RTV: Tree Kernels for Thematic Role Classification

Daniele Pighin
FBK-irst; University of Trento, DIT
pighin@itc.it

Alessandro Moschitti
University of Trento, DIT
moschitti@dit.unitn.it

Roberto Basili
University of Rome *Tor Vergata*, DISP
basili@info.uniroma2.it

Abstract

We present a simple, two-steps supervised strategy for the identification and classification of thematic roles in natural language texts. We employ no external source of information but automatic parse trees of the input sentences. We use a few attribute-value features and tree kernel functions applied to specialized structured features. The resulting system has an F_1 of 75.44 on the SemEval2007 closed task on semantic role labeling.

1 Introduction

In this paper we present a system for the labeling of semantic roles that produces VerbNet (Kipper et al., 2000) like annotations of free text sentences using only full syntactic parses of the input sentences. The labeling process is modeled as a cascade of two distinct classification steps: (1) boundary detection (BD), in which the word sequences that encode a thematic role for a given predicate are recognized, and (2) role classification (RC), in which the type of thematic role with respect to the predicate is assigned. After role classification, a set of simple heuristics are applied in order to ensure that only well formed annotations are output.

We designed our system on a per-predicate basis, training one boundary classifier and a battery of role classifiers for each predicate word. We clustered all the senses of the same verb together and ended up with 50 distinct boundary classifiers (one for each target predicate word) and 619 role classifiers to recognize the 47 distinct role labels that appear in the training set.

The remainder of this paper is structured as follows: Section 2 describes in some detail the architecture of our labeling system; Section 3 describes the features that we use to represent the classifier examples; Section 4 describes the experimental setting and reports the accuracy of the system on the SemEval2007 semantic role labeling closed task; finally, Section 5 discusses the results and presents our conclusions.

2 System Description

Given a target predicate word in a natural language sentence, a SRL system is meant to correctly identify all the arguments of the predicate. This problem is usually divided in two sub-tasks: (a) the detection of the boundaries (i. e. the word span) of each argument and (b) the classification of the argument type, e.g. *Arg0* or *ArgM* in PropBank or *Agent* and *Goal* in FrameNet or VerbNet.

The standard approach to learn both the detection and the classification of predicate arguments is summarized by the following steps:

1 Given a sentence from the *training-set*, generate a full syntactic parse-tree;

2 let $\mathcal{P}$ and $\mathcal{A}$ be the set of predicates and the set of parse-tree nodes (i.e. the potential arguments), respectively;

3 for each pair $\langle p, a\rangle \in \mathcal{P} \times \mathcal{A}$:

3.1 extract the feature representation set, $F_{p,a}$;

3.2 if the sub-tree rooted in a covers exactly the words of one argument of p, put $F_{p,a}$ in T^+ (positive examples), otherwise put it in T^- (negative examples).

For instance, in Figure 1.a, for each combination of the predicate *approve* with any other tree node a

Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), pages 288–291,
Prague, June 2007. ©2007 Association for Computational Linguistics

that does not overlap with the predicate, a classifier example $F_{\text{approve},a}$ is generated. If a exactly covers one of the predicate arguments (in this case: "The charter", "by the EC Commission" or "on Sept. 21") it is regarded as a positive instance, otherwise it will be a negative one, e. g. $F_{\text{approve},(\text{NN charter})}$.

The T^+ and T^- sets are used to train the boundary classifier. To train the role multi-class classifier, T^+ can be reorganized as positive $T^+_{\text{arg}_i}$ and negative $T^-_{\text{arg}_i}$ examples for each argument i. In this way, an individual ONE-vs-ALL classifier for each argument i can be trained. We adopted this solution, according to (Pradhan et al., 2005), since it is simple and effective. In the classification phase, given an unseen sentence, all its $F_{p,a}$ are generated and classified by each individual role classifier. The role label associated with the maximum among the scores provided by the individual classifiers is eventually selected.

