A Supervised Algorithm for Verb Disambiguation into VerbNet Classes

Omri Abend[1] **Roi Reichart**[2] **Ari Rappoport**[1]

[1]Institute of Computer Science , [2]ICNC
Hebrew University of Jerusalem
{omria01|roiri|arir}@cs.huji.ac.il

Abstract

VerbNet (VN) is a major large-scale English verb lexicon. Mapping verb instances to their VN classes has been proven useful for several NLP tasks. However, verbs are polysemous with respect to their VN classes. We introduce a novel supervised learning model for mapping verb instances to VN classes, using rich syntactic features and class membership constraints. We evaluate the algorithm in both in-domain and corpus adaptation scenarios. In both cases, we use the manually tagged Semlink WSJ corpus as training data. For in-domain (testing on Semlink WSJ data), we achieve 95.9% accuracy, 35.1% error reduction (ER) over a strong baseline. For adaptation, we test on the GENIA corpus and achieve 72.4% accuracy with 10.7% ER. This is the first large-scale experimentation with automatic algorithms for this task.

1 Introduction

The organization of verbs into classes whose members exhibit similar syntactic and semantic behavior has been discussed extensively in the linguistics literature (see e.g. (Levin and Rappaport Hovav, 2005; Levin, 1993)). Such an organization helps in avoiding lexicon representation redundancy and enables generalizations across similar verbs. It can also be of great practical use, e.g. in compensating NLP statistical models for data sparseness. Indeed, Levin's seminal work had motivated much research aimed at automatic discovery of verb classes (see Section 2).

VerbNet (VN) (Kipper et al., 2000; Kipper-Schuler, 2005) is a large scale, publicly available domain independent verb lexicon that builds on Levin classes and extends them with new verbs, new classes, and additional information such as semantic roles and selectional restrictions. VN classes were proven beneficial for Semantic Role Labeling (SRL) (Swier and Stevenson, 2005), Semantic Parsing (Shi and Mihalcea, 2005) and building conceptual graphs (Hensman and Dunnion, 2004). Levin-inspired classes have been used in several NLP tasks, such as Machine Translation (Dorr, 1997) and Document Classification (Klavans and Kan, 1998).

Many applications that use VN need to map verb instances onto their VN classes. However, verbs are polysemous with respect to VN classes. Semlink (Loper et al., 2007) is a dataset that maps each verb instance in the WSJ Penn Treebank to its VN class. The mapping has been created using a combination of automatic and manual methods. Yi et al. (2007) have used Semlink to improve SRL.

In this paper we present the first large-scale experimentation with a supervised machine learning classification algorithm for disambiguating verb instances to their VN classes. We use rich syntactic features extracted from a treebank-style parse tree, and utilize a learning algorithm capable of imposing class membership constraints, thus taking advantage of the nature of our task. We use Semlink as the training set.

We evaluate our algorithm in both in-domain and corpus adaptation scenarios. In the former, we test on the WSJ (using Semlink again), obtaining 95.9% accuracy with 35.1% error reduction (ER) over a strong baseline (most frequent

Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 9–16
Manchester, August 2008

class) when using a modern statistical parser. In the corpus adaptation scenario, we disambiguate verbs in sentences taken from outside the training domain. Since the manual annotation of new corpora is costly, and since VN is designed to be a domain independent resource, adaptation results are important to the usability in NLP in practice. We manually annotated 400 sentences from GENIA (Kim et al., 2003), a medical domain corpus[1]. Testing on these, we achieved 72.4% accuracy with 10.7% ER. Our adaptation scenario is complete in the sense that the parser we use was also trained on a different corpus (WSJ). We also report experiments done using gold-standard (manually created) parses.

The most relevant previous works addressing verb instance class classification are (Lapata and Brew, 2004; Li and Brew, 2007; Girju et al., 2005). The former two do not use VerbNet and their experiments were narrower than ours, so we cannot compare to their results. The latter mapped to VN, but used a preliminary highly restricted setup where most instances were monosemous. For completeness, we compared our method to theirs[2], achieving similar results.

