Support Vector Machines for Semantic Relation Extraction in Spanish Language

Jefferson Peña Torres(✉), Raúl Gutierrez de Piñerez Reyes, and Víctor A. Bucheli

Universidad del Valle, Cali, Colombia
{jefferson.amado.pena,raul.gutierrez,
victor.bucheli}@correounivalle.edu.co

Abstract. Relation Extraction (RE) is one of the most important topics in NLP (Natural Language Processing). Many tasks such as semantic relation extraction, sentiment analysis, opinion mining, question answering systems and text summarization are supported by RE. The aim of this paper is to present a semantic relations classifier in which are incorporate lexical features, named entity features and syntactic structures. Relations between two entities are classified based on the Datasets for Generic Relation Extraction (reACE). We translate the reACE corpus to the Spanish language for all relation types and subtypes. The results shows a F-score of 75.25%, it is a significant improvement of 11.5% over the baseline model. Finally, we discuss the results according to the model and the useful information to support the forecasting process.

Keywords: Relation extraction · Support vector machines
Named entity recognition · Sequence labeling

1 Introduction

Relation extraction (RE) seek and classify semantic relations among two entities. RE is crucial to text mining, question answering systems, text summarization, among others applications. Moreover, there exists a big amount of information in the form of unstructured text data, such as blogs, news, emails, journal articles and conference papers. This paper contributes to ongoing efforts to develop mechanisms to automate knowledge extraction from text using traditional learning methods for relation extraction. RE tasks specifically refer to the classification of an entity pair to a set of known relations, using documents containing mentions of the entity pairs [11]. In this context, an important task must be the Named Entity Recognition (NER) which is considered the previous step to relation extraction. In this work, first named entities are recognized and then relation between two entities are identified and classified. Thus, the relationships are ordered by entity pairs.

In reACE the relations semantically are annotated using named entities such as persons, organizations, locations, facilities and geo-political entities (GPE:

J. E. Serrano C. and J. C. Martínez-Santos (Eds.): CCC 2018, CCIS 885, pp. 326–337, 2018.
https://doi.org/10.1007/978-3-319-98998-3_26

geographically defined regions that indicate a political boundary) [6]. We adapt the reACE corpus guidelines to the Spanish language by translating all relation types. We use the official reACE corpus from Linguistic Data Consortium (LDC). We train our models with 7269 sentences and 23 reACE relation subtypes on the 7 ACE relations types. Although the NER task is not our goal in these paper this issues remain a problem that affect the semantic relation extraction. If we want improve the detection of entities a natural step is establishing semantic relations between these entities.

This paper focuses in RE on the reACE corpus in which explicit relations are detected and classified. For this, we employ lexical, syntactic and semantic features by using Support Vector Machines (SVMs). In this paper, we show that the syntactic information added to the local and contextual features improving the F-score. We define a set of structural feature vectors that have syntactic information from AnCora and reACE corpus. We also demonstrate how the use syntactic features improves the accuracy of the classifier.

In short, our contributions they are related to the built an annotated Spanish corpus for semantic relations based on the reACE corpus for English and we also use syntactic features from translated corpus for improving a baseline model that is supported in local and contextual features.

2 Related Work

Relation extraction is one of the most important topics in NLP. Many approaches have been explored for relation extraction, including unsupervised and supervised classification. Various works have been performed and various researchers have spent a lot of time and resources developing classifiers to identify the relations using different learning methods.

In Miller [13] is referenced an integrated parsing model with syntax and semantic information which is done using augmented parse trees. In this work, they use PTB corpus in which their trees are augmented to convey semantic information between entities and relations. Zalenko et al. [19] proposes shallow parsing as a prerequisite for relation extraction and use kernel methods to extracting relations from unstructured natural language resources. In this address, a relation extraction problem is formulated as a shallow parse classification problem. Zelenko also proposes that patterns are learned from a set of already extracted relations rather than annotated hand-written relations. In the same vein, Culotta et al. [3] have been working by estimate kernel functions between augmented dependency trees achieving 63.2% F-measure in relation detection and 45.8% F-measure in relation detection and classification on the ACE corpus. In Kambhatla et al. [10], are combined lexical, syntactic, and semantic features with maximum entropy models for extract relations. This approach can easily scale to include more features from a large amount of sources; WordNet, gazatteers, output of other semantic taggers. However, they only model explicit relations achieving 52.8% F-measure on the ACE corpus. Zhang [21] define a extraction framework with bootstrapping on top of supervised SVM.

