Creating a standard for evaluating Distant Supervision for Relation Extraction

Azad Abad[1] and **Alessandro Moschitti**[2,1]
[1]Department of Information Engineering and Computer Science, University of Trento,
[2]Qatar Computing Research Institute
abad@disi.unitn.it, amoschitti@gmail.com

Abstract

English. This paper defines a standard for comparing relation extraction (RE) systems based on a Distant Supervision (DS). We integrate the well-known New York Time corpus with the more recent version of Freebase. Then, we define a simpler RE system based on DS, which exploits SVMs, tree kernels and a simple one-vs-all strategy. The resulting model can be used as a baseline for system comparison. We also study several example filtering techniques for improving the quality of the DS output.

Italiano. *Questo articolo definisce uno standard per comparare sistemi per l'estrazione di relazioni (ER) basati su Distant Supervision. In questo lavoro, integriamo il famoso corpus New York Time con la recente versione di Freebase. Quindi, definiamo in sistema di ER che usa DS basato su SVMs, tree kernels e la strategia uno-contro-tutti. Il modello risultante puó essere usato come baseline per la comparazione di sistemi. In aggiunta, studiamo diverse tecniche di filtraggio degli esempi prodotti dalla DS per migliorare la qualitá del suo output.*

1 Introduction

Relation Extraction (RE) is a well-known Natural Language Processing subarea, which aims at extracting relation types between two named entities from text. For instance, in the sentence:*"Alaska is a U.S. state situated in the North American continent."*, the identified relation type between two entity mentions can be denoted by a tuple r<e_1,e_2>$\in E \times E$, where the tuple name *r* is the relation type and e_1 and e_2 are the entities that participate in the relation.

$$\underbrace{Location/Contains}_{r} < \underbrace{Alaska}_{e1}, \underbrace{United\,States}_{e2} >$$

Currently, supervised learning approaches are widely used to train relation extractors. However, manually providing large-scale human-labeled training data is costly in terms of resources and time. Besides, (i) a small-size corpus can only contains few relation types and (ii) the resulting trained model is domain-dependent.

Distance Supervision (DS) is an alternative approach to overcome the problem of data annotation (Craven et al., 1999) as it can automatically generate training data by combining (i) a structured Knowledge Base (KB), e.g., Freebase[1] with a large-scale unlabeled corpus, C. The basic idea is: given a tuple r<e_1,e_2> contained in a referring KB, if both e_1 and e_2 appear in a sentence of C, that sentence is assumed to express the relation type r, i.e., it is considered a training sentence for r. For example, given the KB relation, `president(Obama,USA)`, the following sentence, *Obama has been elected in the USA presidential campaign*, can be used as a positive training example for `president(x,y)`.

However, DS suffers from two major drawbacks: first, in early studies, Mintz et al. (2009) assumed that two entity mentions cannot be in a relation with different relation types r_1 and r_2. In contrast, Hoffmann et al. (2011) showed that 18.3% of the entities in Freebase that also occur in the New York Times 2007 corpus (NYT) overlap with more than one relation type.

Second, although DS method has shown some promising results, its accuracy suffers from noisy training data caused by two types of problems (Hoffmann et al., 2011; Intxaurrondo et al., 2013; Riedel et al., 2010): (i) possible mismatch between the sentence semantics and the relation type mapped in it, e.g., the KB correct relation, `located_in(Renzi,Rome)`, cannot be mapped into the sentence, *Renzi does not love the Rome soccer team*; and (ii) coverage of the KB,

[1] http://www.freebase.com/

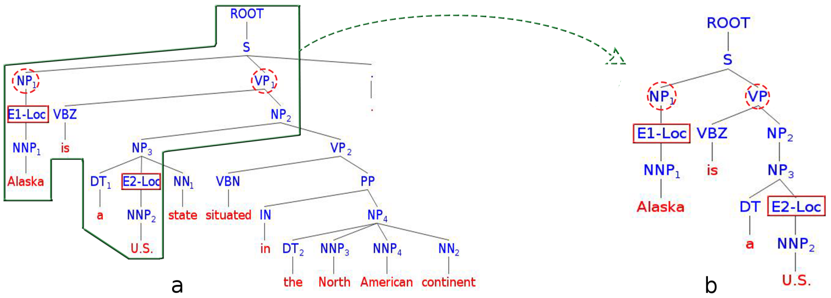

Figure 1: a) The constituent parse tree of the example sentence where ”E1-Loc” denotes the source entity mentions and ”E2-Loc” denotes the target entity. b) PT relation instance space of the sentence.

e.g., a sentence can express relations that are not in the KB (this generates false negatives).

Several approaches for selecting higher quality training sentences with DS have been studies but comparing such methods is difficult for the lack of well-defined benchmarks and models using DS.