To make the annotations consistent with the underlying linguistic model, we employ a few simple heuristics to resolve the overlap situations that may occur, e. g. both "charter" and "the charter" in Figure 1 may be assigned a role:

- if more than two nodes are involved, i. e. a node d and two or more of its descendants n_i are classified as arguments, then assume that d is not an argument. This choice is justified by previous studies (Moschitti et al., 2006b) showing that the accuracy of classification is higher for lower nodes;
- if only two nodes are involved, i. e. they dominate each other, then keep the one with the highest classification score.

3 Features for Semantic Role Labeling

We explicitly represent as attribute-value pairs the following features of each $F_{p,a}$ pair:

- *Phrase Type*, *Predicate Word*, *Head Word*, *Position* and *Voice* as defined in (Gildea and Jurasfky, 2002);
- *Partial Path*, *No Direction Path*, *Head Word POS*, *First and Last Word/POS in Constituent* and *SubCategorization* as proposed in (Pradhan et al., 2005);

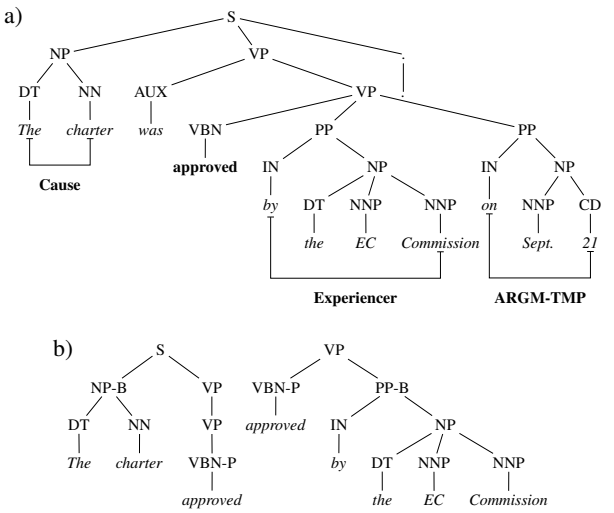

Figure 1: A sentence parse tree (a) and two example AST_1^m structures relative to the predicate *approve* (b).

Set	Props	T	T^+	T^-
Train	15,838	793,104	45,157	747,947
Dev	1,606	75,302	4,291	71,011
Train - Dev	14,232	717,802	40,866	676,936

Table 1: Composition of the dataset in terms of: number of annotations (Props); number of candidate argument nodes (T); positive (T^+) and negative (T^-) boundary classifier examples.

- *Syntactic Frame* as designed in (Xue and Palmer, 2004).

We also employ structured features derived by the full parses in an attempt to capture relevant aspects that may not be emphasized by the explicit feature representation. (Moschitti et al., 2006a) and (Moschitti et al., 2006b) defined several classes of structured features that were successfully employed with tree kernels for the different stages of an SRL process. Figure 1 shows an example of the AST_1^m structures that we used for both the boundary detection and the role classification stages.

4 Experiments

In this section we discuss the setup and the results of the experiments carried out on the dataset of the SemEval2007 closed task on SRL.

Task	Kernel(s)	Precision	Recall	$F_{\beta=1}$
BD	poly	94.34%	71.26%	81.19
	poly + TK	92.89%	76.09%	83.65
BD + RC	poly	88.72%	68.76%	77.47
	poly + TK	86.60%	72.40%	78.86

Table 2: SRL accuracy on the development test for the boundary detection (BD) and the complete SRL task (BD+RC) using the polynomial kernel alone (poly) or combined with a tree kernel function (poly + TK).

4.1 Setup

The training set comprises 15,838[1] training annotations organized on a per-verb basis. In order to build a development set (Dev), we sampled about one tenth, i. e. 1,606 annotations, of the original training set. For the final evaluation on the test set (Test), consisting of 3,094 annotations, we trained our classifiers on the whole training data. Statistics on the dataset composition are shown in Table 1.

The evaluations were carried out with the SVM-Light-TK[2] software (Moschitti, 2004) which extends the SVM-Light package (Joachims, 1999) with tree kernel functions. We used the default polynomial kernel (degree=3) for the linear features and a SubSet Tree (SST) kernel (Collins and Duffy, 2002) for the comparison of AST_1^m structured features. The kernels are normalized and summed by assigning a weight of 0.3 to the TK contribution.