We review related work in Section 2, and discuss the task in Section 3. Section 4 introduces the model, Section 5 describes the experimental setup, and Section 6 presents our results.

2 Related Work

VerbNet. VN is a major electronic English verb lexicon. It is organized in a hierarchical structure of classes and sub-classes, each sub-class inheriting the full characterization of its super-class. VN is built on a refinement of the Levin classes, the intersective Levin classes (Dang et al., 1998), aimed at achieving more coherent classes both semantically and syntactically. VN was also substantially extended (Kipper et al., 2006) using the Levin classes extension proposed in (Korhonen and Briscoe, 2004). VN today contains 3626 verb *lemmas* (forms), organized in 237 main classes having 4991 verb *types* (we refer to a lemma with an ascribed class as a type). Of the 3626 lemmas, 912 are polysemous (i.e., appear in more than a single class). VN's significant coverage of the English verb lexicon is demonstrated by the 75.5% coverage of VN classes over PropBank[3] instances (Loper et al., 2007). Each class contains rich semantic information, including semantic roles of the arguments augmented with selectional restrictions, and possible subcategorization frames consisting of a syntactic description and semantic predicates with temporal information. VN thematic roles are relatively coarse, vs. the situation-specific FrameNet role system or the verb-specific PropBank role system, enabling generalizations across a wide semantic scope. Swier and Stevenson (2005) and Yi et al. (2007) used VN for SRL.

Verb type classification. Quite a few works have addressed the issue of verb type classification and in particular classification to 'Levin inspired' classes (e.g., (Schulte im Walde, 2000; Merlo and Stevenson, 2001)). Such work is not comparable to ours, as it deals with verb type (sense) rather than verb token (instance) classification.

Verb token classification. Lapata and Brew (2004) dealt with classification to Levin classes of polysemous verbs. They established a prior from the BNC in an unsupervised manner. They also showed that this prior helps in the training of a naive Bayes classifier employed to distinguish between possible verb classes of a given verb in a given frame (when the ambiguity is not solved by knowing the frame alone). Li and Brew (2007) extended this model by proposing a method to train the class disambiguator without using hand-tagged data. While these papers have good results, their experimental setup was rather narrow and used only at most 67 polysemous verbs (in 4 frames). VN includes 912 polysemous verbs, of which 695 appeared in our in-domain experiments.

Girju et al. (2005) performed the only previous work we are aware of that addresses the problem of token level verb disambiguation into VN classes. They treated the task as a supervised learning problem, proposing features based on a POS tagger, a Chunker and a named entity classifier. In order to create the data[4], they used a mapping between Propbank rolesets and VN classes, and took the instances in WSJ sections 15-18,20,21 that were annotated by Propbank and for which the roleset determines the VN class uniquely. This resulted in most instances being in fact monosemous. Their

[1] Our annotations will be made available to the community.

[2] Using the same sentences and instances, obtained from the authors.

[3] Propbank (Palmer et al., 2005) is a corpus annotation of the WSJ sections of the Penn Treebank with semantic roles of each verbal proposition.

[4] Semlink was not available then.

experiment was conducted in a WSJ in-domain scenario, and in a much narrower scope than in this paper. They had 870 (39 polysemous) unique verb lemmas, compared to 2091 (695 polysemous) in our in-domain scenario. They did not test their model in an adaptation scenario. The scope and difficulty contrast between our setup and theirs are demonstrated by the large differences in the number of instances and in the percentage of polysemous instances: 972/12431 (7.8%) in theirs, compared to 49571/84749 (58.5%) in our in-domain scenario (training+test). We compared our method to theirs for completeness and achieved similar results.

Semlink. The Semlink project (Yi et al., 2007; Loper et al., 2007) aims to create a mapping of PropBank, FrameNet (Baker et al., 1998), WordNet (henceforth WN) and VN to one another, thus allowing these resources to synergize. In addition, the project includes the most extensive token mapping of verbs to their VN classes available today. It covers all verbs in the WSJ sections of the Penn Treebank within VN coverage (out of 113K verb instances, 97K have lemmas present in VN).

