Shallow Semantic Parsing Based on FrameNet, VerbNet and PropBank

Ana-Maria Giuglea [1] and **Alessandro Moschitti** [2]

Abstract. This article describes a semantic parser based on FrameNet semantic roles that uses a broad knowledge base created by interconnecting three major resources: FrameNet, VerbNet and PropBank. We link the above resources through a mapping between Intersective Levin classes, which are part of PropBank's annotation, and the FrameNet frames. By using Levin classes, we successfully detect FrameNet semantic roles without relying on the frame information. At the same time, the combined usage of the above resources increases the verb coverage and confers more robustness to our parser. The experiments with Support Vector Machines on automatic Levin class detection suggest that (a) tree kernels are well suited for the task and (b) Intersective Levin classes can be used to improve the accuracy of semantic parsing based on FrameNet roles.

1 INTRODUCTION

Knowing the semantic roles played by the entities that appear in a sentence is of major importance for understanding its underlying meaning. The inherent word ambiguity can lead to very different readings of the same sentence. One important step towards deciding the correct reading is to find the sense of the verb in the sentence.

The previous point was highlighted by the results, obtained with and without the frame information during the Senseval-3 competition on FrameNet [10] role labeling task [17]. When such information was not used by the systems, the performance drop was more than 10 percent points. This is quite intuitive as the semantics of many roles strongly depend on the focused frame. Thus, we cannot expect a good performance when this information is not available.

A solution to this problem is the automatic frame detection. Unfortunately, our preliminary experiments showed that given a FrameNet (FN) predicate-argument structure, the task of identifying the associated frame can be performed with very good results when the verb predicates have enough training examples, but becomes very challenging otherwise. The predicates not yet included in FN, e.g. belonging to new application domains, are especially problematic since there is no training data available. In such cases the frame classifier reaches very low accuracy (under 50%).

We have thus studied new means of capturing the semantic context, other than the frame, which can be easily annotated on FrameNet and are available on a larger scale (i.e. have a better coverage). A very good candidate seems to be the Intersective Levin class information [2] that can be found as well in other predicate resources like PropBank and VerbNet.

In this paper:

1. we employ SVM Tree Kernels and structural features [19] for automatic token-based verb classification[3] of the Intersective Levin classes (ILCs);
2. we use the classifiers trained in step 1 to automatically annotate FrameNet with Levin class information which is needed for the semantic role labeling (SRL) task; and
3. we test the effectiveness of the ILC information in the FrameNet SRL task.

On FrameNet, we obtain the gold Levin class annotation through a mapping between Intersective Levin classes and FrameNet frames [7]. This mapping employs also the PropBank corpus [11] and the VerbNet lexicon [12].

In the remainder of this paper Section 2 and 3 introduce the resources used by our study, namely VerbNet, PropBank and FrameNet, Section 4 summarizes previous work on Levin verb sense disambiguation, Section 5 focuses on Tree Kernels and the features used while Section 6 contains a description of the performed experiments. Finally, Section 7 presents our results and conclusions.

2 LEVIN AND PROPBANK

Levin clusters [15] are formed according to diathesis alternation criteria which are variations in the way verbal-arguments are grammatically expressed when a specific semantic phenomenon arises. For example, two different types of diathesis alternations are the following:

(a) Middle Alternation
[$_{Subject,\ Agent}$ The butcher] cuts [$_{Direct\ Object,\ Patient}$ **the meat**].
[$_{Subject,\ Patient}$ **The meat**] cuts easily.

(b) Causative/inchoative Alternation
[$_{Subject,\ Agent}$ Janet] broke [$_{Direct\ Object,\ Patient}$ **the cup**].
[$_{Subject,\ Patient}$ **The cup**] broke.

In both cases, what is alternating is the grammatical function that the Patient role takes when changing from the transitive use of the verb to the intransitive one. The semantic phenomenon accompanying these types of alternations is the change of focus from the entity performing the action to the theme of the event.

Levin documented 79 alternations which constitute the building blocks for the verb classes. Although alternations are chosen as the primary means for identifying the classes, additional properties related to subcategorization, morphology and extended meanings of

[1] University of Rome "Tor Vergata": agiuglea@gmail.com
[2] University of Rome "Tor Vergata": moschitti@info.uniroma2.it

[3] The best semantic class is determined for the verb token given the local context of the phrase rather than using the set of verb occurrences across a corpus or a document (i.e. type-based classification).

verbs are taken into account as well. Thus, from a syntactic point of view, the verbs in one Levin class have a regular behavior, different from the verbs pertaining to other classes. Also, the classes are semantically coherent and all verbs belonging to one class share the same participant roles.