Training all the 50 boundary classifiers and the 619 role classifiers on the whole dataset took about 4 hours on a 64 bits machine (2.2GHz, 1GB RAM)[3].

4.2 Evaluation

All the evaluations were carried out using the CoNLL2005 evaluator tool available at `http://www.lsi.upc.es/~srlconll/soft.html`.

Table 2 shows the aggregate results on boundary detection (BD) and the complete SRL task (BD+RC) on the development set using the polynomial kernel alone (poly) or in conjunction with the tree kernels and structured features (poly+TK). For both tasks, tree kernel functions do trigger automatic feature selection and improve the polynomial kernel by 2.46 and 1.39 F_1 points, respectively.

The SRL accuracy for each one of the 47 distinct role labels is shown in Table 3. Column 2 lists

Role	#TI	Precision	Recall	$F_{\beta=1}$
Ov(BD)	6931	87.09%	72.96%	79.40
Ov(BD+RC)		81.58%	70.16%	75.44
ARG2	4	100.00%	25.00%	40.00
ARG3	17	61.11%	64.71%	62.86
ARG4	4	0.00%	0.00%	0.00
ARGM-ADV	188	55.14%	31.38%	40.00
ARGM-CAU	13	50.00%	23.08%	31.58
ARGM-DIR	4	100.00%	25.00%	40.00
ARGM-EXT	3	0.00%	0.00%	0.00
ARGM-LOC	151	51.66%	51.66%	51.66
ARGM-MNR	85	41.94%	15.29%	22.41
ARGM-PNC	28	38.46%	17.86%	24.39
ARGM-PRD	9	83.33%	55.56%	66.67
ARGM-REC	1	0.00%	0.00%	0.00
ARGM-TMP	386	55.65%	35.75%	43.53
Actor1	12	85.71%	50.00%	63.16
Actor2	1	100.00%	100.00%	100.00
Agent	2551	91.38%	77.34%	83.78
Asset	21	42.42%	66.67%	51.85
Attribute	17	60.00%	70.59%	64.86
Beneficiary	24	65.00%	54.17%	59.09
Cause	48	75.56%	70.83%	73.12
Experiencer	132	86.49%	72.73%	79.01
Location	12	83.33%	41.67%	55.56
Material	7	100.00%	14.29%	25.00
Patient	37	76.67%	62.16%	68.66
Patient1	20	72.73%	40.00%	51.61
Predicate	181	63.75%	56.35%	59.82
Product	106	70.79%	59.43%	64.62
R-ARGM-LOC	2	0.00%	0.00%	0.00
R-ARGM-MNR	2	0.00%	0.00%	0.00
R-ARGM-TMP	4	0.00%	0.00%	0.00
R-Agent	74	70.15%	63.51%	66.67
R-Experiencer	5	100.00%	20.00%	33.33
R-Patient	2	0.00%	0.00%	0.00
R-Predicate	1	0.00%	0.00%	0.00
R-Product	2	0.00%	0.00%	0.00
R-Recipient	8	100.00%	87.50%	93.33
R-Theme	7	75.00%	42.86%	54.55
R-Theme1	7	100.00%	85.71%	92.31
R-Theme2	1	50.00%	100.00%	66.67
R-Topic	14	66.67%	42.86%	52.17
Recipient	48	75.51%	77.08%	76.29
Source	25	65.22%	60.00%	62.50
Stimulus	21	33.33%	19.05%	24.24
Theme	650	79.22%	68.62%	73.54
Theme1	69	77.42%	69.57%	73.28
Theme2	60	74.55%	68.33%	71.30
Topic	1867	84.26%	82.27%	83.25

Table 3: Evaluation of the semantic role labeling accuracy on the SemEval2007 - Task 17 test set using the poly + TK kernel. Column *#TI* reports the number of instances of each role label in the test set. Rows *Ov(BD)* and *Ov(BD + RC)* show the overall accuracy on the boundary detection and the complete SRL task, respectively.

[1] A bunch of unaligned annotations were removed from the dataset.