3 Nature of the Task

Polysemy is a major issue in NLP. Verbs are not an exception, resulting in a single verb form (lemma) appearing in more than a single class. This polysemy is also present in the original Levin classification, where polysemous classes account for more than 48% of the BNC verb instances (Lapata and Brew, 2004).

Given a verb instance whose lemma is within the coverage of VN, given the sentence in which it appears, given a parse tree of this sentence (see below), and given the VN resource, our task is to classify the verb instance to its correct VN class. There are currently 237 possible classes[5]. Each verb has only a few possible classes (no more than 10, but only about 2.5 on the average over the polysemous verbs). Depending on the application, the parse tree for the sentence may be either a gold standard parse or a parse tree generated by a parser. We have experimented with both options.

The task can be viewed in two complementary ways: per-class and per-verb type. The per-class perspective takes into consideration the small number of classes relative to the number of types[6]. A classifier may gather valuable information for all members of a certain VN class, without seeing all of its members in the training data. From this perspective the task resembles POS tagging. In both tasks there are many dozens (or more) of possible labels, while each word has only a small subset of possible labels. Different words may receive the same label.

The per-verb perspective takes into consideration the special properties of every verb type. Even the best lexicons necessarily ignore certain idiosyncratic characteristics of the verb when assigning it to a certain class. If a verb appears many times in the corpus, it is possible to estimate its parameters to a reasonable reliability, and thus to use its specific distributional properties for disambiguation. Viewed in this manner, the task resembles a word sense disambiguation (WSD) task: each verb has a small distinct set of senses (types), and no two different verbs have the same sense.

The similarity to WSD suggests that our task might be solved by WN sense disambiguation followed by a mapping from WN to VN. However, good results are not to be expected, due to the medium quality of today's WSD algorithms and because the mapping between WN and VN is both incomplete and many-to-many[7]. Even for a perfect WN WSD algorithm, the resulting WN synset may not be mapped to VN at all or may be mapped onto multiple VN classes. We experimented with this method and obtained results below the MF baseline we used[8].

The above discussion does not rule out the possibility of obtaining reasonable results through applying a high quality WSD engine followed by a WN to VN mapping. However, there are much fewer VN classes than WN classes per verb. This may result in the WSD engine learning many distinctions that are not useful in this context, which may in turn jeopardize its performance with respect to our task. Moreover, a word sense may belong to a single verb only while a VN class contains many verbs. Consequently, the performance

[5] We ignore sub-class distinctions. This is justified since in 98.2% of the in-coverage instances in Semlink, knowing the verb and its class suffices for knowing its exact sub-class.

[6] 237 classes vs. 4991 types.

[7] In the WN to VN mapping built into VN, 14.69% of the covered WN synsets were mapped to more than a single VN class.

[8] We used the publicly available SenseLearner 2.0, the VB-Collocations model. We chose VN classes containing the lemma in random when a single mapping is not specified. We obtained 67.74% accuracy on section 00 of the WSJ, which is less than the MF baseline. See Sections 5 and 7.

on a certain lemma may benefit from training instances of other lemmas.

Note that our task is not reducible to VN frame identification (a non-trivial task given the richness of the information used to define a frame in VN). Although the categorizing criterion for Levin's classification is the subset of frames the verb may appear in (equivalently, the diathesis alternations the verbal proposition may perform), knowing only the frame in which an instance appears does not suffice, as frames are shared among classes.

4 The Learning Model

As common in supervised learning models, we encode the verb instances into feature vectors and then apply a learning algorithm to induce a classifier. We first discuss the feature set and then the learning algorithm.