He shows that the supervised SVM classifier using various lexical and syntactic features can achieve promising classification accuracy. Another significant outcome has to do with reduce the need for labeled training data by means use of BootProject algorithm. More recently, Zeng et al. [20] show the best results in RE using Deep Learning techniques with 82.7% F-score surpassing SVM models with 82.2% F-score. They exploit a convolutional deep neural network (DNN) to extract lexical and sentence level features taking, finally, these two level features are concatenated to form the final extracted feature vector which fed the DNN.

In this paper, we address the relation extraction as a large-margin binary classification problem. In our work we integrate various tasks such as part-of-speech tagging, named entity recognition, syntactic trees extraction and entities characterization in a single model. We show how syntactic information improves the performance and create a features vector comprised of relations types, full parse trees and contextual entities information for Spanish language.

3 reACE Corpus/Support Vector Machine

Datasets for Generic Relation Extraction (reACE) was developed at The University of Edinburgh, Edinburgh, Scotland. It consists of English broadcast news and newswire data originally annotated for the ACE (Automatic Content Extraction) [4]. In ACE are define PERSON, ORG, LOCATION, FACILITY, GEO POLITICAL ENTITY (GPE), WEAPON entities. We detect relations between entities and next identify these entities considering the Relation Detection and Characterization (RDC) defined by NIST standard; ACE 2005 [18] and ACE 2007 [16].

Hachey et al. [6] describe ACE 2004 and ACE 2005 datasets, their standardization and building the reACE corpus. We translate reACE corpus using Google's web-based translation system. We use a python package called TextBlob [1] which has been used for connect to Google Translator services. Once the translation is completed for Spanish, we obtain a corpus that contains a collection of original texts in English and their translation. In this regard, we keep the sentences's order, named entities and relations labels from English reACE corpus.

Another fundamental issues for development our work is Support Vector Machines (SVM) which are a kind of large-margin binary classifiers [17]. According to Manning [9], an SVM is a vector-space-based machine learning method, where the goal is to find a decision boundary between two classes that is maximally far from any point in the training data. Basically, SVMs are binary classifiers.

Therefore, we must extend SVMs to multi-class using SVM^{struct} developed by Joachims [8]. We built SVM^{struct} models for detecting the relations, predicting the type of relations between every entity pairs within the same sentence from Spanish. As defined in the ACE evaluation, we only model explicit relations rather than implicit ones. For example:

El saxofonista estadounidense David Murray reclutó a Amidu Berry.

This sentence explicitly expresses a GPE-AFF relation, between entities “estadounidense” and David Murray. This relation is found according to the context information within this sentence.

The SVM models proposed need context information expressed as a vector. The vector consisting of values for some specific attribute, commonly called features. A input sentence is associated with a number of features and can be thought of as a characteristic property that can be used to distinguish one type of relation from another but is important noted that the features are not uniquely defined. Choosing the right features is key to successful a SVM.

4 Features Selection

The semantic relation is between two entities where each entity has a contextual features set, surroundings words of the entity and the syntactic features of sentence. We extract the lexical, contextual and syntactic features for the detection task. In this context, we distinguish two mentions; M1 for the first mention in which the first named entity is localized, M2 for the second mention in which the second named entity is localized. These mentions are divided by verbs within the sentence. The following are described the three types of features.

4.1 Local Features

These features concern named entities. In Table 1, we show 10 features related to entities, and the lexical information of verbs in the context of the two mentions M1 y M2. We also define features for locating the named entities for each mention M1 y M2.