In this paper, we aim at building a standard to compare models based on DS: first of all, we considered the most used corpus in DS, i.e., the combination of NYT and Freebase (NYT-FB).

Secondly, we mapped the Freebase entity IDs used in NYT-FB from the old version of 2007 to the newer Freebase 2014. Since entities changed, we asked an annotator to manually tag the entity mentions in the sentence. As the result, we created a new dataset usable as a stand-alone DS corpus, which we make available for research purposes.

Finally, all the few RE models experimented with NYT-FB in the past are based on a complex conditional random fields. This is necessary to encode the dependencies between the overlapping relations. Additionally, such models use very particular and sparse features, which make the replicability of the models and results complex, thus limiting the research progress in DS. Indeed, for comparing a new DS approach with the previous work using NYT-FB, the researcher is forced to re-implement a very complicated model and its sparse features. Therefore, we believe that simpler models can be very useful as (i) a much simpler reimplementation would enable model comparisons and (ii) it would be easier to verify if a DS method is better than another. In this perspective, our proposed approach is based on convolution tree kernels, which can easily exploit syntactic/semantic structures. This is an important aspect to favor replicability of our results.

Moreover, our method differers from previous state of the art on overlapping relations (Riedel et al., 2010) as we apply a modification of the simple one-vs-all strategy, instead of the complex graphical models. To make our approach competitive, we studied several parameters for optimizing SVMs and filtering out noisy negative training examples. Our extensive experiments show that our models achieve satisfactory results.

2 Related Work

Extracting relations from the text has become popular in IE community. In fully-supervised approach, all the instances are manually labeled by humans and it has been the most popular method so far (Zelenko et al., 2003; Culotta and Sorensen, 2004; Kambhatla, 2004). In semi-supervised approach, initially a small number of seed instances are manually annotated and used to extract the patterns from a big corpus (Agichtein and Gravano, 2000; Blum and Mitchell, 1998).

Distant Supervision (DS) has emerged to be a popular method for training semantic relation extractors. It was used for the first time in the biomedical domain (Craven et al., 1999) and the basic idea was to extract binary relations between protein and cell/tissues by using Yeast Protein Database (YPD) corpus. This method is getting more and more popular and different types of RE problems are being addressed (Bunescu and Mooney, 2007; Mintz et al., 2009; Riedel et al., 2010; Nguyen and Moschitti, 2011; Hoffmann et al., 2010; Riedel et al., 2013; Surdeanu et al., 2012; Hoffmann et al., 2011). Among others, tree kernels (TKs) have been widely used in supervised and weakly supervised setting and shown promising results. (Bunescu and Mooney, 2005; Nguyen et al., 2009; Nguyen and Moschitti, 2011; Bunescu and Mooney, 2005; Zelenko et al., 2003)

3 Basic RE using SVMs and TKs

Support Vector Machines (SVMs) are linear supervised binary classifiers that separate the class boundaries by constructing hyperplanes in a multidimensional space. They can also be used in nonseparable linear space by applying kernel functions. Tree kernels (TKs) (Collins et al., 2001) have been proved to achieve state-of-the-art in relation extraction (Zhang et al., 2006b). Different TKs have been proposed in the past (Moschitti, 2006). We modeled our RE system by using feature vectors along with syntactic/semantic trees (see (Zhang et al., 2006a; Nguyen et al., 2009)).

3.1 Feature Vectors

In our experiment, we used the features proposed by Mintz et al. (2009). It consists of two standard lexical and syntactic feature levels. Lexical/syntactic features extracted from a candidate sentence are decorated with different syntactic features such as: (i) Part of Speech (POS); (ii) the window of k words of the left and right of matched entities; (iii) the sequences of words between them; and (iv) finally, syntactic features extracted in terms of dependency patterns between entity pairs. The proposed features yield low-recall as they appear in conjunctive forms but at the same time they produce a high precision.

3.2 Tree Kernels for RE

We used the model proposed in (Zhang et al., 2006a). This, given two relation examples, R_1 and R_2, computes a composite kernel $K(R_1, R_2)$, which combines a tree kernel with a linear kernel. More formally:

$$K(R_1, R_2) = \alpha \vec{x_1} \cdot \vec{x_2} + (1 - \alpha) K_T(T_1, T_2),$$

where α is a coefficient that assigns more weight to the target kernel, $\vec{x_1}$ and $\vec{x_2}$ are feature vectors representing the two relations R_1 and R_2, respectively, and $K_T(T_1, T_2)$ is the tree kernel applied to the syntactic/semantic trees representing the two relations. T_i $(i = 1, 2)$ is the minimal subtree containing the shortest path between the two target entity mentions. Figure 1 shows a sentence tree (part a) and its associated tree (part b).

