Towards Free-text Semantic Parsing: A Unified Framework Based on FrameNet, VerbNet and PropBank

Ana-Maria Giuglea and **Alessandro Moschitti**
University of Rome "Tor Vergata",
Rome, Italy
ana-maria.giuglea@topex.ro
moschitti@info.uniroma2.it

Abstract

This article describes a robust semantic parser that uses a broad knowledge base created by interconnecting three major resources: FrameNet, VerbNet and PropBank. The FrameNet corpus contains the examples annotated with semantic roles whereas the VerbNet lexicon provides the knowledge about the syntactic behavior of the verbs. We connect VerbNet and FrameNet by mapping the FrameNet frames to the VerbNet Intersective Levin classes. The PropBank corpus, which is tightly connected to the VerbNet lexicon, is used to increase the verb coverage and also to test the effectiveness of our approach. The results indicate that our model is an interesting step towards the design of free-text semantic parsers.

1 Introduction

During the last years a noticeable effort has been devoted to the design of lexical resources that can provide the training ground for automatic semantic role labelers. Unfortunately, most of the systems developed until now are confined to the scope of the resource that they use during the learning stage. A very recent example in this sense was provided by the CONLL 2005 Shared Task on PropBank (Kingsbury and Palmer, 2002) role labeling (Carreras and Màrquez, 2005). While the best F-measure recorded on a test set selected from the training corpus (WSJ) was 80%, on the Brown corpus, the F-measure dropped below 70%. The most significant causes for this performance decay were highly ambiguous and unseen predicates (i.e. predicates that do not have training examples, unseen in the training set).

On the FrameNet (Johnson et al., 2003) role labeling task, the Senseval-3 competition (Litkowski, 2004) registered similar results (~80%) by using the gold frame information as a given feature. No tests were performed outside FrameNet. In this paper, we show that when the frame feature is not used, the performance decay on different corpora reaches 30 points. Thus, the context knowledge provided by the frame is very important and a free-text semantic parser using FrameNet roles depends on the accurate automatic detection of this information.

In order to test the feasibility of such a task, we have trained an SVM (Support Vector Machine) Tree Kernel model for the automatic acquisition of the frame information. Although FrameNet contains three types of predicates (nouns, adjectives and verbs), we concentrated on the verb predicates and the roles associated with them. Therefore, we considered only the frames that have at least one verb lexical unit. Our experiments show that given a FrameNet predicate-argument structure, the task of identifying the originating 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 FrameNet and the predicates belonging to new application domains (that require new frames) are especially problematic as for them there is no available training data.

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 classes (Dang et al., 1998) that can be found as well in other predicate resources like PropBank and VerbNet (Kipper et al., 2000). Thus, we have designed a semi-automatic algorithm for assigning an Intersective Levin class to each FrameNet verb predicate.

The algorithm creates a mapping between FrameNet frames and the Intersective Levin classes. By doing that we could connect FrameNet to VerbNet and PropBank and obtain an *increased training set* for the Intersective Levin class. This leads to better verb coverage and a more robust semantic parser. The newly created knowledge base allows us to surpass the shortcomings that arise when FrameNet, VerbNet and PropBank are used separately while, at the same time, we benefit from the extensive research involving each of them (Pradhan et al., 2004; Gildea and Jurafsky, 2002; Moschitti, 2004).

We mention that there are 3,672 distinct verb senses[1] in PropBank and 2,351 distinct verb senses in FrameNet. Only 501 verb senses are in common between the two corpora which mean 13.64% of PropBank and 21.31% of FrameNet. Thus, by training an Intersective Levin class classifier on both PropBank and FrameNet we extend the number of available verb senses to 5,522.

In the remainder of this paper, Section 2 summarizes previous work done on FrameNet automatic role detection. It also explains in more detail why models based exclusively on this corpus are not suitable for free-text parsing. Section 3 focuses on VerbNet and PropBank and how they can enhance the robustness of our semantic parser. Section 4 describes the mapping between frames and Intersective Levin classes whereas Section 5 presents the experiments that support our thesis. Finally, Section 6 summarizes the conclusions.

