Barrier Features for Classification of Semantic Relations

Anita Alicante
University "Federico II", Naples, Italy
anita.alicante@unina.it

Anna Corazza
University "Federico II", Naples, Italy
corazza@na.infn.it

Abstract

Approaches based on machine learning, such as Support Vector Machines, are often used to classify semantic relations between entities. In such framework, classification accuracy strongly depends on the set of features which are used to represent the input to the classifier. We are proposing here a new type of features, namely the *barrier features*, which can be used in addition to more usual features, such as n-grams of PoS, word suffixes and prefixes, hypernyms from WordNet etc., and to the parse tree of the whole sentence. Barrier features aim at giving a compact representation of the context of each entity involved in the relation. The effectiveness of the new features is assessed on documents from the TREC data set annotated by Roth and Yih. The obtained results show not only that the performance of the proposed approach are state-of-the-art but also that such improvement is due to the introduction of the barrier features.

1 Introduction

Different approaches have been proposed for the identification of entities and relations in text. In this paper we focus only on the task of classification of semantic relations, that is of assigning to each semantic relation a label taken from a finite set, and therefore we assume that pairs of related entities are given us together with their labels. In general, not all relation labels are compatible with every pair of entity types. Table 3 reports all such constraints in the considered data set. Thus not all relation labels can be compatible with all entity pairs and therefore we decompose the problem in several binary classifications, one for each possible relation label and apply Support Vector Machines (SVMs) to each of these subtasks.

Moreover, by applying SVMs with a combination of different kernel functions we can handle together different kinds of information, both structured and not. In fact, we applied tree kernels (Moschitti, 2006) to the whole sentence parse tree and a linear kernel to a vector of binary features extracted from the words surrounding each of the involved entities. Among the latter, we introduce a novel kind of binary feature, which we call *barrier features* and experimentally prove that they improve classification performance. Although inspired by the barrier rules of the constraint grammar framework proposed in (Karlsson et al., 1995) for PoS tagging, barrier features have been completely redesigned for this task as binary values rather than rules.

The experimental assessment of the approach is performed on the newswire documents annotated by Roth and Yih (Roth and Yih, 2004). The best published results for relation classification on this data set have been obtained by the $M_{O|K}$ system, described in (Giuliano et al., 2007). The following Section 2 is devoted to the discussion of related works. Afterwards, the approach we are proposing is presented in Section 3, with a discussion of the adopted features, and in particular of barrier features. In Section 4 the experimental assessment is described. In the final section some planned development of the presented results are considered.

2 Related work

In the past few years a lot of works have been devoted to relation extraction and classification. Because of space limitation, we are giving a preference here to systems which have been assessed on the Roth and Yih data set. Systems assessed on other data sets include (Beamer et al., 2007; Davidov and Rappoport, 2008) for the Task 4 of SemEval07, (Rink and Harabagiu, 2010) for the Task08 of SemEval10 and (Kambhatla, 2004; Culotta and Sorensen, 2004; Guodong et al., 2005;

Proceedings of Recent Advances in Natural Language Processing, pages 509–514,
Hissar, Bulgaria, 12-14 September 2011.

GuoDong et al., 2006; Qian et al., 2008) for the Automatic Content Extraction (ACE), which is not freely available.

Systems devoted to relation extraction and classification usually first extract and label entities and afterwards relations. An important exception to this two pass approach is represented by (Roth and Yih, 2007), where entity and relation extraction and labeling are integrated and (Kate and Mooney, 2010) where a new method for joint entity and relation extraction is presented using a "card-pyramid" graph.

Most relation labeling systems are based on some machine learning approach, and build a classifier which associates a label to a representation of the input sentence. In such approaches the representation of the input is crucial, as only the information it contains can be used for labeling. Nearly all such systems consider some form of parsing: the complete parse tree of the input sentence is considered, among the others, by (Miller et al., 2000) and (Kambhatla, 2004), which considers both constituency and dependency parse trees. Another approach based on dependency parse trees is presented in (Reichartz et al., 2009). Systems that instead of the complete parse tree only consider some form of shallow parsing include (Giuliano et al., 2007) and (Zhang et al., 2005).

We compare our performance with the $M_{O|K}$ on the Roth and Yih data set (Giuliano et al., 2007). In addition to presenting a novel approach to relation extraction and labeling, they evaluate the effect of automatic named-entity recognition on its performance. Their approach is based on shallow linguistic features, which are combined with semantic information, such as WordNet hypernym relations of the candidate entities. Kernels are employed to combine two different information sources: the global context where the two entities appear and (independently) the two local contexts of the entities. A specific kernel function is associated to each of the different types of information.