This constraint of having the same semantic roles is further ensured inside the VerbNet lexicon which is constructed based on a more refined version of the Levin classification called Intersective Levin classes [2]. The lexicon provides a regular association between the syntactic and semantic properties of each of the described classes. It also provides information about the syntactic frames (alternations) in which the verbs participate together with the set of possible semantic roles.

One corpus associated with the VerbNet lexicon is PropBank. The annotation scheme of PropBank ensures that the verbs belonging to the same Levin class and exhibiting the same diathesis alternations share similarly-labeled arguments. Inside one Levin class to one argument corresponds one semantic role numbered sequentially from Arg0 to Arg5. Higher numbered argument labels are less consistent and assigned per verb basis.

Besides semantic roles, PropBank is annotated also with Intersective Levin class information and contains gold parse trees. These features made PropBank very suitable for testing the effectiveness of our Tree Kernel approach for ILC detection. We were able to measure the accuracy of our machine learning algorithm in the presence of gold predicate-argument structures, which gave us a performance upper bound.

During the second step (i.e. annotating FrameNet with ILC), in order to train ILC classifiers also on FrameNet we need gold Intersective Levin class annotations. To achieve that we employed a semi-automatic algorithm that mapped FrameNet frames to Levin classes, thus assigning gold ILC to FrameNet. More details about FrameNet and the mapping algorithm are presented in the next section.

3 LEVIN AND FRAMENET

One of the goals of the FrameNet project is to design a hierarchical linguistic ontology that can be used for automatic processing of semantic information. This hierarchy contains an extensive semantic analysis of verbs, nouns, adjectives and situations in which they are used, called frames. The basic assumption on which the frames are built is that each word evokes a particular situation with specific participants [5]. The situations depict the entities involved and the roles they play. The word that evokes a particular *frame* is called *target word* or predicate and can be an adjective, noun or verb. The participant entities are defined using semantic roles and they are called *frame elements*.

Predicates belonging to the same FrameNet frame were proven [6] to have a coherent syntactic behavior that is also different from predicates pertaining to other frames. This finding is consistent with the assumption on which Levin's verb classification is build. This insight determined us to study the relation between FrameNet frames and Levin classes.

The Levin classes were constructed based on regularities exhibited at grammatical level and the resulting clusters were shown to be semantically coherent. As opposed, the FrameNet frames were build on semantic bases, by putting together verbs, nouns and adjectives that evoke the same situations. Although different in conception, the FrameNet verb clusters and VerbNet verb clusters have common properties:

1. Different syntactic properties between distinct verb clusters (as proven by the experiments in [6])
2. Shared sets of possible semantic roles for all verbs pertaining to the same cluster.

Having these insights, we have assigned a correspondent VerbNet class not to each verb predicate but rather to each frame. In doing this we have applied the simplifying assumption that a frame has a unique corresponding Levin class. Thus, we have created a one-to-many mapping between the Levin classes and the frames.

The mapping algorithm consists of three steps: (a) we link the frames and Intersective Levin verb classes that have the largest number of verbs in common and we create a set of pairs ⟨FN frame, VN class⟩ (see Table 1); (b) we refine the pairs obtained in the previous step based on diathesis alternation criteria, i.e. the verbs pertaining to the FN frame have to undergo the same diathesis alternation that characterize the corresponding VN class and (c) we manually check the resulting mapping.

INPUT
$VN = \{C | C\ is\ a\ VerbNet\ class\}$
$VN\ Class\ C = \{v | c\ is\ a\ verb\ of\ C\}$
$FN = \{F | F\ is\ a\ FrameNet\ frame\}$
$FN\ frame\ F = \{v | v\ is\ a\ verb\ of\ F\}$
OUTPUT
$Pairs = \{\langle F, C\rangle\, | F \in FN, C \in VN : F\ maps\ to\ C\}$
COMPUTE PAIRS:
$Let\ Pairs = \emptyset$
$for\ each\ F \in FN$
$\quad (I)\ compute\ C^* = arg\max_{C \in VN}\ |F \cap C|$
$\quad (II)\ if\ \ |F \cap C^*| \geq 3\ then\ Pairs = Pairs \cup \langle F, C^*\rangle$

Table 1. Linking FrameNet frames and VerbNet classes.