Features. Our feature set heavily relies on syntactic annotation. Dorr and Jones (1996) showed that perfect knowledge of the allowable syntactic frames for a verb allows 98% accuracy in type assignment to Levin classes. This motivates the encoding of the syntactic structure of the sentence as features, since we have no access to *all* frames, only to the one appearing in the sentence.

Since some verbs may appear in the same syntactic frame in different VN classes, a model relying on the syntactic frame alone would not be able to disambiguate instances of these verbs when appearing in those frames. Hence our features include lexical context words. The parse tree enables us to use words that appear in specific syntactic slots rather than in a linear window around the verb. To this end, we use the head words of the neighboring constituents. The definition of the head of a constituent is given in (Collins, 1999).

Our feature set is comprised of two parallel sets of features. The first contains features extracted from the parse tree and the verb's lemma as a standalone feature. In the second set, each feature is a conjunction of a feature from the first set with the verb's lemma. By doing so we created a general feature space shared by all verbs, and replications of it for each and every verb. This feature selection strategy was chosen in view of the two perspectives on the task (per-class and per-verb) discussed in Section 3.

Our first set of features encodes the verb's context as inferred from the sentence's parse tree (Figure 1). We attempt to encode both the syntactic frame, by encoding the tree structure, and the argument preferences, by encoding the head words of the arguments and their POS. The restriction on the verb's parent being the head constituent of its grandparent is done in order to focus on the correct verb in verb series such as 'intend to run'.

First Feature Set
The stemmed head words, POS, parse tree labels, function tags, and ordinals of the verb's right k_r siblings (k_r is the maximum number of right siblings in the corpus. These are at most $5k_r$ different features).
The stemmed head words, POS, labels, function tags and ordinals of the verb's left k_l siblings, as above.
The stemmed head word & POS of the 'second head word' nodes on the left and right (see text for precise definition).
All of the above features employed on the siblings of the parent of the verb (only if the verb's parent is the head constituent of its grandparent)
The number of right/left siblings of the verb.
The number of right/left siblings of the verb's parent.
The parse tree label of the verb's parent.
The verb's voice (active or passive).
The verb's lemma.

Figure 1: The first set of features in our model. All of them are binary. The final feature set includes two sets: the set here, and a set obtained by its conjunction with the verb's lemma.

The 3rd cell in the table makes use of a 'second head word' node, defined as follows. Consider a left sibling (right siblings are addressed analogously) M of the verb's node. Take the node H in the subtree of M where M's head appears. H is a descendent of a node J which is a child of M. The 'second head word' node is J's sibling on the right. For example, in the sentence *We went to school* (see Figure 2) the head word of the PP 'to school' is 'to', and the 'second head word' node is 'school'. The rationale is that 'school' could be a useful feature for 'went', in addition to 'to', which is highly polysemous (note that it is also a feature for 'went', in the 1st and 2nd cells of the table). The voice feature was computed using a simple heuristic based on the verb's POS tag (past participle) and presence of auxiliary verbs to its left.

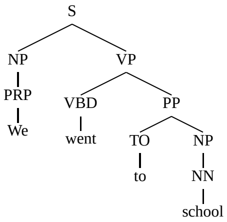

Figure 2: An example parse tree for the 'second head word' feature.

The current set of features does not detect verb particle constructions. We leave this for future research.

Learning Algorithm. Our learning task can be formulated as follows. Let x_i denote the feature vector of an instance i, and let X denote the space of all such feature vectors. The subset of possible labels for x_i is denoted by C_i, and the correct label by $c_i \in C_i$. We denote the label space by S. Let T be the training set of instances $T = \{< x_1, C_1, c_1 >, < x_2, C_2, c_2 >, ..., < x_n, C_n, c_n > \} \subseteq (X \times 2^S \times S)^n$, where n is the size of the training set. Let $< x_{n+1}, C_{n+1} > \in (X \times 2^S)$ be a new instance. Our task is to select which of the labels in C_{n+1} is its correct label c_{n+1} (x_{n+1} does not have to be a previously observed lemma, but its lemma must appear in a VN class).