3 The proposed approach

3.1 Problem definition

In this subsection, we briefly introduce a few definitions together with some examples. A sentence of length n is a string of n tokens $S = t_1 t_2 \dots t_n$ and can include a number $N \geq 0$ of entities $\{E_1, E_2, \dots, E_N\}$, each corresponding to a sequence of consecutive tokens, that is a substring of S. The entity indexes follow their order in the sentence, and each entity is labeled by an *entity-type* in a finite set E of labels. Although in the corpus we used for assessment entities are also represented by noun phrases, this definition is more general.

A subset of all ordered entity pairs corresponds to relations: $R_{i,j} = (E_i, E_j)$; E_i is called *agent* and E_j *target*, where the entities E_i and E_j can be composed by one or more tokens of the sentence and E_i can either precede or follow E_j. This definition excludes cross-sentence relations. A label taken from a finite set R of possible labels is associated to each relation. We are considering the task of associating the correct label to each relation, that is the *classification* task of *sematic relations*.

For the sake of clarity, let us consider the example sentence s_1 of Table 1. It contains four different relations containing six entities, namely (e_1, e_2) with label "work for", (e_2, e_3) with label "orgbased in", (e_4, e_5) labeled as "work for", and (e_5, e_6) for "orgbased in". Indeed, entities e_2 and e_5 are involved in two different relations.

3.2 The proposed solution

For each possible relation label we build a binary classifier based on SVMs which takes as input both a feature vector and the parse tree of the whole sentence. The former refers to the two input entities and its elements are therefore called *entity features*, while the latter refers to the whole sentence. Entity features include word and PoS unigrams, PoS bigrams and trigrams, word prefixes and suffixes, word length, and a set of word features indicating whether the initial letter is upper case, whether all letters are upper or lower case and whether the token contains a period or number or hyphen. Furthermore, we also included the most likely WordNet[1] (Fellbaum, 1998) sense tag for each token involved in the entity, which always corresponds to the first one in the list of possible senses, together with all the hypernyms.

These two sets of features are combined in the SVM-based classifier by integrating two kernels, namely tree kernels (Moschitti, 2006) and a linear kernel. The former is applied to the parse tree and evaluate the similarity between two trees in terms

[1] http://wordnet.princeton.edu/

s_1 Also being considered are $\langle e_1\rangle$*Judge Ralph K. Winter* $\langle/e_1\rangle$of the $\langle e_2\rangle$*2nd U. S. Circuit Court of Appeals*$\langle/e_2\rangle$ in $\langle e_3\rangle$*New York City* $\langle/e_3\rangle$ and $\langle e_4\rangle$*Judge Kenneth Starr* $\langle/e_4\rangle$ of the $\langle e_5\rangle$*U. S. Circuit Court of Appeals*$\langle/e_5\rangle$ for the $\langle e_6\rangle$*District of Columbia*$\langle/e_6\rangle$, said the source, who spoke on condition of anonymity.

s_2 The/DT spy/NN ,/, high-ranking/JJ $\langle e2\rangle$Korean/JJ CIA/NNP$\langle/e2\rangle$ official/JJ $\langle e1\rangle$Sohn/NNP Ho/NNP Young/NNP$\langle/e1\rangle$,/, wanted/VBD to/TO defect/VB ....

Table 1: *Example sentences taken from the Roth and Yih data set used for assessment.*

of the number of fragments they have in common; the latter has been chosen in the system tuning phase as described in Section 4. The same weight is associated to the two kernels.

3.2.1 Barrier features

In addition to these, we consider a novel kind of features, which we call *barrier features*, to model the context of each token in entities. Their definition is based on the set of PoS tags in a window surrounding the token. The length of the window varies and is based on the PoS's of the corresponding tokens: for each token in the entity (*trigger*), an *endpoint* token is chosen on the basis of the PoS of the trigger. In fact, for each token in the entity the corresponding endpoint is defined as the closest preceding or following token having one of the PoS associated with the PoS of the considered token. Such (trigger PoS, endpoint PoS) pairs are predefined and depend on the considered language: in the experiments we used the ones reported in Table 2.