2 Automatic semantic role detection on FrameNet

One of the goals of the FrameNet project is to design a 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 (Fillmore, 1968). The situations can be fairly simple depicting the entities involved and the roles they play or can be very complex and in this case they are called *scenarios*. 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*.

Several models have been developed for the automatic detection of the frame elements based on the FrameNet corpus (Gildea and Jurafsky, 2002; Thompson et al., 2003; Litkowski, 2004). While the algorithms used vary, almost all the previous studies divide the task into 1) the identification of the verb arguments to be labeled and 2) the tagging of each argument with a role. Also, most of the models agree on the core features as being: *Predicate*, *Headword*, *Phrase Type*, *Governing Category*, *Position*, *Voice* and *Path*. These are the initial features adopted by Gildea and Jurafsky (2002) (henceforth G&J) for both frame element identification and role classification.

A difference among the previous machine-learning models is whether the frame information was used as gold feature. Of particular interest for us is the impact of the frame over unseen predicates and unseen words in general. The results obtained by G&J are relevant in this sense; especially, the experiment that uses the frame to generalize from predicates seen in the training data to other predicates (i.e. when no data is available for a target word, G&J use data from the corresponding frame). The overall performance induced by the frame usage increased.

Other studies suggest that the frame is crucial when trying to eliminate the major sources of errors. In their error analysis, (Thompson et al., 2003) pinpoints that the verb arguments with headwords that are “rare” in a particular frame but not rare over the whole corpus are especially hard to classify. For these cases the frame is very important because it provides the context information needed to distinguish between different word senses.

Overall, the experiments presented in G&J’s study correlated with the results obtained in the Senseval-3 competition show that the frame feature increases the performance and decreases the amount of annotated examples needed in training (i.e. frame usage improves the generalization ability of the learning algorithm). On the other hand the results obtained without the frame information are very poor.

This behavior suggests that predicates in the same frame behave similarly in terms of their argument structure and that they differ with respect to other frames. From this perspective, having a broader verb knowledge base becomes of major importance for free-text semantic parsing.

[1] A verb sense is an Intersective Levin class in which the verb is listed.

Unfortunately, the 321 frames that contain at least one verb predicate cover only a small fraction of the English verb lexicon and of possible domains. Also from these 321 frames only 100 were considered to have enough training data and were used in Senseval-3 (see Litkowski, 2004 for more details).

Our approach for solving such problems involves the usage of a frame-like feature, namely the Intersective Levin class. We show that the Levin class is similar in many aspects to the frame and can replace it with almost no loss in performance. At the same time, Levin class provides better coverage as it can be learned also from other corpora (i.e. PropBank). We annotate FrameNet with Intersective Levin classes by using a mapping algorithm that exploits current theories of linking. Our extensive experimentation shows the validity of our technique and its effectiveness on corpora different from FrameNet. The next section provides the theoretical support for the unified usage of FrameNet, VerbNet and PropBank, explaining why and how is possible to link them.

3 Linking FrameNet to VerbNet and PropBank

In general, predicates belonging to the same FrameNet frame have a coherent syntactic behavior that is also different from predicates pertaining to other frames (G&J). This finding is consistent with theories of linking that claim that the syntactic behavior of a verb can be predicted from its semantics (Levin 1993, Levin and Rappaport Hovav, 1996). This insight determined us to study the impact of using a feature based on Intersective Levin classes instead of the frame feature when classifying FrameNet semantic roles. The main advantage of using Levin classes comes from the fact that other resources like PropBank and the VerbNet lexicon contain this kind of information. Thus, we can train a Levin class classifier also on the PropBank corpus, considerably increasing the verb knowledge base at our disposal. Another advantage derives from the syntactic criteria that were applied in defining the Levin clusters. As shown later in this article, the syntactic nature of these classes makes them easier to classify than frames, when using only syntactic and lexical features.