The structure of the task lets us apply a learning algorithm that is especially appropriate for it. What we need is an algorithm that allows us to restrict the possible labels of each instance, both in training and in testing. The sequential model algorithm presented by Even-Zohar and Roth (2001) directly supports this requirement. We use the SNOW learning architecture for multi-class classification (Roth, 1998), which contains an implementation of that algorithm [9].

5 Experimental Setup

We used SemLink VN annotations and parse trees on sections 02-21 of the WSJ Penn Treebank for training, and section 00 as a development set, as is common in the parsing community. We performed two parallel sets of experiments, one using manually created gold standard parse trees and one using parse trees created by a state-of-the-art parser (Charniak and Johnson, 2005) (Note that this parser does not output function tags). The parser was also trained on sections 02-21 and tuned on section 00[10]. Consequently, our adaptation scenario is a full adaptation situation in which both the parser and the VerbNet training data are not in the test domain. Note that generative parser adaptation results are known to be of much lower quality than in-domain results (Lease and Charniak, 2005). The quality of the discriminative parser we used did indeed decrease in our adaptation scenario (Section 7).

The training data included 71209 VN in-scope instances (of them 41753 polysemous) and the development 3624 instances (2203 polysemous). An 'in-scope' instance is one that appears in VN and is tagged with a verb POS. The same trained model was used in both the in-domain and adaptation scenarios, which only differ in their test sets.

In-Domain. Tests were held on sections 01,22,23,24 of WSJ PTB. Test data includes all in-scope instances for which there is a SemLink annotation, yielding 13540 instances, 7798 (i.e., 57.6%) of them polysemous.

Adaptation. For the testing we annotated sentences from GENIA (Kim et al., 2003) (version 3.0.2). The GENIA corpus is composed of MEDLINE abstracts related to transcription factors in human blood cells. We annotated 400 sentences from the corpus, each including at least one in-scope verb instance. We took the first 400 sentences from the corpus that met that criterion[11]. After cleaning some GENIA POS inconsistencies, this amounts to 690 in-scope instances (380 of them polysemous). The tagging was done by two annotators with an inter-annotator agreement rate of 80.35% and Kappa 67.66%.

Baselines. We used two baselines, random and most frequent (MF). The random baseline selects uniformly and independently one of the possible classes of the verb. The most frequent (MF) baseline selects the most frequent class of the verb in the training data for verbs seen while training, and selects in random for the unseen ones. Consequently, it obtains a perfect score over the monosemous verbs. This baseline is a strong one and is common in disambiguation tasks.

We repeated all of the setup above in two sce-

[9] Experiments on development data revealed that for verbs for which almost all of the training instances are mapped to the same VN class, it is most beneficial to select that class. Thus, where more than 90% of the training instances of a verb are mapped to the same class, our algorithm mapped the instances of the verb to that class regardless of the context.

[10] For the very few sentences out of coverage for the parser, we used the MF baseline (see below).

[11] Discarding the first 120 sentences, which were used to design the annotator guidelines.

narios. In the first (*main*) scenario, in-scope instances were always mapped to VN classes, while in the second (*'other is possible'* (OIP)) scenario, in-scope instances were allowed to be tagged (during training) and classified (during test) as not belonging to any existing VN class[12]. In all cases, out-of-scope instances (verbs whose lemmas do not appear in VN) were ignored. For the OIP scenario, we used a different 'other' label for each of the lemmas, not a single label shared by them all.

6 Results

Table 1 shows our results. In addition to the overall results, we also show results for the polysemous ones alone, since the task is trivial for the monosemous ones. The results using gold standard parses effectively set an upper bound on our model's performance, while those using statistical parser output demonstrate its current usability.