4 Experiments

Corpus. We trained our system on the NYT news wire corpus (Sandhaus, 2008). The original corpus includes 1.8 million articles written and published by the NYT between January 1987 and June 2007. We used the same subset of data as Riedel et al. (2010). The data set consists of two parts for training and the test, where the first part refers to the years 2005-2006 of the NYT whereas the second refer to the year 2007.

In the corpus provided by Riedel et al. (2010), instead of the entity mentions, their corresponding IDs in Freebase have been tagged (this because of previous copyright issues). The old version of Freebase 2007 is not available anymore and in many cases the IDs or entities have changed in Freebase 2014. So, it was not possible to combine NYT with the newer Freebase to apply DS. To deal with this problem, we mapped the old Freebase IDs with Freebase 2014 and, if the entities were not the same, we asked an annotator to manually tag the entity mentions in the sentence. As the result, we created a new dataset that is mapped with Freebase 2014 and it is usable as a stand-alone DS corpus, which we are making freely available[2]. Overall, we found 4,700 relations in the training set and 1,950 in the test set. The number of positive and negative examples is heavily imbalanced (1:134). So, we applied simple filtering to discard noisy negative examples from the training set.

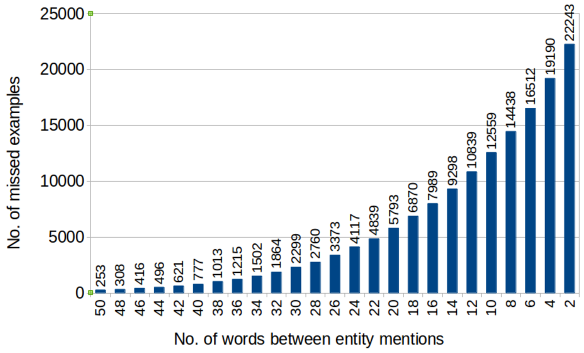

Figure 2: Recall of positive examples with respect to word distance between entity mentions.

4.1 Data Pre-processing

In the introduction, we pointed out that (i) some sentences containing the target entities may not semantically realize the target relation and (ii) other sentences express a correct relation not in the KB. We tackle such problems by applying sentence filtering and enriching the relations of previous KB.

Sentence Filtering. We used four levels of noise cleaning to remove potential incorrect sentences from the corpus. More specifically, we remove a sentence if:

- The distance between the two target entity mentions is more than k words (e.g., 26). We set the k threshold value equal to 10% of the total number of positive examples as shown in Figure 2.
- The number of tagged entities between the entity mentions are greater than a constant h (e.g.,10).
- None of the entity mentions in the sentence appeared in positive examples before, i.e., at least one of the entity in the negative example has to be

[2] http://goo.gl/M7I7fL

Relation Type	P%	R%	F1%
company/founders	66.7	11.4	19.5
location/contains	13.5	40.4	20.3
person/company	11.6	60.7	19.5
company/place_founded	20.0	6.7	10.0
person/place_lived	10	20.2	13.46

Table 1: Precision and recall of different relation types.

	P%	R%	F1%
Mintz++	31.28	15.43	20.67
Intxaurrondo et al.	29.79	17.48	22.03
Basic SVM	12.4	7.6	9.5
Our Model	11.3	23.0	15.1
Our Model + filtering	13.2	22.5	16.6

Table 2: Results for different models

in a relation with another entity (i.e., it has to be part of previously generated positive examples).

- The same entity pairs were in a relation in positive examples but with different relation type (Overlap Relation). For instance, in the mention *Edmonton, Alberta , one of six Canadian N.H.L. markets, is the smallest in the league.*, the entity mentions $<Edmonton, Alberta>$ are in relations with two relation types: *Province/Capital* and *Location/Contains*. Thus, to train Rel. 1, all the instances of Rel. 2 are removed and viceversa.

Enriching KB with new relations types. We analyzed the entity pairs in the sentences of our corpus with respect to the relations in Freebase 2007. We discovered that many pairs receive no-relation because they did not exist in Freebase 2007. This creates many false negative (FN) errors in the generation of training data. In the new release of Freebase many new relations are added, thus we could recover many of such FNs. However to keep the compatibility with the previous NYT-FB corpus, we simply discard such examples from the training set (instead of including them as new positive examples). We could match 1,131 new pairs, which are around 1.4% of the total number of the matched pairs in the training set. Overall, 3,373 mentions from the positive examples and 11,818 mentions from negative examples are discarded from the training set.