[2] `http://ai-nlp.info.uniroma2.it/moschitti/`

[3] In order to have a faster development cycle, we only used 60k training examples to train the boundary classifier of the verb *say*. The accuracy on this relation is still very high, as we measured an overall F_1 of 87.18 on the development set and of 85.13 on the test set.

the number of instances of each role in the test set. Many roles have very few positive examples both in the training and the test sets, and therefore have little or no impact on the overall accuracy which is dominated by the few roles which are very frequent, such as *Theme*, *Agent*, *Topic* and *ARGM-TMP* which account for almost 80% of all the test roles.

5 Final Remarks

In this paper we presented a system that employs tree kernels and a basic set of flat features for the classification of thematic roles.

We adopted a very simple approach that is meant to be as general and fast as possible. The issue of generality is addressed by training the boundary and role classifiers on a per-predicate basis and by employing tree kernel and structured features in the learning algorithm. The resulting architecture can indeed be used to learn the classification of roles of non-verbal predicates as well, and the automatic feature selection triggered by the tree kernel should compensate for the lack of *ad-hoc*, well established explicit features for some classes of non-verbal predicates, e. g. adverbs or prepositions.

Splitting the learning problem also has the clear advantage of noticeably improving the efficiency of the classifiers, thus reducing training and classification time. On the other hand, this split results in some classifiers having too few training instances and therefore being very inaccurate. This is especially true for the boundary classifiers, which conversely need to be very accurate in order to positively support the following stages of the SRL process. The solution of a monolithic boundary classifier that we previously employed (Moschitti et al., 2006b) is noticeably more accurate though much less efficient, especially for training. Indeed, after the SemEval2007 evaluation period was over, we ran another experiment using a monolithic boundary classifier. On the test set, we measured F1 values of 82.09 vs 79.40 and 77.17 vs 75.44 for the boundary detection and the complete SRL tasks, respectively.

Although it was provided as part of both the training and test data, we chose not to use the verb sense information. This choice is motivated by our intention to depend on as less external resources as possible in order to be able to port our SRL system to other linguistic models and languages, for which such resources may not exist. Still, identifying the predicate sense is a key issue especially for role classification, as the argument structure of a predicate is largely determined by its sense. In the near feature we plan to use larger structured features, i. e. spanning all the potential arguments of a predicate, to improve the accuracy of our role classifiers.

Acknowledgments

The development of the SRL system was carried out at the University of Rome *Tor Vergata* and financed by the EU project PrestoSpace[4] (FP6-507336).

References

Michael Collins and Nigel Duffy. 2002. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *ACL02*.

Daniel Gildea and Daniel Jurasfky. 2002. Automatic labeling of semantic roles. *Computational Linguistic*, 28(3):496–530.

T. Joachims. 1999. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*.

Karin Kipper, Hoa Trang Dang, and Martha Palmer. 2000. Class-based construction of a verb lexicon. In *Proceedings of AAAI-2000 Seventeenth National Conference on Artificial Intelligence, Austin, TX*.

Alessandro Moschitti, Daniele Pighin, and Roberto Basili. 2006a. Semantic role labeling via tree kernel joint inference. In *Proceedings of the Tenth Conference on Computational Natural Language Learning, CoNLL-X*.

Alessandro Moschitti, Daniele Pighin, and Roberto Basili. 2006b. Tree kernel engineering in semantic role labeling systems. In *Proceedings of the Workshop on Learning Structured Information in Natural Language Applications, EACL 2006*, pages 49–56, Trento, Italy, April. European Chapter of the Association for Computational Linguistics.

Alessandro Moschitti. 2004. A study on convolution kernel for shallow semantic parsing. In *proceedings of ACL-2004*, Barcelona, Spain.

Sameer Pradhan, Kadri Hacioglu, Valeri Krugler, Wayne Ward, James H. Martin, and Daniel Jurafsky. 2005. Support vector learning for semantic argument classification. *to appear in Machine Learning Journal*.

Nianwen Xue and Martha Palmer. 2004. Calibrating features for semantic role labeling. In *Proceedings of EMNLP 2004*, pages 88–94, Barcelona, Spain, July.

[4] http://www.prestospace.org