In-Domain. Results are shown in the WSJ → WSJ columns of Table 1. Using gold standard parses (top), we achieve 96.42% accuracy overall. Over the polysemous verbs, the accuracy is 93.68%. This translates to an error reduction over the MF baseline of 43.35% overall and 43.22% for the polysemous verbs. In the 'other is possible' scenario (right), we obtained 36.67% error reduction. Using a state-of-the-art parser (Charniak and Johnson, 2005) (bottom), we experienced some degradation of the results (as expected), but they remained significantly above baseline. We achieve 95.9% accuracy overall and 92.77% for the polysemous verbs, which translates to about 35.13% and 35.04% error reduction respectively. In the OIP scenario, we obtained 28.95% error reduction.

The results of the random baseline for the in-domain scenario are substantially worse than the MF baseline. On the WSJ the random baseline scored 66.97% (37.51%) accuracy in the main (OIP) scenarios.

Adaptation. Here we test our model's ability to generalize across domains. Since VN is supposed to be a domain independent resource, we hope to acquire statistics that are relevant across domains as well and so to enable us to automatically map verbs in domains of various genres. The results are shown in the WSJ → GENIA columns of Table 1. When using gold standard parses, our model scored 73.16%. This translates to about 13.17% ER on GENIA. We interestingly experienced very little degradation in the results when moving to parser output, achieving 72.4% accuracy which translates to 10.71% error reduction over the MF baseline. The random baseline on GENIA was again worse than MF, obtaining 66.04% accuracy as compared to 69.09% of MF (in the OIP scenario, 39.12% compared to 46.41%).

Run-time performance. Given a parsed corpus, our main model trains and runs in no more than a few minutes for a training set of ∼60K instances and a test set of ∼11K instances, using a Pentium 4 CPU 2.40GHz with 1GB main memory. The bottleneck in tagging large corpora using our model is thus most likely the running time of current parsers.

7 Discussion

In this paper we introduced a new statistical model for automatically mapping verb instances into VerbNet classes, and presented the first large-scale experiments for this task, for both in-domain and corpus adaptation scenarios.

Using gold standard parse trees, we achieved 96.42% accuracy on WSJ test data, showing 43.35% error reduction over a strong baseline. For adaptation to the GENIA corpus, we showed 13.1% error reduction over the baseline. A surprising result in the context of adaptation is the little influence of using gold standard parses versus using parser output, especially given the relatively low performance of today's parsers in the adaptation task (91.4% F-score for the WSJ in-domain scenario compared to 81.24% F-score when parsing our GENIA test set). This is an interesting direction for future work.

In addition, we conducted some additional preliminary experiments in order to shed light on some aspects of the task. The experiments reported below were conducted on the development data, given gold standard parse trees.

First, motivated by the close connection between WSD and our task (see Section 3), we conducted an experiment to test the applicability of using a WSD engine. In addition to the experiments listed above, we also attempted to encode the output of a modern WSD engine (the VBCollocations Model of SenseLearner 2.0 (Mihalcea and Csomai, 2005)), both by encoding the synset (if exists) of the verb instance as a feature, and by encoding each possible mapped class of the WSD engine output synset as a feature. There are k

[12] i.e., including instances tagged by SemLink as 'none'.

		Main Scenario				'Other is Possible' (OIP) Scenario			
		WSJ→WSJ		WSJ→GENIA		WSJ→WSJ		WSJ→GENIA	
		MF	Model	MF	Model	MF	Model	MF	Model
Gold Std	Total	93.68	**96.42**	69.09	**73.16**	88.6	**92.78**	46.41	**52.46**
	ER		43.35		13.17		36.67		11.29
	Poly.	88.87	**93.68**	48.58	**55.35**	–	–	–	–
	ER		43.22		13.17		–		–
Parser	Total	93.68	**95.9**	69.09	**72.4**	88.6	**91.9**	46.41	**52.46**
	ER		35.13		10.71		28.95		11.29
	Poly.	88.87	**92.77**	48.58	**55.35**	–	–	–	–
	ER		35.04		10.72		–		–