Table 1. Local features in the sentence

Feature name	Value
NE	1 if sentence contains named entities. 0 if else
FNE	1 first named entity in sentence is defined. 0 if else
SNE	1 second named entity in sentence is defined. 0 if else
MW1	1 first named entity is multiword. 0 if else
MW2	1 second named entity is multiword. 0 if else
VB	1 sentence have a verb. 0 if else
Per	1 first person, 2 second person, 3 third person and 0 in the other case
Mood	1 Indicative, 2 Subjunctive, 3 Imperative, 5 Infinitive, 6 Gerund, 7 Participle and 0 in the other case
FNE categories	1 Person, 2 Organization, 3 Location/GPE, 4 WEA and 0 in the other case
SNE categories	1 Person, 2 Organization, 3 Location/GPE, 4 WEA and 0 in the other case

4.2 Contextual Features

The semantic relation is given between two named entities that are present for each mention. In this address, we define word features between the two mentions, the words before M1, the words before M2, the words after M1 y M2 and the words between two mentions. In the Table 2 we define features according context between the named entities M1 y M2.

Table 2. Contextual features from sentence

Feature name	Value
Before of FNE and SNE	1 if it was an adjective, 2 adverb 3 determinant, 4 noun 5 verb, 6 pronoun 7 conjunction, 8 preposition 9 abbreviations, 10 numbers, 11 interjection 12 punctuation, 13 multiword 0 in the other case
Before of SNE	1 if it was an adjective, 2 adverb 3 determinant, 4 noun 5 verb, 6 pronoun 7 conjunction, 8 preposition 9 abbreviations, 10 numbers, 11 interjection 12 punctuation, 13 multiword 0 in the other case
After of FNE	1 if it was an adjective, 2 adverb 3 determinant, 4 noun 5 verb, 6 pronoun 7 conjunction, 8 preposition 9 abbreviations, 10 numbers, 11 interjection 12 punctuation, 13 multiword 0 in the other case
After of SNE	1 if it was an adjective, 2 adverb 3 determinant, 4 noun 5 verb, 6 pronoun 7 conjunction, 8 preposition 9 abbreviations, 10 numbers, 11 interjection 12 punctuation, 13 multiword 0 in the other case
Tokens between entities	N words beetwen, first and second entity, 0 if there is only a named entity
Pairs of entities in the sentence	1 if it is PER-PER 2 if it is PER-ORG 3 if it is PER-GPE, 4 if it is PER-OTHER 5 if it is ORG-PER, 6 if it is ORG-ORG 7 if it is ORG-GPE, 8 if it is ORG-OTHER 9 if it is GPE-PER, 10 if it is GPE-ORG 11 if it is GPE-GPE, 12 if it is GPE-OTHER, 13 if it is OTHER-PER, 14 if it is OTHER-ORG 15 if it is OTHER-GPE 16 is OTHER-OTHER, 0 in the other case

4.3 Syntactic Features

The syntactic features are extract from the parse trees. As we mentioned above, we use Stanford CoreNLP toolkit [12] to generate the full parse tree and the semantic representation of each sentence. For example, we show the sentence (see below) with its respective parse tree from CoreNLP parsing in Fig. 1.

El saxofonista estadounidense David Murray reclutó a Amidu Berry.
(ROOT (sentence (sn (spec (da0000 El)) (grup.nom (nc0s000 saxofonista) (s.a (grup.a (aq0000 estadounidense))) (sn (grup.nom (np00000 David) (np00000 Murray))))) (grup.verb (vmis000 reclutó)) (sp (prep (sp000 a)) (sn (grup.nom (np00000 Amidu) (np00000 Berry)))) (fp .)))

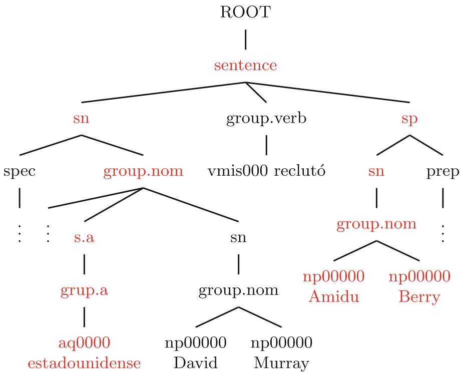

Fig. 1. Parse tree of a sentence from CoreNLP.

Thereafter, we extract the paths of phrase labels that connecting entity pairs and nominal phrases. Namely, first to reach the estadounidense adjective is extracted the path sn, group.nom, s.a, grup.a, aq0000 and to reach the Amidu Berry entity is extracted the path sp, sn, group.nom, np00000, np00000 path (see Fig. 1). These syntactic features were obtained in order to improve the SVM performance. Among other things, we use SVM-struct in order to measure the similarity between two syntactic trees and select the substructures that better describe the tree's structure [14].