In the task we are considering here, barrier features aim at describing the syntactic context of tokens in entities, which can only be nouns or adjectives. Therefore, we only considered patterns for this PoS, while completely disregarding other important PoS tags including verbs. On the other hand, endpoints try to bound the interesting context of the considered trigger. The choice of the pairs (trigger PoS, endpoint PoS) has been inspired by the corresponding barrier rules. We think that their favourable impact is connected to their complementarity to simpler features like bigrams and trigrams on one side and the complete parse tree on the other. We plan to explore the possibility of considering other pairs in the future.

In the experiments described in Section 4 barrier features only consider the case where the endpoint token precedes the entity. An entity token can have several endpoints and a new barrier feature corresponding to the set of PoS's between the endpoint and the token is introduced for every possible endpoint. If no endpoint is found before the entity token, the set of all the PoS tags from the beginning of the sentence to this entity are considered. As the barrier features are based on *sets* of tags, order and possible repetitions of tags are not considered.

3.3 Smoothing

A very large number of different barrier features can occur in addition to all other usual features and therefore the choice of the smoothing strategy is crucial. First of all, if the feature we observe in the input to the classifier does not exist in the set of features collected on the training set, then we consider all barrier features having the same (trigger PoS, endpoint PoS) pair and a PoS's set including the considered one. A side effect of this strategy is that more than one barrier feature can be positive (equal to 1) at the same time.

Furthermore, we introduce for every kind of feature a new feature UNKNOWN which intuitively corresponds to the unseen (rare) case. We train this feature by cumulating all cases having less than 3 occurrences. Therefore, there is an UNKNOWN feature for barriers, one for word unigrams, one for PoS unigrams, and so on. Whenever an input would not activate any value for a given type of features, the corresponding UNKNOWN feature is set.

In the sentence s_2 in Table 1 the relation corresponding to the entity pair (e_1, e_2) is labeled as *work for*. As entity e_2 is composed by two tokens ("Korean CIA") the corresponding feature vector results from the OR combination of the features corresponding to each token. If the entity were composed by only one token, the feature vector would only contain 1's in correspondence of the features computed for this token.

Thus, features based on words are extracted from the window "The/DT spy/NN ,/, high-ranking/JJ Korean/JJ CIA/NNP". Barrier features construction is based on a window whose length is not predetermined, but depends on the PoS's of

the tokens preceding the one we are considering, in this case "CIA". Since "CIA" PoS is NNP, we apply the first rule reported in Table 2: the endpoint is the closest determiner preceding the token *CIA*, namely *The*. In this case the endpoint does not belong to the entity, but this is not always so. The resulting barrier feature is then given by the set {*JJ, NN, ,*}, and contains, as discussed, only one repetition of *JJ*, corresponding to the tokens *high-ranking/JJ Korean/JJ* , *spy/NN* and *,/,*.

Endpoint	Trigger
DT	NN, NNP
PRP	NNS
JJ	JJR, RBR

Table 2: *Endpoints PoS and Trigger PoS of the barrier features employed in the assessment.*

All features we consider are binary, taking values 0 if the considered pattern does not occur, 1 otherwise. The entity feature vector is constructed by merging a different feature vector for each of the tokens composing the considered entities by a logical OR: the element of the final vector takes value 1 if the corresponding features takes value 1 in at least one of the all involved vectors. More precisely, let $E_i = t_{i,1}, ..., t_{i,k_i}$ and $E_j = t_{j,1}, ..., t_{j,k_j}$ be the two entities we are considering. To obtain the feature vector corresponding to this entity pair, we merge the feature vectors of $t_{i,1}, ..., t_{i,k_i}$ and $t_{j,1}, ..., t_{j,k_j}$ corresponding to both entities. Note that this representation is independent of the order of the two considered entities.

4 Experimental evaluation

The aim of the experimental assessment is twofold: to verify whether the system employing barrier features is competitive with state of the art systems, and to evaluate the role of barrier features in the results. Performance is evaluated by computing Precision (P), Recall (R) and F_1, as usual.

4.1 Data set

For experimental assessment we used the data set used by Roth and Yih (Roth and Yih, 2004), derived from TREC corpus[2], which is freely available. It includes three types of entities, namely PER (person), LOC (location) and ORG (organization) and the five types of binary relations reported in Table 3.

Relation	Example	agent	target
work for	employ-company	PER	ORG
kill	murderer-victim	PER	PER
live in	Clinton-USA	PER	LOC
located in	Rome - Italy	LOC	LOC
orgbased in	Harvad -U.S.	ORG	LOC

Table 3: List of relations with the type of the involved entities and the number of occurrences in the Roth Yih Data set.

