Which Are the Best Features for Automatic Verb Classification

Jianguo Li
Department of Linguistics
The Ohio State University
Columbus Ohio, USA
jianguo@ling.ohio-state.edu

Chris Brew
Department of Linguistics
The Ohio State University
Columbus Ohio, USA
cbrew@ling.ohio-state.edu

Abstract

In this work, we develop and evaluate a wide range of feature spaces for deriving Levin-style verb classifications (Levin, 1993). We perform the classification experiments using Bayesian Multinomial Regression (an efficient log-linear modeling framework which we found to outperform SVMs for this task) with the proposed feature spaces. Our experiments suggest that subcategorization frames are not the most effective features for automatic verb classification. A mixture of syntactic information and lexical information works best for this task.

1 Introduction

Much research in lexical acquisition of verbs has concentrated on the relation between verbs and their argument frames. Many scholars hypothesize that the behavior of a verb, particularly with respect to the expression of arguments and the assignment of semantic roles is to a large extent driven by deep semantic regularities (Dowty, 1991; Green, 1974; Goldberg, 1995; Levin, 1993). Thus measurements of verb frame patterns can perhaps be used to probe for linguistically relevant aspects of verb meanings. The correspondence between meaning regularities and syntax has been extensively studied in Levin (1993) (hereafter Levin). Levin's verb classes are based on the ability of a verb to occur or not occur in pairs of syntactic frames that are in some sense meaning preserving (*diathesis alternation*). The focus is on verbs for which distribution of syntactic frames is a useful indicator of class membership, and, correspondingly, on classes which are relevant for such verbs. By using Levin's classification, we obtain a window on some (but not all) of the potentially useful semantic properties of verbs.

Levin's verb classification, like others, helps reduce redundancy in verb descriptions and enables generalizations across semantically similar verbs with respect to their usage. When the information about a verb type is not available or sufficient for us to draw firm conclusions about its usage, the information about the class to which the verb type belongs can compensate for it, addressing the pervasive problem of data sparsity in a wide range of NLP tasks, such as automatic extraction of subcategorization frames (Korhonen, 2002), semantic role labeling (Swier and Stevenson, 2004; Gildea and Jurafsky, 2002), natural language generation for machine translation (Habash et al., 2003), and deriving predominant verb senses from unlabeled data (Lapata and Brew, 2004).

Although there exist several manually-created verb lexicons or ontologies, including Levin's verb taxonomy, VerbNet, and FrameNet, automatic verb classification (AVC) is still necessary for extending existing lexicons (Korhonen and Briscoe, 2004), building and tuning lexical information specific to different domains (Korhonen et al., 2006), and bootstrapping verb lexicons for new languages (Tsang et al., 2002).

AVC helps avoid the expensive hand-coding of such information, but appropriate features must be identified and demonstrated to be effective. In this work, our primary goal is not necessarily to obtain the optimal classification, but rather to investigate

Proceedings of ACL-08: HLT, pages 434–442,
Columbus, Ohio, USA, June 2008.

the linguistic conditions which are crucial for lexical semantic classification of verbs. We develop feature sets that combine syntactic and lexical information, which are in principle useful for any Levin-style verb classification. We test the general applicability and scalability of each feature set to the distinctions among 48 verb classes involving 1,300 verbs, which is, to our knowledge, the largest investigation on English verb classification by far. To preview our results, a feature set that combines both syntactic information and lexical information works much better than either of them used alone. In addition, mixed feature sets also show potential for scaling well when dealing with larger number of verbs and verb classes. In contrast, subcategorization frames, at least on their own, are largely ineffective for AVC, despite their evident effectiveness in supporting Levin's initial intuitions.

2 Related Work

Earlier work on verb classification has generally adopted one of the two approaches for devising statistical, corpus-based features.

Subcategorization frame (SCF): Subcategorization frames are obviously relevant to alternation behaviors. It is therefore unsurprising that much work on verb classification has adopted them as features (Schulte im Walde, 2000; Brew and Schulte im Walde, 2002; Korhonen et al., 2003). However, relying solely on subcategorization frames also leads to the loss of semantic distinctions. Consider the frame NP-V-PP*with*. The semantic interpretation of this frame depends to a large extent on the NP argument selected by the preposition *with*. In (1), the same surface form NP-V-PP*with* corresponds to three different underlying meanings. However, such semantic distinctions are totally lost if lexical information is disregarded.