During the second step of the mapping we make use of the property (2) of the Levin classes and FN frames presented in this section. According to this property, all verbs pertaining to one frame or Levin class have the same participant roles. Thus, a first test of compatibility between a frame and a Levin class is that they share the same participant roles. As FN is annotated with frame-specific semantic roles, we manually mapped these roles into the VN set of thematic roles. Given a frame, we assigned thematic roles to all frame elements that are associated with verbal predicates. For example the *Speaker*, *Addressee*, *Message* and *Topic* roles from the *Telling* frame were respectively mapped into the *Agent*, *Recipient*, *Theme* and *Topic* theta roles.

After the role matching, the mapping algorithm checks both the syntactic and semantic consistency by comparing the role frequency distributions on different syntactic positions for the two candidates. More details are given in [7]. We mention that the algorithm identifies correctly the cases for which our simplifying assumption does not hold, having an overall accuracy of 89.6%.

Having gold ILC annotation on FrameNet allows us to train Intersective Levin class classifiers also on this corpus. In this way, we can extend the verb coverage to encompass both PropBank and FrameNet.

In the next sections we describe our approach on ILC automatic detection and also some of the literature work on this subject.

4 PREVIOUS WORK ON LEVIN CLASS DETECTION

Levin's verb classification is based on straightforward syntactic criteria which makes it especially appealing for automation. As a consequence, it was used in many different studies ranging from machine translation [3] and information retrieval [16] to automatic acquisition of lexical semantic information and creation of dictionaries [4].

Regarding automatic verb classification, most of the previous studies focused on type-based classification. We mention Merlo and Stevenson [18, 21] who use grammatical features to classify verbs into three classes: unergative, unaccusative and object-drop. These classes comprise several Levin classes and were chosen because they participate in similar alternations (i.e. transitive alternations) but they assign different thematic roles to their arguments.

In their study, Merlo and Stevenson investigate to what extent features extracted from the predicate argument structures are useful for disambiguating among unergative, unaccusative and object-drop. Some of the most successful features used were Causativity and Transitivity. Causativity feature is a marker for verbs participating in causative alternations (e.g. the sentences of th example (b) of Section 2) and measures how many times the same noun occurs as subject and as object of the verb (i.e. the degree of overlap). Transitivity is a binary feature that marks whether the verb is used in a transitive or intransitive form. We will show in the next section that structural features capture both the causative and transitive markers.

Other studies use subcategorization information and selectional restrictions to cluster verbs into Levin compatible classes [8]. The resulting clusters are measured against Levin's classification having a 61% match. Although it is counterintuitive, the selectional preferences for the arguments in the subcategorization frames seem to have a negative impact on performance.

The subcategorization feature is used also in Lapata and Brew's studies [14, 13] which focus on dative and benefactive alternations. They view the choice for a class as being estimated by the joint probability $P(verb, syntacticframe, class)$ and they design a distributional model that uses the BNC corpus. The results obtained are very promising considering that only subcategorization information is used (74.6% accuracy for genuinely ambiguous verbs). Also, Lapata and Brew incorporate their distributional model as prior knowledge in a naive Bayes classifier for performing token-level disambiguation. Unfortunately, their results are reported per syntactic frame which makes them hard to compare with ours.

Overall, the previous studies are restricted in scope by focusing only on specific aspects or alternations involved in the Levin's verb classification. One novelty of our approach is the fact that we use corpora that are annotated with Levin class information. Thus, we are able to appreciate better to what extent our method is applicable for Levin class disambiguation in general. Our analysis is conducted on a number of 179 classes that had training examples in the corpora.

In the following section we will present the algorithm and the features used by our model. We will show that structural features include some of the best features developed in the literature like causativity, transitivity and also, the very important subcategorization information.

5 Tree Kernels and Feature Space

The main idea of tree kernels is the modeling of a $K_T(T_1, T_2)$ function which computes the number of common substructures between two trees T_1 and T_2. Thus, we can use, as features, structures drawn directly from the syntactic parse tree of the sentence.

The kernel that we employed in our experiments was devised in [19]. We used a reduced version of the predicate-argument structure (Figure 1) which contains also the headwords of the arguments, useful for representing the selectional preferences. This feature is a variant of the SCF feature from [19], but in a reduced format (hereafter called SCF-reduced). In the following figure we present an example of the SCF-reduced feature constructed for the sentences of the example (b) of Section 2.