5 Model's Architecture

The model' architecture have the threes traditional phases of a SVM classifier, preprocessing phase have as a result the feature vectors, training phase takes as input the vectors and provides a learning model file and finally, relation extraction phase classify the relationship between entities. Following we find greater detail of the phases of the Fig. 2.

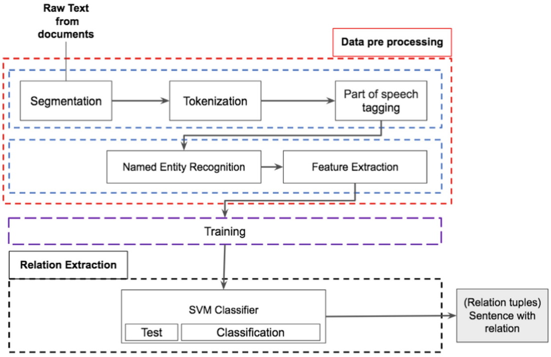

Fig. 2. Models's architecture

5.1 Pre-processing

Data pre-processing is the first phase of extraction process in which raw text is cleaned and prepared to information extraction. Specifically, to the raw text are carried out tasks such as tokenization, lemmatization, part-of-speech tagging, parsing, among others using CoreNLP toolkit [12] and Freeling 3.0 [15].

5.2 Training

Training of a SVM for relation extraction require the following steps (See Fig. 3a):

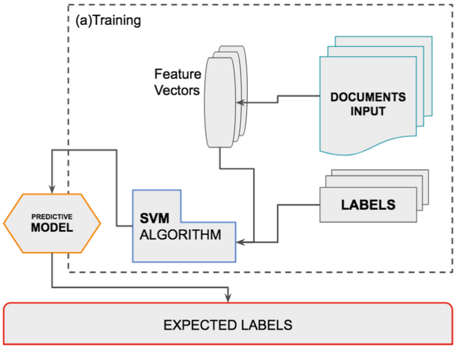

Fig. 3. Training steps

- Preprocess: The input to a SVM is a numeric vector. In order to train, first we need to represent each data instance as a vector. The preparation of training data is setting according to features as described in Sect. 4 where each feature vector containing all the features extracted from sentence.
- Kernel selection: a kernel function is selected depending of data. We use a linear kernel. The training data are linearly separable data.

- Predictive model: a SVM model file is generated after training like a standard format file. This file contains all patterns trained by SVM to annotate new relationship between entity pairs.

5.3 Relation Extraction

This phase allow us to extract relations between named entities from Spanish. We use several NER systems for Spanish in which are identify; Location (LOC), Person (PER) and Organization (ORG). [5,12,15]. We also define the test data with same feature vectors of the training model. Finally, we experiment with three models based on SVMs and classify relationship between entity pairs. Our models were trained and tested on the automatically translated reACE corpus exploring features than have been described in Sect. 4.

5.3.1 Baseline Model

We formulate the relation extraction as a multiclass classification problem using the SVM-*multiclass* [2]. Namely, we have converted the multiclass problem into a number of binary-class problems. We use the ONE vs ALL (OVA) formalism, which involves training n binary classifiers for a n-class problem.

The baseline model training have two approaches as follows:

- The model filter out the sentences that have classified with relation. A SVM-*struct* (class: NON-REL, REL) is trained on the entire dataset.
- The training data with all relation types is used to train OVA classifiers, one of which is the NON-REl vs REL classifier. Where REL is each relation type in dataset.

With this strategy the SVM-*struct* identifies relation in the sentences. While the OVA classifies relation types in the sentence improving the training time. For the development of the baseline system we need to select a suitable feature set. Our baseline model use the set of features described in Sect. 4.1. We build the baseline classifier using a SVM where numeric representation of the features is used. In this representation a particular sentence is converted into several numeric features. For all experiments of the baseline model we use the local features of the Table 1 and use SVM*struct*[1] which is an SVM implementation that can model complex (multivariate) input data such as trees, sequences, or sets. [7]

5.3.2 Syntactic Model

Syntactic model is combination of local, contextual and syntactic features (see Sect. 4.3). We introduce features based on contextual and semantic information as we have explained in Table 2 and Sect. 4.3. The following are features's combinations for each model:

[1] https://www.cs.cornell.edu/people/tj/svm_light/svm_struct.html.