Table 1: Accuracy and error reduction (ER) results (in percents) for our model and the MF baseline. Error reduction is computed as $\frac{MODEL-MF}{100-MF}$. Results are given for the WSJ and GENIA corpora test sets. The top table is for a model receiving gold standard parses of the test data. The bottom is for a model using (Charniak and Johnson, 2005) state-of-the-art parses of the test data. In the main scenario (left), instances were always mapped to VN classes, while in the OIP one (right) it was possible (during both training and test) to map instances as not belonging to any existing class. For the latter, no results are displayed for polysemous verbs, since each verb can be mapped both to 'other' and to at least one class.

features if there are k possible classes[13]. There was no improvement over the previous model. A possible reason for this is the performance of the WSD engine (e.g. 56.1% precision on the verbs in Senseval-3 all-words task data). Naturally, more research is needed to establish better methods of incorporating WSD information to assist in this task.

Second, we studied the relative usability of class information as opposed to verb idiosyncratic information in the VN disambiguation task. By measuring the accuracy of our model, first given the per-class features (the first set of features excluding the verb's lemma feature) and second given the per-verb features (the conjunction of the first set with the verb's lemma), we tried to address this question. We obtained 94.82% accuracy for the per-class experiment, and 95.51% for the per-verb experiment, compared to 95.95% when using both in the in-domain gold standard scenario. The MF baseline scored 92.45% on this development set. These results, which are close in the per-class experiment to those of the MF baseline, indicate that combining both approaches in the construction of the classifier is justified.

Third, we studied the importance of having a learning algorithm utilizing the task's structure (mapping into a large label space where each instance can be mapped to only a small subspace). Our choice of the algorithm in (Even-Zohar and Roth, 2001) was done in light of this requirement. We conducted an experiment in which we omitted these per-instance restrictions on the label space, effectively allowing each verb to take every label in the label space. We obtained 94.54% accuracy, which translates to 27.68% error reduction, compared to 95.95% accuracy (46.36% error reduction) when using the restrictions. These results indicate that although our feature set keeps us substantially above baseline even without the above algorithm, using it boosts our results even further. This result is different from the results obtained in (Girju et al., 2005), where the results of the unconstrained (flat) model were significantly below baseline.

As noted earlier, the field of instance level verb classification into Levin-inspired classes is far from being exhaustively explored. We intend to make our implementation of the model available to the community, to enable others to engage in further research on this task.

Acknowledgements. We would like to thank Dan Roth, Mark Sammons and Ran Luria for their help.

References

Collin F. Baker, Charles J. Fillmore and John B. Lowe, 1998. *The Berkeley FrameNet Project. Proc. of the 36th Meeting of the ACL and the 17th COLING.*

Eugene Charniak and Mark Johnson, 2005. *Coarse-*

[13] The mapping is many-to-many and partial. To overcome the first issue, given a WN sense of the verb, we encoded all possible VN classes that correspond to it. To overcome the second, we treated a verb in a certain VN class, for which the mapping to WN was available, as one that can be mapped to all WN senses of the verb.

to-fine n-best parsing and maxent discriminative reranking. Proc. of the 43rd Meeting of the ACL.

Michael Collins, 1999. *Head-driven statistical models for natural language parsing. Ph.D. thesis, University of Pennsylvania.*

Hoa Trang Dang, Karin Kipper, Martha Palmer and Joseph Rosenzweig, 1998. *Investigating regular sense extensions based on intersective Levin classes. Proc. of the 36th Meeting of the ACL and the 17th COLING.*

Bonnie J. Dorr, 1997. *Large-Scale Dictionary Construction for Foreign Language Tutoring and Interlingual Machine Translation. Machine Translation,* 12:1-55.

