table 3. Bidirectional edges represent a co-ocurrence relationship and are labeled with “CO”. Unidirectional edges represent a relationship of semantic similarity according to *word2vec* and are labeled with “w2v”. It must be noted that the nodes can represent many words with different grammatical functions. This semantic multiplicity is due to the fact that during analysis words have been reduced to their root. Given this, for illustrative purposes just one representative has been chosen.

The generated rules show that semantic relationships are not always directly interpretable, meaningful or useful. It is possible that with a larger data set the

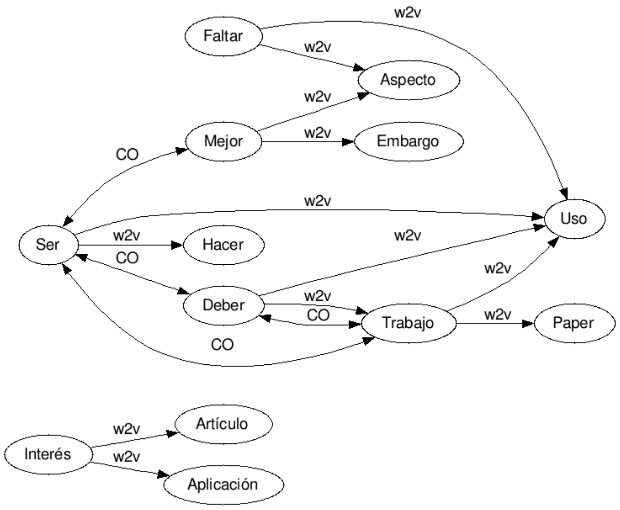

Fig. 2. Graph of relationships for the terms of Table 3.

relationships found with *word2vec* would be more significant. As an example of this, the verb "ser" (to be) is related with "uso" (use) and "hacer" (to do), it is difficult to find a meaningful relationship between words as common as the conjugations of the verb "ser" and two other words of the Spanish language (except perhaps with the verb "estar"(to be)).

The quality of the semantic similarity will depend on the training set of the model *word2vec*. In this particular case it is possible that the data set is too small to obtain deep and relevant semantic relationships. It is possible that in a larger data set the results of the RBS-X and RBS-BX methods would be better. Hence, it would be possible to discover more useful relationships between the different terms present in the text.

Finally, it should be noted that without considering the aspect of classification, the proposal allows finding new association rules. Then, the AR-SVS method developed can be used independently from the RBS algorithm. Taking this into account, it is possible to apply this proposal on other data sets in text form for exploratory purposes.

5 Conclusions

In this work, the concept of extended association rules has been developed, specifically focusing on the case of association rules for classification in sentiment analysis. The results show that the proposed method is competitive, but there are still opportunities of improvement. Association rules have not been exhaustively exploited in the field of sentiment analysis, so this work presents a contribution in terms of a new way of applying them.

Regarding the limitations of the method, having just one term in the LHS of the rule is one of the serious limitations of the method in this proposal. However, as a concept test for the extension of association rules, this proposal meets its objective. The construction of extended association rules with more terms in the LHS would require the development of new ways of evaluating the rules. The classic metrics of support and confidence would not be sufficient, because these are based on the concept of co-ocurrences, while the terms found by the AR-SVS method will not necessarily be able to be evaluated by a co-ocurrence criterion, since their relationship can be deeper or more indirect than this.

Finally, the use of vector representations of the terms to extend the rules can be seen as a method to find association rules in itself for data in text form. Future work will have to consider the development of metrics and algorithms that enable the construction of extended rules that allow more than one term both in the consequent and in the antecedent.

References

1. Baeza-Yates, R., Moffat, A., Navarro, G.: Searching large text collections. In: Handbook of massive data sets, pp. 195–243. Springer (2002)
2. Keith, B., Fuentes, E., Meneses, C.: A hybrid approach for sentiment analysis applied to paper reviews (2017)
3. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining. pp. 80–86. AAAI Press (1998)
4. Man, Y., Yuanxin, O., Hao, S.: Investigating association rules for sentiment classification of web reviews. Journal of Intelligent & Fuzzy Systems 27(4), 2055–2065 (2014)
5. Medhat, W., Hassan, A., Korashy, H.: Sentiment analysis algorithms and applications: A survey. Ain Shams Engineering Journal 5(4), 1093 – 1113 (2014), http://www.sciencedirect.com/science/article/pii/S2090447914000550
6. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
7. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. Journal of Machine Learning Research 12(Oct), 2825–2830 (2011)
8. Porter, M.F.: An algorithm for suffix stripping. Program 14(3), 130–137 (1980)
9. Ravi, K., Ravi, V.: A survey on opinion mining and sentiment analysis: Tasks, approaches and applications. Knowledge-Based Systems 89, 14–46 (2015)
10. Rehurek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. Citeseer (2010)
11. Wilson, B.J., Schakel, A.M.: Controlled experiments for word embeddings. arXiv preprint arXiv:1510.02675 (2015)
12. Yuan, M., Ouyang, Y.X., Xiong, Z.: A text categorization method using extended vector space model by frequent term sets. Journal of Information Science and Engineering 29(1), 99–114 (2013)