(1) a. I ate with *a fork*. [INSTRUMENT]
b. I left with *a friend*. [ACCOMPANIMENT]
c. I sang with *confidence*. [MANNER]

This deficiency of unlexicalized subcategorization frames leads researchers to make attempts to incorporate lexical information into the feature representation. One possible improvement over subcategorization frames is to enrich them with lexical information. Lexicalized frames are usually obtained by augmenting each syntactic slot with its head noun (2).

(2) a. NP(*I*)-V-PP(*with:fork*)
b. NP(*I*)-V-PP(*with:friend*)
c. NP(*I*)-V-PP(*with:confidence*)

With the potentially improved discriminatory power also comes increased exposure to sparse data problems. Trying to overcome the problem of data sparsity, Schulte im Walde (2000) explores the additional use of selectional preference features by augmenting each syntactic slot with the concept to which its head noun belongs in an ontology (e.g. WordNet). Although the problem of data sparsity is alleviated to certain extent (3), these features do not generally improve classification performance (Schulte im Walde, 2000; Joanis, 2002).

(3) a. NP(*PERSON*)-V-PP(*with:ARTIFACT*)
b. NP(*PERSON*)-V-PP(*with:PERSON*)
c. NP(*PERSON*)-V-PP(*with:FEELING*)

JOANIS07: Incorporating lexical information directly into subcategorization frames has proved inadequate for AVC. Other methods for combining syntactic information with lexical information have also been attempted (Merlo and Stevenson, 2001; Joanis et al., 2007). These studies use a small collection of features that require some degree of expert linguistic analysis to devise. The deeper linguistic analysis allows their feature set to cover a variety of indicators of verb semantics, beyond that of frame information. Joanis et al. (2007) reports an experiment that involves 15 Levin verb classes. They define a general feature space that is supposed to be applicable to all Levin classes. The features they use fall into four different groups: *syntactic slots*, *slot overlaps*, *tense, voice and aspect*, and *animacy of NPs*.

- *Syntactic slots*: They encode the frequency of the syntactic positions (e.g. SUBJECT, OBJECT, PP*at*). They are considered approximation to subcategorization frames.

- *Slot overlaps*: They are supposed to capture the properties of alternation by identifying if a given noun can occur in different syntactic positions relative to a particular verb. For instance, in the alternation *The ice melted* and

The sun melted the ice, *ice* occurs in the subject position in the first sentence but in the object position in the second sentence. An overlap feature records that there is a subject-object alternation for *melt*.

- *Tense, voice and aspect*: Verb meaning and alternations also interact in interesting ways with *tense*, *voice*, and *aspect*. For example, *middle* construction is usually used in present tense (e.g. *The bread cuts easily*).

- *Animacy of NPs*: The animacy of the semantic role corresponding to the head noun in each syntactic slot can also distinguish classes of verbs.

Joanis et al. (2007) demonstrates that the general feature space they devise achieves a rate of error reduction ranging from 48% to 88% over a chance baseline accuracy, across classification tasks of varying difficulty. However, they also show that their general feature space does not generally improve the classification accuracy over subcategorization frames (see table 1).

Experimental Task	All Features	SCF
Average 2-way	83.2	80.4
Average 3-way	69.6	69.4
Average ($\geq$ 6)-way	61.1	62.8

Table 1: Results from Joanis et al. (2007) (%)

3 Integration of Syntactic and Lexical Information

In this study, we explore a wider range of features for AVC, focusing particularly on various ways to mix syntactic with lexical information.

Dependency relation (DR): Our way to overcome data sparsity is to break lexicalized frames into *lexicalized slots* (a.k.a. dependency relations). Dependency relations contain both syntactic and lexical information (4).