More precisely, the Levin clusters 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**

[$_{\text{Subject, Agent}}$ The butcher] cuts [$_{\text{Direct Object, Patient}}$ the meat].
[$_{\text{Subject, Patient}}$ The meat] cuts easily.

(b) **Causative/inchoative Alternation**

[$_{\text{Subject, Agent}}$ Janet] broke [$_{\text{Direct Object, Patient}}$ the cup].
[$_{\text{Subject, Patient}}$ 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 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 that is constructed based on a more refined version of the Levin classification called Intersective Levin classes (Dang et al., 1998). 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 and 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 share similarly labeled arguments. Inside one Intersective 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.

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[2]:

(1) Coherent syntactic behavior of verbs inside one cluster,
(2) Different syntactic properties between any two distinct verb clusters,
(3) Shared set 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 Intersective Levin classes and the frames. In order to create a pair ⟨FrameNet frame, VerbNet class⟩, our mapping algorithm checks both the syntactic and semantic consistency by comparing the role frequency distributions on different syntactic positions for the two candidates. The algorithm is described in detail in the next section.

4 Mapping FrameNet frames to VerbNet classes

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 ⟨FrameNet frame, VerbNet class⟩ (see Figure 1); (b) we refine the pairs obtained in the previous step based on diathesis alternation criteria, i.e. the verbs pertaining to the FrameNet frame have to undergo the same diathesis alternation that characterize the corresponding VerbNet class (see Figure 2) and (c) we manually check and correct the resulting mapping. In the next sections we will explain in more detail each step of the mapping algorithm.

4.1 Linking frames and Intersective Levin classes based on common verbs

During the first phase of the algorithm, given a frame, we compute its intersection with each VerbNet class. We choose as candidate for the mapping the Intersective Levin class that has the largest number of verbs in common with the given frame (Figure 1, line (I)). If the size of the intersection between the FrameNet frame and the candidate VerbNet class is bigger than or equal to 3 elements then we form a pair ⟨FrameNet frame, VerbNet class⟩ that qualifies for the second step of the algorithm.

Only the frames that have more than three verb lexical units are candidates for this step (frames with less than 3 members cannot pass condition (II)). This excludes a number of 60 frames that will subsequently be mapped manually.

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

Figure 1. Linking FrameNet frames and VerbNet classes

4.2 Refining the mapping based on verb alternations

In order to assign a VerbNet class to a frame, we have to check that the verbs belonging to that frame respect the diathesis alternation criteria used to define the VerbNet class. Thus, the pairs ⟨FrameNet frame, VerbNet class⟩ formed in step (I) of the mapping algorithm have to undergo a validation step that verifies the similarity between the enclosed FrameNet frame and VerbNet class. This validation process has several substeps.

First, we make use of the property (3) of the Levin classes and FrameNet frames presented in the previous 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 FrameNet is annotated with frame-specific semantic roles we manually mapped these roles into the VerbNet set of thematic roles. Given a frame, we assigned thematic roles to all frame elements that are associated with verbal predicates. For example the roles *Speaker*, *Addressee*, *Message* and *Topic* from the *Telling* frame were respectively mapped into *Agent*, *Recipient*, *Theme* and *Topic*.

[2] For FrameNet, properties 1 and 2 are true for most of the frames but not for all. See section 4.4 for more details.

Second, we build a frequency distribution of VerbNet thematic roles on different syntactic position. Based on our observation and previous studies (Merlo and Stevenson, 2001), we assume that each Levin class has a distinct frequency distribution of roles on different grammatical slots. As we do not have matching grammatical function in FrameNet and VerbNet, we approximate that *subjects* and *direct objects* are more likely to appear on positions adjacent to the predicate, while *indirect objects* appear on more distant positions. The same intuition is used successfully by G&J in the design of the *Position* feature.