Bonnie J. Dorr and Douglas Jones, 1996. *Role of Word Sense Disambiguation in Lexical Acquisition: Predicting Semantics from Syntactic Cues. Proc. of the 16th COLING.*

Yair Even-Zohar and Dan Roth, 2001. *A Sequential Model for Multi-Class Classification. Proc. of the 2001 Conference on Empirical Methods in Natural Language Processing.*

Roxana Girju, Dan Roth and Mark Sammons, 2005. *Token-level Disambiguation of VerbNet classes. The Interdisciplinary Workshop on Verb Features and Verb Classes.*

Svetlana Hensman and John Dunnion, 2004. *Automatically building conceptual graphs using VerbNet and WordNet. International Symposium on Information and Communication Technologies (ISICT).*

Jin–Dong Kim, Tomoko Ohta, Yuka Teteisi and Jun'ichi Tsujii, 2003. *GENIA corpus – a semantically annotated corpus for bio-textmining. Bioinformatics, 19:i180–i182, Oxford U. Press 2003.*

Karin Kipper, Hoa Trang Dang and Martha Palmer, 2000. *Class-Based Construction of a Verb Lexicon. Proc. of the 17th National Conference on Artificial Intelligence.*

Karin Kipper-Schuler, 2005. *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon. Ph. D. thesis, University of Pennsylvania.*

Karin Kipper, Anna Korhonen, Neville Ryant and Martha Palmer, 2006. *Extending VerbNet with Novel Verb Classes. Proc. of the 5th International Conference on Language Resources and Evaluation.*

Judith Klavans and Min-Yen Kan, 1998. *Role of verbs in document analysis. Proc. of the 36th Meeting of the ACL and the 17th International Conference on Computational Linguistics.*

Anna Korhonen and Ted Briscoe, 2004. *Extended Lexical-Semantic Classification of English Verbs. Proc. of the 42nd Meeting of the ACL, Workshop on Computational Lexical Semantics.*

Mirella Lapata and Chris Brew, 2004. *Verb Class Disambiguation using Informative Priors. Computational Linguistics,* 30(1):45-73

Matthew Lease and Eugene Charniak, 2005. *Towards a Syntactic Account of Punctuation. Proc. of the 2nd International Joint Conference on Natural Language Processing.*

Beth Levin, 1993. *English Verb Classes And Alternations: A Preliminary Investigation. The University of Chicago Press.*

Beth Levin and Malka Rappaport Hovav, 2005. *Argument Realization. Cambridge University Press.*

Juanguo Li and Chris Brew, 2007. *Disambiguating Levin Verbs Using Untagged Data. Proc. of the 2007 International Conference on Recent Advances in Natural Language Processing.*

Edward Loper, Szu-ting Yi and Martha Palmer, 2007. *Combining Lexical Resources: Mapping Between PropBank and VerbNet. Proc. of the 7th International Workshop on Computational Linguistics, Tilburg, the Netherlands.*

Paola Merlo and Suzanne Stevenson. 2001. *Automatic Verb-Classification Based On Statistical Distribution Of Argument Structure. Computational Linguistics,* 27(3):373–408.

Rada Mihalcea and Andras Csomai 2005. *Sense-Learner: word sense disambiguation for all words in unrestricted text. Proc. of the 43rd Meeting of the ACL , Poster Session.*

Martha Palmer, Daniel Gildea and Paul Kingsbury, 2005. *The proposition bank: A corpus annotated with semantic roles. Computational Linguistics,* 31(1).

Dan Roth, 1998. *Learning to resolve natural language ambiguities: A unified approach. Proc. of the 15th National Conference on Artificial Intelligence*

Sabine Schulte im Walde, 2000. *Clustering verbs semantically according to their alternation behavior. Proc. of the 18th COLING.*

Lei Shi and Rada Mihalcea, 2005. *Putting pieces together: Combining FrameNet, VerbNet and WordNet for robust semantic parsing. Proc. of the International Conference on Intelligent Text Processing and Computational Linguistics.*

Robert S. Swier and Suzanne Stevenson, 2005. *Exploiting a Verb Lexicon in Automatic Semantic Role Labelling. Proc. of the 2005 conference on empirical methods in natural language processing.*

Szu-ting Yi, Edward Loper and Martha Palmer, 2007. *Can Semantic Roles Generalize Across Genres? Proc. of the 2007 conference of the north american chapter of the association for computational linguistics.*