(4) a. SUBJ(*I*), PP(*with:fork*)
b. SUBJ(*I*), PP(*with:friend*)
c. SUBJ(*I*), PP(*with:confidence*)

However, augmenting PP with nouns selected by the preposition (e.g. PP(*with:fork*)) still gives rise to data sparsity. We therefore decide to break it into two individual dependency relations: PP(*with*), PP-*fork*. Although dependency relations have been widely used in automatic acquisition of lexical information, such as detection of polysemy (Lin, 1998) and WSD (McCarthy et al., 2004), their utility in AVC still remains untested.

Co-occurrence (CO): CO features mostly convey lexical information only and are generally considered not particularly sensitive to argument structures (Rohde et al., 2004). Nevertheless, it is worthwhile testing whether the meaning components that are brought out by syntactic alternations are also correlated to the neighboring words. In other words, Levin verbs may be distinguished on the dimension of neighboring words, in addition to argument structures. A test on this claim can help answer the question of whether verbs in the same Levin class also tend to share their neighboring words.

Adapted co-occurrence (ACO): Conventional CO features generally adopt a stop list to filter out function words. However, some of the functions words, prepositions in particular, are known to carry great amount of syntactic information that is related to lexical meanings of verbs (Schulte im Walde, 2003; Brew and Schulte im Walde, 2002; Joanis et al., 2007). In addition, whereas most verbs tend to put a strong selectional preference on their nominal arguments, they do not care much about the identity of the verbs in their verbal arguments. Based on these observations, we propose to adapt the conventional CO features by (1) keeping all prepositions (2) replacing all verbs in the neighboring contexts of each target verb with their part-of-speech tags. ACO features integrate at least some degree of syntactic information into the feature space.

SCF+CO: Another way to mix syntactic information with lexical information is to use subcategorization frames and co-occurrences together in hope that they are complementary to each other, and therefore yield better results for AVC.

4 Experiment Setup

4.1 Corpus

To collect each type of features, we use the Gigaword Corpus, which consists of samples of recent newswire text data collected from four distinct in-

ternational sources of English newswire.

4.2 Feature Extraction

We evaluate six different feature sets for their effectiveness in AVC: **SCF**, **DR**, **CO**, **ACO**, **SCF+CO**, and **JOANIS07**. **SCF** contains mainly syntactic information, whereas **CO** lexical information. The other four feature sets include both syntactic and lexical information.

SCF and **DR**: These more linguistically informed features are constructed based on the grammatical relations generated by the C&C CCG parser (Clark and Curran, 2007). Take *He broke the door with a hammer* as an example. The grammatical relations generated are given in table 2.

he broke the door with a hammer.
(det door_3 the_2)
(dobj _ broke_1 door_3)
(det hammer_6 a_5)
(dobj with_4 hammer_6)
(iobj broke_1 with_4)
(ncsubj broke_1 He_0 _)

Table 2: grammatical relations generated by the parser

We first build a lexicalized frame for the verb *break*: NP1(*he*)-V-NP2(*door*)-PP(*with*:*hammer*). This is done by matching each grammatical label onto one of the traditional syntactic constituents. The set of syntactic constituents we use is summarized in table 3.

constituent	remark
NP1	subject of the verb
NP2	object of the verb
NP3	indirect object of the verb
PP*p*	prepositional phrase
TO	infinitival clause
GER	gerund
THAT	sentential complement headed by *that*
WH	sentential complement headed by a *wh*-word
ADJP	adjective phrase
ADVP	adverb phrase

Table 3: Syntactic constituents used for building SCFs

Based on the lexicalized frame, we construct an SCF NP1-NP2-PP*with* for *break*. The set of DRs generated for *break* is [SUBJ(*he*), OBJ(*door*), PP(*with*), PP-*hammer*].

CO: These features are collected using a flat 4-word window, meaning that the 4 words to the left/right of each target verb are considered potential CO features. However, we eliminate any CO features that are in a stopword list, which consists of about 200 closed class words including mainly prepositions, determiners, complementizers and punctuation. We also lemmatize each word using the English lemmatizer as described in Minnen et al. (2000), and use lemmas as features instead of words.