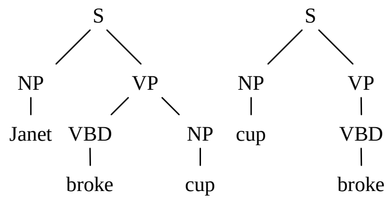

Figure 1. SCF-reduced of the sentences from the example (b) of Section 2

The trees in Figure 1 have respectively 17 and 10 substructures from which 3 are shared (Figure 2).

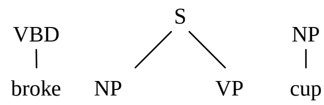

Figure 2. Common substructures of the sentences of Figure 1.

We note that the overlap between "cup" used on the subject position and "cup" used on the object position increases the similarity measure between the two sentences of the example (b). Thus, the Causativity feature from [18] is subsumed by the SCF-reduced feature. Other substructures contain the subcategorization frame with or without the verb marked (Figure 3). As in general the subcategorization frame embeds also the transitivity or intransitivity marker, we conclude that both the Transitivity feature [18] and Subcategorization feature [14, 8] are subsumed by SCF-reduced.

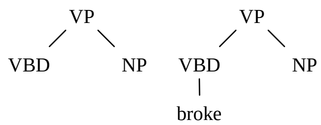

Figure 3. Substructures embedding the Subcategorization feature.

The next section shows that SCF-reduced detects ILCs with high accuracy and is very robust with respect to automatic parse trees and different corpora.

6 EXPERIMENTS

The aim of this research is to show that the ILC feature is very effective for the FrameNet semantic role labeling (SRL) task. For this purpose we carried out several tests.

During our first experiment set we trained (1) an ILC multiclassifier from FN, (2) an ILC multiclassifier from PB and (3) a frame multiclassifier from FN. We compared the results obtained when trying to classify ILCs with the results obtained when classifying frame. We show that ILCs are easier to detect than FN frames.

	run-51.3.2	cooking-45.3	characterize-29.2	other_cos-45.4	say-37.7	correspond-36.1	Multiclassifier
PB #Train Instances	262	6	2,945	2,207	9,707	259	52,172
PB #Test Instances	5	5	134	149	608	20	2,742
PB Results	**75**	**33.33**	**96.3**	**97.24**	**100**	**88.89**	**92.96**
FN #Train Instances	5,381	138	765	721	1,860	557	46,734
FN #Test Instances	1,343	35	40	184	1,343	111	11,650
FN Results	**96.36**	**72.73**	**95.73**	**92.43**	**94.43**	**78.23**	**92.63**

Table 2. Argument classifier F1s and the overall multiclassifier accuracy for ILC labeling.

	Body_part	Crime	Degree	Agent	Multiclassifier
FN #Train Instances	1,511	39	765	6,441	102,724
FN #Test Instances	356	5	187	1,643	25,615
LF+Gold Frame	90.91	88.89	70.51	93.87	90.8
LF+Gold ILC	90.80	88.89	71.52	92.01	88.23
LF+Automatic Frame	**84.87**	**88.89**	**70.10**	**87.73**	**85.64**
LF+Automatic ILC	**85.08**	**88.89**	**69.62**	**87.74**	**84.45**
LF	79.76	75.00	64.17	80.82	80.99

Table 3. Argument classifier F1s and the overall multiclassifier accuracy for FN semantic role labeling.

Our second set of experiments regards the automatic labeling of FN semantic roles on FN corpus when using as features: gold frame, gold ILC, automatically detected frame and automatically detected ILC. We show that in all situations in which the ILC feature is used, the accuracy loss, compared to the usage of the frame feature, is negligible. We thus show that the ILC can successfully replace the frame feature for the task of semantic role labeling.

Another set of experiments regards the generalization property of the ILC. We show the impact of this feature on SRL when very few training data is available and its evolution when adding more and more training examples. We again perform the experiments for: gold frame, gold ILC, automatically detected frame and automatically detected ILC.

Finally, we simulate the difficulty of free text by annotating PB with FN semantic roles. We used PB because it is different from FN from a domain point of view. This characteristic makes PB a complex test bed for semantic role models trained on FN. In the following section we present the results obtained for each of the experiments mentioned above.

6.1 Experimental setup

The corpora available for the experiments were PB and FN. PB contains about 54,900 sentences and gold parse trees. We used sections from 02 to 22 (52,172 sentences) to train the ILC classifiers and section 23 (2,742 sentences) for testing purposes.