We will acquire from the corpus, for each thematic role θ_i, the frequencies with which it appears on an adjacent (ADJ) or distant (DST) position in a given frame or VerbNet class (i.e. #(θ_i, class, position)). Therefore, for each frame and class, we obtain two vectors with thematic role frequencies corresponding respectively to the adjacent and distant positions (see Figure 2). We compute a score for each pair ⟨FrameNet frame, VerbNet class⟩ using the normalized scalar product. We give a bigger weight to the adjacent dot product multiplying its score by 2/3 with respect to the distant dot product that is multiplied by 1/3. We do this to minimize the impact that adjunct roles like Temporal and Location (that appear mostly on the distant positions) could have on the final outcome.

$$TR = \{\theta_i : \theta_i \text{ is the } i^{th} \text{ theta role of the VerbNet theta role set}\}$$
$$for\ each\ (F, C) \in Pairs$$
$$\overrightarrow{ADJ}^F = (o_1,..,o_n), where\ o_i = \#(\theta_i, F, position = \textbf{adjacent})$$
$$\overrightarrow{DST}^F = (o_1,..,o_n), where\ o_i = \#(\theta_i, F, position = \textbf{distant})$$
$$\overrightarrow{ADJ}^C = (o_1,..,o_n), where\ o_i = \#(\theta_i, C, position = \textbf{adjacent})$$
$$\overrightarrow{DST}^C = (o_1,..,o_n), where\ o_i = \#(\theta_i, C, position = \textbf{distant})$$
$$Score_{F,C} = \frac{2}{3} \times \frac{\overrightarrow{ADJ}^F \bullet \overrightarrow{ADJ}^C}{\| \overrightarrow{ADJ}^F \| \times \| \overrightarrow{ADJ}^C \|} + \frac{1}{3} \times \frac{\overrightarrow{DST}^F \bullet \overrightarrow{DST}^C}{\| \overrightarrow{DST}^F \| \times \| \overrightarrow{DST}^C \|}$$

Figure 2. Mapping algorithm – refining step

The above frequency vectors are computed for FrameNet directly from the corpus of predicate-argument structure examples associated with each frame. The examples associated with the VerbNet lexicon are extracted from the PropBank corpus. In order to do this we apply a preprocessing step in which each label ARG0..N is replaced with its corresponding thematic role given the Intersective Levin class of the predicate. We assign the same roles to the adjuncts all over PropBank as they are general for all verb classes. The only exception is ARGM-DIR that can correspond to Source, Goal or Path. We assign different roles to this adjunct based on the prepositions. We ignore some adjuncts like ARGM-ADV or ARGM-DIS because they cannot bear a thematic role.

4.3 Mapping Results

We found that only 133 VerbNet classes have correspondents among FrameNet frames. Also, from the frames mapped with an automatic score smaller than 0.5 points almost a half did not match any of the existing VerbNet classes[3]. A summary of the results is depicted in Table 1. The first column contains the automatic score provided by the mapping algorithm when comparing frames with Intersective Levin classes. The second column contains the number of frames for each score interval. The third column contains the percentage of frames, per each score interval, that did not have a corresponding VerbNet class and finally the forth column contains the accuracy of the mapping algorithm.

Score	*No. of Frames*	*Not mapped*	*Correct*	*Overall Correct*
[0,0.5]	118	48.3%	82.5%	89.6%
(0.5,0.75]	69	0	84%	
(0.75,1]	72	0	100%	

Table 1. Results of the mapping algorithm

4.4 Discussion

In the literature, other studies compared the Levin classes to the FrameNet frames (Baker and Ruppenhofer, 2002). Their findings suggest that although the two set of clusters are roughly equivalent there are also several types of mistmaches: 1) Levin classes that are narrower than the corresponding frames, 2) Levin classes that are broader that the corresponding frames and 3) overlapping groupings. For our task, point 2 does not pose a problem. Points 1 and 3 however suggest that there are cases in which to one FrameNet frame corresponds more than one Levin class. By investigating such cases we noted that the mapping algorithm consistently assigns scores below 75% to cases that match problem 1 (two Levin classes inside one frame) and below 50% to cases that match problem 3 (more than two Levin classes inside one frame). Thus, in order to increase the accuracy of our results a first step should be to assign an

[3] The automatic mapping can be improved by manually assigning the FrameNet frames of the pairs that receive a score lower than 0.5.