1. M1 model = Local features (Baseline) + Context features
2. Base2 model = Syntactic features
3. M2 model = Local features (Baseline) + Context features + Syntactic features

On all the model improvements, we take advantage of the fact that SVM allow high-dimensional feature spaces and the implementation described in Sect. 3 can predict complex objects like trees. In this model, the feature set decisions are taken independently of each other. We perform experiments using different sets of features and evaluate the incremental performance improvement they provide on classifier. Following, we report the results which show significant improvement in performance for each model.

6 Results

To measure relation extraction between entities we use F1-score which is the weighted harmonic mean of precision (P) and recall (R). These three performance metrics are common used in machine learning. Precision is the fraction of relation classified to class C that belong to class C indeed, while recall is the fraction of relation in class C that are correctly retrieved.

F1-score (F1) takes both precision and recall of classification into account, and hence can be considered as a measure of interest, with maximum and minimum value of 1 and 0, respectively. The general expressions for precision, recall and F1-score:

$$R = \frac{TP}{TP + FN} \qquad P = \frac{TP}{TP + FP} \qquad F1 = \frac{2 \cdot P \cdot R}{P + R}$$

To visualize the performance of the classifier, we also introduce Receiver Operating Characteristics (ROC) curve, AUC (Area Under ROC Curve) and Precision-Recall curve. Larger values of F1-score and area under the ROC curve indicate better classifier performance.

6.1 Baseline Model Performance

Table 3 shows the baseline model performance on the relation extraction tasks mentioned earlier; these results are based on local features extracted with several NLP tools. We report the precision(P), recall (R) and the F1 scores. This test set was automatically translated like as described in the Sect. 3.

Table 3. Baseline performance

Feature	P	R	F1-score
SVM (Local features)	46.79%	99.96%	63.74%

During the evaluation we have followed the exact match strategy; which means a detected relation is assumed as correct if it matches exactly with the corresponding test data relation in terms of the category and sentence.

6.2 Syntactic Model Performance

Table 4 shows the performance each model has on the relation extraction task, when added features to baseline and when tree structure are used.

Table 4. Results for M1, Base2 and M2 model.

Feature	P	R	F1-score
SVM (Local features)	46.79%	99.96%	63.74%
SVM (context features)	67.03%	49.80%	57.14%
SVM (Syntactic features)	82.90%	67.67%	74.51%
SVM (All features)	80.18%	70.89%	75.25%

In short, in the Fig. 4 the curve comparative between two models M1 and M2 show that there is a highest growth of the M2 curve to the M1 curve, this means that in the M2 model the quantity of False Positives are minor than True Positives, therefore, precision measure is major than recall measure. In the M1 model, the False Positives and true Positive are very close, therefore, precision is smaller than recall.

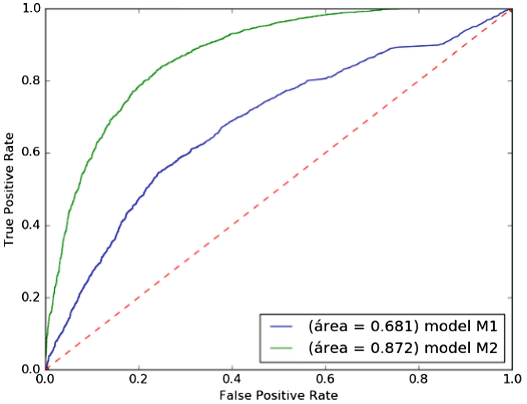

Fig. 4. The receiver operating characteristic (ROC) curve comparing M1 and M2 models on set test data

7 Conclusions

In this paper, we aim to implement a Support Vector Machine to classify to relation between named entities for Spanish language using an dataset translated from English language. As baseline model, a total of ten features were setting and for improve the model also were add contextual and syntactic features to complete the classification task in Spanish. We show that SVM achieve a performance comparable to the state of the art in automatically translated collections using syntactic and semantic features including the structure of the tree.