ACO: As mentioned before, we adapt the conventional CO features by (1) keeping all prepositions (2) replacing all verbs in the neighboring contexts of each target verb with their part-of-speech tags. (3) keeping words in the left window only if they are tagged as a nominal.

SCF+CO: We combine the SCF and CO features.

JOANIS07: We use the feature set proposed in Joanis et al. (2007), which consists of 224 features. We extract features on the basis of the output generated by the C&C CCG parser.

4.3 Verb Classes

Our experiments involve two separate sets of verb classes:

Joanis15: Joanis et al. (2007) manually selects pairs, or triples of classes to represent a range of distinctions that exist among the 15 classes they investigate. For example, some of the pairs/triples are syntactically dissimilar, while others show little syntactic distinction across the classes.

Levin48: Earlier work has focused only on a small set of verbs or a small number of verb classes. For example, Schulte im Walde (2000) uses 153 verbs in 30 classes, and Joanis et al. (2007) takes on 835 verbs and 15 verb classes. Since one of our primary goals is to identify a general feature space that is not specific to any class distinctions, it is of great importance to understand how the classification accuracy is affected when attempting to classify more verbs into a larger number of classes. In our automatic verb classification, we aim for a larger scale experiment. We select our experimental verb classes and verbs as follows: We start with all Levin 197 verb classes. We first remove all verbs that belong to at least two Levin classes. Next, we remove any verb that does not occur at least 100 times in the English Gigaword Corpus. All classes that are left with at least 10 verbs are chosen for our experi-

ment. This process yields 48 classes involving about 1,300 verbs. In our automatic verb classification experiment, we test the applicability of each feature set to distinctions among up to 48 classes [1]. To our knowledge, this is, by far, the largest investigation on English verb classification.

5 Machine Learning Method

5.1 Preprocessing Data

We represent the semantic space for verbs as a matrix of frequencies, where each row corresponds to a Levin verb and each column represents a given feature. We construct a semantic space with each feature set. Except for **JONAIS07** which only contains 224 features, all the other feature sets lead to a very high-dimensional space. For instance, the semantic space with **CO** features contains over one million columns, which is too huge and cumbersome. One way to avoid these high-dimensional spaces is to assume that most of the features are irrelevant, an assumption adopted by many of the previous studies working with high-dimensional semantic spaces (Burgess and Lund, 1997; Pado and Lapata, 2007; Rohde et al., 2004). Burgess and Lund (1997) suggests that the semantic space can be reduced by keeping only the *k* columns (features) with the highest variance. However, Rohde et al. (2004) have found it is simpler and more effective to discard columns on the basis of feature frequency, with little degradation in performance, and often some improvement. Columns representing low-frequency features tend to be noisier because they only involve few examples. We therefore apply a simple frequency cutoff for feature selection. We only use features that occur with a frequency over some threshold in our data.

In order to reduce undue influence of outlier features, we employ the four normalization strategies in table 4, which help reduce the range of extreme values while having little effect on others (Rohde et al., 2004). The raw frequency ($w_{v,f}$) of a verb *v* occurring with a feature *f* is replaced with the normalized value ($w'_{v,f}$), according to each normalization method. Our experiments show that using correlation for normalization generally renders the best results. The results reported below are obtained from using correlation for normalization.

	$w'_{v,f} =$
row	$\frac{w_{v,f}}{\sum_j w_{v,j}}$
column	$\frac{w_{v,f}}{\sum_i w_{i,f}}$
length	$\frac{w_{v,f}}{\sum_j w_{v,j}^2{}^{1/2}}$
correlation	$\frac{T w_{v,f} - \sum_j w_{v,j} \sum_i w_{i,f}}{(\sum_j w_{v,j}(T - \sum_j w_{v,j}) \sum_i w_{i,f}(T - \sum_i w_{i,f}))^{1/2}}$
	$T = \sum_i \sum_j w_{i,j}$

Table 4: Normalization techniques

To preprocess data, we first apply a frequency cutoff to our data set, and then normalize it using the correlation method. To find the optimal threshold for frequency cut, we consider each value between 0 and 10,000 at an interval of 500. In our experiments, results on training data show that performance declines more noticeably when the threshold is lower than 500 or higher than 10,000. For each task and feature set, we select the frequency cut that offers the best accuracy on the preprocessed training set according to *k*-fold stratified cross validation [2].