The Roth and Yih data set is not divided in training and test set. Therefore assessment is performed by following the 5-fold cross validation protocol, as in (Giuliano et al., 2007; Roth and Yih, 2007; Kate and Mooney, 2010).

4.2 System tuning and kernel choice

The Roth and Yih data set comes along with the correct PoS tagging and therefore we consider gold case PoS's while constructing the features, as in (Giuliano et al., 2007). Then, the syntactic parse tree is automatically associated to each input sentence by using the Stanford Parser (Klein and Manning, 2003a; Klein and Manning, 2003b)[3], by employing the grammar for English distributed together with the parser. For the sake of precision, we mention that we do not give correct PoS's in input to the parser.

The classification was performed by using the SVM package SVMLight-TK[4] (Moschitti, 2006), which is based on SVMLight (Joachims, 1999), but also includes tree kernels, offering the possibility of combining tree-structured features with vectors, which is what we need to combine the input syntactic tree with the entity feature vector. In such approach, a further kernel can be introduced in addition to the tree kernel to apply to the unstructured features. To choose this second kernel, we compared the performance of different combinations including the tree kernel alone and in conjunction with other kernels.

As the data set has not been split in training and test sets, performance has been evaluated by fol-

[2]The annotated data are freely available at `http://l2r.cs.uiuc.edu/~cogcomp/Data/ER/conll04.corp`

[3]The parser can be freely downloaded from `http://nlp.stanford.edu/software/lex-parser.shtml`.

[4]The package is available from `http://dit.unitn.it/~moschitt/Tree-Kernel.htm`.

lowing a 5-fold cross-validation protocol, including 5 iterations with a different split in training and test sets at each step. The choice of the best kernel combination has been again based on a 5-fold cross-validation protocol, applied to the training set defined at each iteration. Significantly, the linear kernel showed the best performance for all split.

4.3 System performance

Assessment considers five classifiers, one for each relation. The data set is divided in subsets corresponding to the different relations. For each relation, training has been performed by considering gold positive examples for the considered relation while negative examples are represented by all the other pairs of entities having labels compatible with the relation. In this way, the number of negative examples is much larger than for positive examples. The SVM implementation we used allows to balance the number of positive and negative examples by a cost factor. We set it to the rate between the number of negative and positive examples. Table 4 reports the comparison between the performance of our system and the results presented in (Giuliano et al., 2007) for the $M_{O|K}$ system. With the only exception of the *located in* relation, our system has an F_1 larger than $M_{O|K}$ both on single relations and on average. Although we are not able to estimate the statistical significance of such comparison because we do not have the output of that system on each sentence, we think that this consistency is quite convincing. Note however that in two cases their precision is better than ours, and in three cases their recall is better. However, the average values are always better for our system. Although we are not reporting the exact results here, we noticed that the WordNet features (hypernyms of each entity tokens) do not give any significant improvement on performance.

4.4 Barrier feature contribution

Last but not least, we tried to understand the contribution of barrier feature to the global system performance. In order to obtain a numerical estimation, we run exactly the same experiment with and without barrier features. Results are reported in Table 4 and show that their contribution to performance is always relevant and on average can be evaluated in an improvement in F_1 of nearly the 15%.

5 Conclusions and future work

In this work we proposed a new kind of features for classifying semantic relations and showed how they are indeed effective in improving classification performance. Experimental assessment on the Roth and Yih data set not only shows that they performance are state of the art, but also that their contribution is relevant.

In the future, we plan to assess the barrier features on new data sets and on different tasks, such as relation extraction and entity classification. As the number of possible barrier features is very large, we plan to invest on the search for an effective smoothing strategy, in order to limit as much as possible the effect of data sparsity.

6 Aknowledgments

We are in debt with Giorgio Satta for indicating us the potentialities of barrier rules.

References

Brandon Beamer, Suma Bhat, Brant Chee, Andrew Fister, Alla Rozovskaya, and Roxana Girju. 2007. UIUC: A Knowledge-rich Approach to Identifying Semantic Relations between Nominals. In *Proc. of SemEval07: the Fourth International Workshop on Semantic Evaluations*, pages 386–389, Prague, Czech Republic, June. Association for Computational Linguistics.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proc. of ACL04: 42nd Meeting of the Association for Computational Linguistics , Main Volume*, pages 423–429, Barcelona, Spain, July.