Intersective Levin class to each of the verbs pertaining to frames with score lower than 0.75. Nevertheless the current results are encouraging as they show that the algorithm is achiving its purpose by successfully detecting syntactic incoherencies that can be subsequently corrected manually. Also, in the next section we will show that our current mapping achieves very good results, giving evidence for the effectivenes of the Levin class feature.

5 Experiments

In the previous section we have presented the algorithm for annotating the verb predicates of FrameNet with Intersective Levin classes. In order to show the effectiveness of this annotation and of the Intersective Levin class in general we have performed several experiments.

First, we trained (1) an ILC multiclassifier from FrameNet, (2) an ILC multiclassifier from PropBank and (3) a frame multiclassifier from FrameNet. We compared the results obtained when trying to classify the VerbNet class with the results obtained when classifying frame. We show that Intersective Levin classes are easier to detect than FrameNet frames.

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

Another set of experiments regards the generalization property of the Intersective Levin class. We show the impact of this feature 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 Intersective Levin class, automatically detected frame and automatically detected Intersective Levin class.

Finally, we simulate the difficulty of free text by annotating PropBank with FrameNet semantic roles. We use PropBank because it is different from FrameNet from a domain point of view. This characteristic makes PropBank a difficult test bed for semantic role models trained on FrameNet.

In the following section we present the results obtained for each of the experiments mentioned above.

5.1 Experimental setup

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

For the experiments on FrameNet 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 (Charniak, 2000) to generate parse trees automatically.

For classification, we used the SVM-light-TK software available at `http://ai-nlp.info.uniroma2.it/moschitti` which encodes tree kernels in the SVM-light software (Joachims, 1999). The classification performance was evaluated using the F_1 measure for the single-argument classifiers and the accuracy for the multiclassifiers.

5.2 Automatic VerbNet vs. automatic FrameNet frame detection

In these experiments we classify Intersective Levin classes (ILC) on PropBank (PB) and FrameNet (FN) and frame on FrameNet. For the training stage we use SVMs with Tree Kernels.

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 train SVMs with structures drawn directly from the syntactic parse tree of the sentence.

The kernel that we employed in our experiments is based on the SCF structure devised in (Moschitti, 2004). We slightly modified SCF by adding the headwords of the arguments, useful for representing the selectional preferences.

For frame detection on FrameNet, we trained our classifier on 46,734 training instances and tested on 11,650 testing instances, obtaining an accuracy of 91.11%. 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 that ILC detection is performed better than frame detection on both FrameNet and PropBank. Also, the results obtained on ILC on PropBank are similar with the ones obtained on ILC on FrameNet. This suggests that the training corpus does not have a major influence. Also, the SCF-based tree kernel seems to be robust in what concerns the quality of the parse trees. The performance decay is very small on FrameNet that uses automatic parse trees with respect to PropBank that contains gold parse trees. These properties suggest that ILC are very suitable for free text.

	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 . F_1 and accuracy of the argument classifiers and the overall multiclassifier for Intersective Levin class

5.3 Automatic semantic role labeling on FrameNet

In the experiments involving semantic role labelling, we used a SVM with a polynomial kernel. We adopted the standard features developed for semantic role detection by Gildea and Jurafsky (see Section 2). Also, we considered some of the features designed by (Pradhan et al., 2004): *First and Last Word/POS in Constituent*, *Subcategorization*, *Head Word of Prepositional Phrases* and the *Syntactic Frame* feature from (Xue and Palmer, 2004). 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 (G&J).

We note that automatic frame results are very similar to automatic ILC results suggesting that ILC feature is a very good candidate for replacing the frame feature. Also, both automatic features are very effective, decreasing the error rate of 20%.