5.2 Classifier

For all of our experiments, we use the software that implements the Bayesian multinomial logistic regression (a.k.a BMR). The software performs the so-called 1-of-*k* classification (Madigan et al., 2005). BMR is similar to Maximum Entropy. It has been shown to be very efficient with handling large numbers of features and extremely sparsely populated matrices, which characterize the data we have for AVC [3]. To begin, let $x = [x_1, ..., x_j, ..., x_d]^T$ be a vector of feature values characterizing a verb to be classified. We encode the fact that a verb belongs to a class $k \in 1, ..., K$ by a *K*-dimensional 0/1 valued vector $y = (y_1, ..., y_K)^T$, where $y_k = 1$ and all other coordinates are 0. Multinomial logistic regres-

[1] In our experiment, we only use monosemous verbs from these 48 verb classes. Due to the space limit, we do not list the 48 verb classes. The size of the most classes falls in the range between 10 to 30, with a couple of classes having a size over 100.

[2] 10-fold for Joanis15 and 9-fold for Levin48. We use a balanced training set, which contains 20 verbs from each class in Joanis15, but only 9 verbs from each class in Levin48.

[3] We also tried Chang and Lin (2001)'s LIBSVM library for Support Vector Machines (SVMs), however, BMR generally outperforms SVMs.

sion is a conditional probability model of the form, parameterized by the matrix $\beta = [\beta_1, ..., \beta_K]$. Each column of β is a parameter vector corresponding to one of the classes: $\beta_k = [\beta_{k1}, ..., \beta_{kd}]^T$.

$$P(y_k = 1|\beta_k, x) = exp(\beta_k^T x) / \sum_{k_i} exp(\beta_{k_i}^T x)$$

6 Results and Discussion

6.1 Evaluation Metrics

Following Joanis et al. (2007), we adopt a single evaluation measure - macro-averaged recall - for all of our classification tasks. As discussed below, since we always use balanced training sets for each individual task, it makes sense for our accuracy metric to give equal weight to each class. Macro-averaged recall treats each verb class equally, so that the size of a class does not affect macro-averaged recall. It usually gives a better sense of the quality of classification across all classes. To calculate macro-averaged recall, the recall value for each individual verb class has to be computed first.

$$recall = \frac{no.\ of\ test\ verbs\ in\ class\ c\ correctly\ labeled}{no.\ of\ test\ verbs\ in\ class\ c}$$

With a recall value computed for each verb class, the macro-averaged recall can be defined by:

$$macro\text{-}averaged\ recall = \frac{1}{|C|} \sum_{c \in C} recall\ for\ class\ c$$

$$C : a\ set\ of\ verb\ classes$$

$$c : an\ individual\ verb\ class$$

$$|C| : the\ number\ of\ verb\ classes$$

6.2 Joanis15

With those manually-selected 15 classes, Joanis et al. (2007) conducts 11 classification tasks including six 2-way classifications, two 3-way classifications, one 6-way classification, one 8-way classification, and one 14-way classification. In our experiments, we replicate these 11 classification tasks using the proposed six different feature sets. For each classification task in this task set, we randomly select 20 verbs from each class as the training set. We repeat this process 10 times for each task. The results reported for each task is obtained by averaging the results of the 10 trials. Note that for each trial, each feature set is trained and tested on the same training/test split.

The results for the 11 classification tasks are summarized in table 5. We provide a chance baseline and the accuracy reported in Joanis et al. (2007) [4] for comparison of our results. A few points are worth noting:

- Although widely used for AVC, **SCF**, at least when used alone, is not the most effective feature set. Our experiments show that the performance achieved by using **SCF** is generally worse than using the feature sets that mix syntactic and lexical information. As a matter of fact, it even loses to the simplest feature set **CO** on 4 tasks, including the 14-way task.