For the experiments on FN corpus, we extracted 58,384 sentences from the 319 frames that contain at least one verb annotation. There are 128,339 argument instances of 454 semantic roles. Only verbs are selected to be predicates in our evaluations. Moreover, as there is no fixed split between training and testing, we randomly selected 20% of sentences for testing and 80% for training. The sentences were processed using Charniak's parser [1] to generate parse trees automatically.

The classification models were implemented by means of the SVM-light-TK software available at `http://ai-nlp.info.uniroma2.it/moschitti` which encodes tree kernels in the SVM-light software [9]. We used the default parameters. The classification performance was evaluated using the F_1 measure for the single-argument classifiers and the accuracy for the multiclassifiers.

6.2 Automatic VerbNet class vs. automatic FrameNet frame detection

In these experiments, we classify the ILCs on PB and FN and the frames on FN. For the training stage we use SVMs with Tree Kernels.

For ILC detection the results are depicted in Table 2. The first six columns report the F_1 measure of some verb class classifiers whereas the last column shows the global multiclassifier accuracy. We note the ILC results on PB are similar to those obtained for the ILCs on FN. This suggests that the training corpus does not have a major influence. Also, the SCF-based tree kernel seems to be robust with respect to the quality of the parse trees. The performance decay is very small on FN that uses automatic parse trees with respect to PB that contains gold parse trees.

For frame detection on FN, we trained our classifier on 46,734 training instances and tested on 11,650 testing instances, obtaining an accuracy of 91.11%. Consequently, the ILC detection on both PB (92.96%) and FN (92.63%) is more accurate than the frame detection.

6.3 Automatic semantic role labeling on FrameNet

In the experiments involving semantic role labeling, we used SVMs with polynomial kernels. We adopted the standard features developed for semantic role detection by Gildea and Jurafsky : *Predicate*, *Headword*, *Phrase Type*, *Governing Category*, *Position*, *Voice* and *Path*. Also, we considered some of the features designed by [20]: *First and Last Word/POS in Constituent*, *Subcategorization*, *Head Word of Prepositional Phrases* and the *Syntactic Frame* feature from [22]. For the rest of the paper, we will refer to these features as being literature features (LF). The results obtained when using the literature features alone or in conjunction with the gold frame feature, gold ILC, automatically detected frame feature and automatically detected ILC are depicted in Table 3.

The first four columns report the F_1 measure of some role classifiers whereas the last column shows the global multiclassifier accuracy. The first row contains the number of training and testing instances and each of the other rows contains the performance obtained for different feature combinations. The results are reported for the labeling task as the argument-boundary detection task is not affected by the frame-like features [6].

We note that automatic frame produces an accuracy very close to the one obtained with automatic ILCs suggesting that these latter are very good candidate for replacing the frame feature. Also, both automatic features are very effective, decreasing the error rate of 20%.

To test the impact of the ILC feature on SRL with different amount of training data, we additionally draw the learning curves with respect to different features: LF, LF+ gold ILC , LF+automatic ILC trained on PB and LF+automatic ILC trained on FN. As can be noted, the automatic ILC information provided by the ILC classifiers (trained on FN or PB) performs almost as good as the gold ILC (Figure 4).

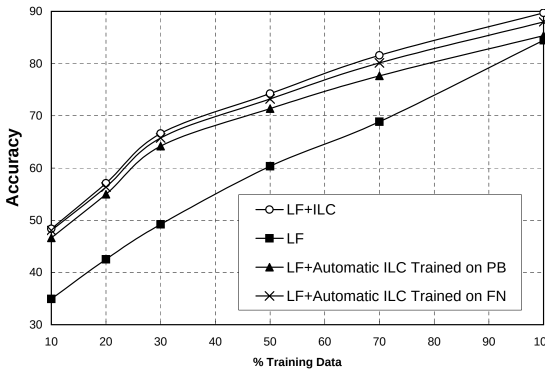

Figure 4. Semantic role learning curve.

6.4 Annotating PB with FN semantic roles

To show that our approach can be suitable for semantic role free-text annotation, we have automatically classified PB sentences with the FN semantic-role classifiers. In order to measure the quality of the annotation, we randomly selected 100 sentences and manually verified them. We measured the performance obtained with and without the automatic ILC feature. The sentences contained 189 arguments from which 35 were incorrect when ILC was used compared to 72 incorrect in the absence of this feature, i.e. an accuracy of 81% with ILC versus 62% without it. This demonstrates the importance of the ILC feature outside the scope of FrameNet where the frame feature is not available.