Given the features of the sentence in Spanish, the input vector is classified with our SVM model, first indicate whether relation exist, second which relation is in sentence. We believe that our work can provide important insights to applications using relations among two entities and present a dataset relevant for RE task in Spanish language. We have experimented with four models and we have adjusted the number of feature to improve our SVM classifier. We describes our model features as local, contextual and syntactic. We found that using the SVM with these features was helpful in increasing the F1 score. Besides F1-score, we also take precision and recall as evaluation indicators the classifier performance. We further plot the ROC curves and Precision-Recall curves to visualize the performance models.

According to the results and the problems we encountered during our experiments, we give two potential directions as future work. (1) improving performance of the classifier component by exploiting other features and (2) exploring RE task with deep learning architectures. Overall, the results with deep learning are feasible prospect for the detection and classification in RE task. It should be noted that our experiments not analyze the impact of automatically translate collections of training and test and we will also like to analyze it and to compare the performance using others RE system for Spanish.

Acknowledgments. The authors would like to acknowledge the systems and Computing Engineering school, Faculty of Engineer, The Universidad del Valle of Cali, Colombia.

References

1. Textblob: Simplified text processing. http://textblob.readthedocs.org/. Accessed 22 Feb 2015
2. Crammer, K., Singer, Y.: On the algorithmic implementation of multiclass kernel-based vector machines. J. Mach. Learn. Res. **2**(Dec), 265–292 (2001)
3. Culotta, A., Sorensen, J.: Dependency tree kernels for relation extraction. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, ACL 2004, Stroudsburg, PA, USA. Association for Computational Linguistics (2004)
4. Doddington, G.R., Mitchell, A., Przybocki, M.A., Ramshaw, L.A., Strassel, S., Weischedel, R.M.: The automatic content extraction (ACE) program-tasks, data, and evaluation (2004)
5. Gutiérrez, R., Castillo, A., Bucheli, V., Solarte, O.: Named entity recognition for Spanish language and applications in technology forecasting reconocimiento de entidades nombradas para el idioma español y su aplicación en la vigilancia tecnológica (2015)

6. Hachey, B., Grover, C., Tobin, R.: Datasets for generic relation extraction. Nat. Lang. Eng. **18**(1), 21–59 (2012)
7. Joachims, T.: Support vector machines for complex outputs (2008). http://www.cs.cornell.edu/people/tj/svm_light/svm_struct.html
8. Joachims, T.: Making large-scale SVM learning practical. Technical report, SFB 475: Komplexitätsreduktion in Multivariaten Datenstrukturen, Universität Dortmund (1998)
9. Jurafsky, D., Martin, J.H.: Speech and Language Processing. Pearson Education, London (2009). International edition
10. Kambhatla, N.: Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In: Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions, ACLdemo 2004, Stroudsburg, PA, USA. Association for Computational Linguistics (2004)
11. Kumar, S.: A survey of deep learning methods for relation extraction. CoRR, abs/1705.03645 (2017)
12. Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55–60 (2014)
13. Miller, S., Fox, H., Ramshaw, L., Weischedel, R.: A novel use of statistical parsing to extract information from text. In: Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference, NAACL 2000, Stroudsburg, PA, USA, pp. 226–233. Association for Computational Linguistics (2000)
14. Moschitti, A.: A study on convolution kernels for shallow semantic parsing. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, p. 335. Association for Computational Linguistics (2004)
15. Padró, L., Stanilovsky, E.: Freeling 3.0: towards wider multilinguality. In: LREC2012 (2012)
16. Song, Z., et al.: ACE 2007 multilingual training corpus LDC2014t18. Linguistic Data Consortium, Philadelphia (2014)
17. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995). https://doi.org/10.1007/978-1-4757-3264-1
18. Walker, C., et al.: ACE 2005 multilingual training corpus LDC2006t06. Linguistic Data Consortium, Philadelphia (2006)
19. Zelenko, D., Aone, C., Richardella, A.: Kernel methods for relation extraction. J. Mach. Learn. Res. **3**, 1083–1106 (2003)
20. Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J.: Relation classification via convolutional deep neural network. In: Proceedings of COLING 2014, The 25th International Conference on Computational Linguistics: Technical Papers, pp. 2335–2344 (2014)
21. Zhang, Z.: Weakly-supervised relation classification for information extraction. In: Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management, CIKM 2004, pp. 581–588. ACM, New York (2004)