	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. F_1 and accuracy of the argument classifiers and the overall multiclassifier for FrameNet semantic roles

5.4 Semantic role learning curve when using Intersective Levin classes

The next set of experiments show the impact of the ILC feature on semantic role labelling when few training data is available (Figure 3). As can be noted, the automatic ILC features (i.e. derived with classifers trained on FrameNet or PB) produce accuracy almost as good as the gold ILC one. Another observation is that the SRL classifiers are not saturated and more training examples would improve their accuracy.

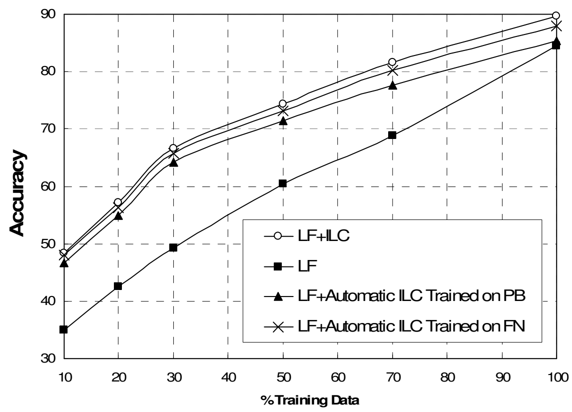

Figure 3. Semantic Role learning curve

5.5 Annotating PropBank with FrameNet semantic roles

To show that our approach can be suitable for semantic role free-text annotation, we have automatically classified PropBank sentences with the FrameNet 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. This corresponds to an accuracy of 81% with Intersective Levin class versus 62% without it.

6 Conclusions

In this paper we have shown that the Intersective Levin class feature can successfully replace the FrameNet frame feature. By doing that we could interconnect FrameNet to VerbNet and PropBank obtaining better verb coverage and a more robust semantic parser. Our good results show that we have defined an effective framework which is a promising step toward the design of free-text semantic parsers.
In the future, we intend to measure the effectiveness of our system by testing on larger, more comprehensive corpora and without relying on any manual annotation.

Reference

Collin Baker and Josef Ruppenhofer. 2002. FrameNet's frames vs. Levin's verb classes. 28th Annual Meeting of the Berkeley Linguistics Society.

Xavier Carreras and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling. CONLL'05.

Eugene Charniak. 2000. A Maximum-Entropy-Inspired Parser. ANLP'00

Hoa Trang Dang, Karin Kipper, Martha Palmer and Joseph Rosenzweig. 1998. Investigating regular sense extensions based on Intersective Levin classes. Coling-ACL'98.

Charles Fillmore. 1968. The case for case. Universals in Linguistic Theory.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. CL Journal.

Christopher Johnson, Miriam Petruck, Collin Baker, Michael Ellsworth, Josef Ruppenhofer, and Charles Fillmore. 2003. FrameNet: Theory and Practice. Berkeley, California.

Paul Kingsbury, Martha Palmer. 2002. From Tree-Bank to PropBank. LREC'02.

Karin Kipper, Hoa Trang Dang and Martha Palmer. 2000. Class-based construction of a verb lexicon. AAAI'00.

Beth Levin. 1993. English Verb Classes and Alternations A Preliminary Investigation. Chicago: University of Chicago Press.

Kenneth Litkowski. 2004. Senseval-3 task automatic labeling of semantic roles. Senseval-3.

Paola Merlo and Suzanne Stevenson. 2001. Automatic verb classification based on statistical distribution of argument structure. CL Journal.

Alessandro Moschitti. 2004. A study on convolution kernel for shallow semantic parsing. ACL'04.

Sameer Pradhan, Kadri Hacioglu, Valeri Krugler, Wayne Ward, James H. Martin, and Daniel Jurafsky. 2004. Support vector learning for semantic argument classification. Machine Learning Journal.

Cynthia A. Thompson, Roger Levy, and Christopher Manning. 2003. A Generative Model for FrameNet Semantic Role Labeling. ECML'03.

Thorsten Joachims. 1999. Making large-scale SVM learning practical.. Advances in Kernel Methods - Support Vector Learning.

Nianwen Xue and Martha Palmer. 2004. Calibrating features for semantic role labeling. EMNLP'04.