4.2 NLP Pipeline

Configurations. We use standard NLP tools in our pipeline: we parsed all the sentences using the Charniak parser (Charniak, 2000) and tagged the named entities with the Stanford NER toolkit (Finkel et al., 2005) into 4 classes (e.g. Person, Location, Organization and Other). We used SVM-Light-TK [3] for training our classifiers, and employed the one-vs-all strategy for multiclass classification but with some modifications to handle the overlap relations: instead of selecting the class with the highest score assigned by the classifier to sentences, we selected all the labels if the assigned scores are larger than a certain threshold (e.g., 0). Hence, the classifier can select more than one class for each example. We normalize both the tree kernel and the feature vectors.

Parameter Optimization. The SVM accuracy is highly influenced by selecting the suitable values for the cost-factor (option j) and trade-off (option c) parameters. As we mentioned, the dataset is very imbalance thus we tuned the j parameter to outweigh the positive example errors with respect to the negative examples during training. We used 30% of our training set as a development set to optimize the parameters. Then, the best combination of c and j values with the highest F-measure in the development set are used to train the classifier.

Evaluation. We compared our model with the two recent state-of-the-art algorithms such as: (1) Mintz++ (Surdeanu et al., 2012), which is an improved version of the original work by Mintz et al. (2009) and (2) Intxaurrondo et al. (2013). The results for different classes and the overall Micro-average F1 are shown in tables 1 and 2, respectively. Noted that, due to lack of space, only the performance of the most populated 5 classes out of 52 are reported. The results show that (i) our model improves the micro-average F1 of the basic RE implementation (basic SVM), i.e., by Zhang et al. (2006b), by more than 7 absolute percent points, i.e., 74% relative; and (ii) applying our simple filtering approach improves our model by 1.5% absolute points. However, our models are still outperformed by the state of the art: this is not critical considering that our aim is to build simpler baseline systems.

5 Conclusion

We have proposed a standard framework, simple RE models and an upgraded version of NYT-FB for more easily measuring the research progress in DS research. Our RE model is based on SVMs, can manage overlapping relations and exploit syntactic information and lexical features thanks to tree kernels. Additionally, we have shown that filtering techniques applied to DS data can discard noisy examples and significantly improve the RE accuracy.

[3] http://disi.unitn.it/moschitti/Tree-Kernel.htm

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 85–94. ACM.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM.

Razvan C Bunescu and Raymond J Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 724–731. Association for Computational Linguistics.

Razvan Bunescu and Raymond Mooney. 2007. Learning to extract relations from the web using minimal supervision. In *Annual meeting-association for Computational Linguistics*, volume 45, page 576.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 132–139. Association for Computational Linguistics.

Michael Collins, Nigel Duffy, et al. 2001. Convolution kernels for natural language. In *NIPS*, volume 2001, pages 625–632.

Mark Craven, Johan Kumlien, et al. 1999. Constructing biological knowledge bases by extracting information from text sources. In *ISMB*, volume 1999, pages 77–86.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 423. Association for Computational Linguistics.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics.

Raphael Hoffmann, Congle Zhang, and Daniel S Weld. 2010. Learning 5000 relational extractors. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 286–295. Association for Computational Linguistics.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 541–550. Association for Computational Linguistics.

Ander Intxaurrondo, Mihai Surdeanu, Oier Lopez de Lacalle, and Eneko Agirre. 2013. Removing noisy mentions for distant supervision. In *Proceedings of the 29th "Congreso de la Sociedad Española para el Procesamiento del Lenguaje Natural" (SEPLN 2013)*.

Nanda Kambhatla. 2004. 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*, page 22. Association for Computational Linguistics.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Alessandro Moschitti. 2006. Efficient convolution kernels for dependency and constituent syntactic trees. In *Machine Learning: ECML 2006*, pages 318–329. Springer.

Truc-Vien T Nguyen and Alessandro Moschitti. 2011. End-to-end relation extraction using distant supervision from external semantic repositories. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 277–282. Association for Computational Linguistics.

Truc-Vien T Nguyen, Alessandro Moschitti, and Giuseppe Riccardi. 2009. Convolution kernels on constituent, dependency and sequential structures for relation extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1378–1387. Association for Computational Linguistics.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas.

Evan Sandhaus. 2008. The new york times annotated corpus ldc2008t19. philadelphia: Linguistic data consortium.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 455–465. Association for Computational Linguistics.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *The Journal of Machine Learning Research*, 3:1083–1106.

Min Zhang, Jie Zhang, and Jian Su. 2006a. Exploring syntactic features for relation extraction using a convolution tree kernel. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 288–295. Association for Computational Linguistics.

Min Zhang, Jie Zhang, Jian Su, and Guodong Zhou. 2006b. A composite kernel to extract relations between entities with both flat and structured features. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 825–832. Association for Computational Linguistics.