Dmitry Davidov and Ari Rappoport. 2008. Classification of Semantic Relationships between Nominals Using Pattern Clusters. In *Proc. of ACL-08: HLT*, pages 227–235, Columbus, Ohio, June. Association for Computational Linguistics.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Claudio Giuliano, Alberto Lavelli, and Lorenza Romano. 2007. Relation extraction and the influence of automatic named-entity recognition. *ACM Trans. Speech Lang. Process.*, 5(1):1–26.

Zhou Guodong, Su Jian, Zhang Jie, and Zhang Min. 2005. Exploring various knowledge in relation extraction. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 427–434, Morristown, NJ, USA. Association for Computational Linguistics.

	OS No Barrier Features			OS With Barrier Features			$M_{O\|K}$		
Relation	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
kill	70.30	71.29	70.79	92.39	**75.63**	83.17	82.80	81.00	81.89
live in	73.26	63.65	68.12	74.69	**73.39**	74.33	78.00	65.80	71.38
work for	66.22	63.12	64.63	**76.38**	86.18	80.99	76.80	80.00	78.37
located in	61.53	72.23	71.88	**70.00**	**75.40**	**72.60**	79.60	76.00	77.76
orgabased in	68.13	66.33	67.22	86.58	77.70	81.90	74.30	77.20	75.72
average	69.89	67.32	68.53	80.01	77.66	78.54	78.30	76.00	77.02

Table 4: Comparison of performance of Our System (OS) with and without barrier features and the best performing one $M_{O|K}$ on the Roth and Yih data set.

Zhou GuoDong, Su Jian, and Zhang Min. 2006. Modeling commonality among related classes in relation extraction. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 121–128, Stroudsburg, PA, USA. Association for Computational Linguistics.

Thorsten Joachims. 1999. Making large-scale support vector machine learning practical. In *Advances in kernel methods: support vector learning*, pages 169–184. MIT Press, Cambridge, MA, USA.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *Proc. of ACL04: Companion Volume to the Proceedings of 42st Annual Meeting of the Association for Computational Linguistics*, pages 178–181, Barcelona, Spain, July. Association for Computational Linguistics.

Fred Karlsson, Atro Voutilainen, Juha Heikkila, and Arto Anttila, editors. 1995. *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text*. Mouton de Gruyter.

Rohit J. Kate and Raymond J. Mooney. 2010. Joint entity and relation extraction using card-pyramid parsing. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, CoNLL '10, pages 203–212, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dan Klein and Christopher D. Manning. 2003a. Accurate unlexicalized parsing. In *Proc. of ACL 03 of the 41st Annual Meeting on Association for Computational Linguistics*, pages 423–430, Morristown, NJ, USA. Association for Computational Linguistics.

Dan Klein and Christopher D. Manning. 2003b. Fast exact inference with a factored model for natural language parsing. In *Proc of NIPS03: In Advances in Neural Information Processing Systems*, pages 3–10. MIT Press.

Scott Miller, Heidi Fox, Lance Ramshaw, and Ralph Weischedel. 2000. A novel use of statistical parsing to extract information from text. In *Proc. of ANLP00: In 6th Applied Natural Language Processing Conference*, pages 226–233.

Alessandro Moschitti. 2006. Making tree kernels practical for natural language learning. In *EACL*.

Longhua Qian, Guodong Zhou, Fang Kong, Qiaoming Zhu, and Peide Qian. 2008. Exploiting constituent dependencies for tree kernel-based semantic relation extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 697–704, Stroudsburg, PA, USA. Association for Computational Linguistics.

Frank Reichartz, Hannes Korte, and Gerhard Paass. 2009. Dependency tree kernels for relation extraction from natural language text. In Wray Buntine, Marko Grobelnik, Dunja Mladenić, and John Shawe-Taylor, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 5782, chapter 18, pages 270–285. Springer Berlin Heidelberg, Berlin, Heidelberg.

Bryan Rink and Sanda Harabagiu. 2010. Utd: Classifying semantic relations by combining lexical and semantic resources. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 256–259, Uppsala, Sweden, July. Association for Computational Linguistics.

D. Roth and W. Yih. 2004. A Linear Programming Formulation for Global Inference in Natural Language Tasks. In *Proc. of CoNLL-2004*, pages 1–8. Boston, MA, USA.

D. Roth and W. Yih. 2007. Global inference for entity and relation identification via a linear programming formulation. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press.

Min Zhang, Jian Su, Danmei Wang, Guodong Zhou, and Chew Lim Tan. 2005. Discovering relations between named entities from a large raw corpus using tree similarity-based clustering. In *Proc. of IJCNLP05: International Joint Conference on Natural Language Processing*, pages 378–389.