7 CONCLUSIONS

In this paper, we pursue Levin's thesis and we automatically classify verbs based on their predicate-argument structure and their selectional preferences on different argument slots. We show that Tree Kernels and structural features are more suitable for our goal compared with other methods that use for example linear features. Also, by comparing our structural features with previously developed and linguistically motivated features we found a reasonably degree of overlap. Additionally, we demonstrate that our approach is very robust and can be applied successfully on any type of parse trees (gold or automatic) or corpora.

Regarding the impact of ILC feature on SRL, we have shown that it can replace the frame feature without appreciable accuracy decay. This is very important as (1) several frames are not covered by enough training data and (2) thanks to our mapping algorithm we can reuse the data from PropBank to extend the verb coverage.

REFERENCES

[1] Eugene Charniak, 'A maximum-entropy-inspired parser', in *NACL00*, Seattle, Washington, (2000).

[2] Hoa Trang Dang, Karin Kipper, Martha Palmer, and Joseph Rosenzweig, 'Investigating regular sense extensions based on intersective levin classes', in *COLING-ACL98*, (1998).

[3] Bonnie Dorr, 'Large-scale dictionary construction for foreign language tutoring and interlingual machine translation', *Machine Translation*, (1997).

[4] Bonnie Dorr and Doug Jones, 'Role of word sense disambiguation in lexical acquisition: Predicting semantics from syntactic cues', in *COLING-ACL96*, (1996).

[5] Charles J. Fillmore, 'The case for case', in *Universals in Linguistic Theory*, (1968).

[6] Daniel Gildea and Daniel Jurafsky, 'Automatic labeling of semantic roles', *Computational Linguistic*, (2002).

[7] Ana-Maria Giuglea and Alessandro Moschitti, 'Semantic role labeling via FrameNet, VerbNet and PropBank', in *COLING-ACL06*, Sydney, Australia, (2006).

[8] Sabine Schulte im Walde, 'Clustering verbs semantically according to their alternation behaviour', *COLING*, (2000).

[9] T. Joachims, 'Making large-scale SVM learning practical.', in *Advances in Kernel Methods - Support Vector Learning*, eds., B. Schölkopf, C. Burges, and A. Smola, (1999).

[10] Christopher Johnson, Miriam Petruck, Collin Baker, Michael Ellsworth, Josef Ruppenhofer, and Charles Fillmore, 'Framenet: Theory and practice', *Berkeley, California*, (2003).

[11] Paul Kingsbury and Martha Palmer, 'From Treebank to PropBank', in *LREC02*, (2002).

[12] Karin Kipper, Hoa Trang Dang, and Martha Palmer, 'Class-based construction of a verb lexicon', in *AAAI00*, (2000).

[13] Mirella Lapata and Chris Brew, 'Using subcategorization to resolve verb class ambiguity.', in *Joint SIGDAT Conference on EMNLP and Very Large Corpora*, eds., Pascal Fung and Joe Zhou, College Park, Maryland, (1999).

[14] Mirella Lapata and Chris Brew, 'Verb class disambiguation using informative priors.', *Computational Linguistics*, (2004).

[15] Beth Levin, *English Verb Classes and Alternations A Preliminary Investigation*, Chicago: University of Chicago Press., 1993.

[16] Gina-Anne Levow, Bonnie Dorr, and Dekang Lin, 'Construction of chinese-english semantic hierarchy for information retrieval', *Technical report,University of Maryland, College Park*, (2000).

[17] Kenneth Litkowski, 'Senseval-3 task automatic labeling of semantic roles', in *Senseval-3*, (2004).

[18] Paola Merlo and Suzanne Stevenson, 'Automatic verb classification based on statistical distribution of argument structure', *Computational Linguistics*, (2001).

[19] Alessandro Moschitti, 'A study on convolution kernels for shallow semantic parsing', in *ACL04*, Barcelona, Spain, (2004).

[20] Sameer Pradhan, Kadri Hacioglu, Valeri Krugler, Wayne Ward, James H. Martin, and Daniel Jurafsky, 'Support vector learning for semantic argument classification', *Machine Learning*, (2005).

[21] Suzanne Stevenson, 'Automatic verb classification using grammatical features', in *EACL99*, Bergen, Norway, (1999).

[22] Nianwen Xue and Martha Palmer, 'Calibrating features for semantic role labeling', in *Proceedings of EMNLP 2004*, Barcelona, Spain, (2004). ACL04.