- The two feature sets (**DR**, **SCF+CO**) we propose that combine syntactic and lexical information generally perform better than those feature sets (**SCF**, **CO**) that only include syntactic or lexical information. Although there is not a clear winner, **DR** and **SCF+CO** generally outperform other feature sets, indicating that they are effective ways for combining syntactic and lexical information. In particular, these two feature sets perform comparatively well on the tasks that involve more classes (e.g. 14-way), exhibiting the tendency to scale well with larger number of verb classes and verbs. Another feature set that combines syntactic and lexical information, **ACO**, which keeps function words in the feature space to preserve syntactic information, outperforms the conventional **CO** on the majority of tasks. All these observations suggest that how to mix syntactic and lexical information is one of keys to an improved verb classification.

- Although **JOANIS07** also combines syntactic and lexical information, its performance is not comparable to that of other feature sets that mix syntactic and lexical information. In fact, **SCF**

[4] Joanis et al. (2007) is different from our experiments in that they use a chunker for feature extraction and the Support Vector Machine for classification.

Experimental Task	Random Baseline	As Reported in Joanis et al. (2007)	Feature Set					
			SCF	DR	CO	ACO	SCF+CO	JOANIS07
1) Benefactive/Recipient	50	86.4	88.6	88.4	88.2	89.1	**90.7**	88.9
2) Admire/Amuse	50	93.9	96.7	**97.5**	92.1	90.5	96.4	96.6
3) Run/Sound	50	86.8	85.4	89.6	**91.8**	90.2	90.5	87.1
4) Light/Sound	50	75.0	74.8	**90.8**	86.9	89.7	88.8	82.1
5) Cheat/Steal	50	76.5	77.6	**80.6**	72.1	75.5	77.8	76.4
6) Wipe/Steal	50	80.4	**84.8**	80.6	79.0	79.4	84.4	83.9
7) Spray/Fill/Putting	33.3	65.6	73.0	72.8	59.6	66.6	**73.8**	69.6
8) Run/State Change/Object drop	33.3	74.2	74.8	77.2	76.9	77.6	**80.5**	75.5
9) Cheat/Steal/Wipe/Spray/Fill/Putting	16.7	64.3	64.9	**65.1**	54.8	59.1	65.0	64.3
10) 9)/Run/Sound	12.5	61.7	62.3	65.8	55.7	60.8	**66.9**	63.1
11) 14-way (all except Benefactive)	7.1	58.4	56.4	65.7	57.5	59.6	**66.3**	57.2

Table 5: Experimental results for Joanis15 (%)

and **JOANIS07** yield similar accuracy in our experiments, which agrees with the findings in Joanis et al. (2007) (compare table 1 and 5).

6.3 Levin48

Recall that one of our primary goals is to identify the feature set that is generally applicable and scales well while we attempt to classify more verbs into a larger number of classes. If we could exhaust all the possible *n*-way ($2 \leq n \leq 48$) classification tasks with the 48 Levin classes we will investigate, it will allow us to draw a firmer conclusion about the general applicability and scalability of a particular feature set. However, the number of classification tasks grows really huge when *n* takes on certain value (e.g. $n = 20$). For our experiments, we set *n* to be 2, 5, 10, 20, 30, 40, or 48. For the *2*-way classification, we perform all the possible 1,028 tasks. For the 48-way classification, there is only one possible task. We randomly select 100 *n*-way tasks each for *n* = 5, 10, 20, 30, 40. We believe that this series of tasks will give us a reasonably good idea of whether a particular feature set is generally applicable and scales well.

The smallest classes in Levin48 have only 10 verbs. We therefore reduce the number of training verbs to 9 for each class. For each *n* = 2, 5, 10, 20, 30, 40, 48, we will perform certain number of *n*-way classification tasks. For each *n*-way task, we randomly select 9 verbs from each class as training data, and repeat this process 10 times. The accuracy for each *n*-way task is then computed by averaging the results from these 10 trials. The accuracy reported for the overall *n*-way classification for each selected *n*, is obtained by averaging the results from each individual *n*-way task for that particular *n*. Again, for each trial, each feature set is trained and tested on the same training/test split.

The results for Levin48 are presented in table 6, which clearly reveals the general applicability and scalability of each feature set.

- Results from Levin48 reconfirm our finding that **SCF** is not the most effective feature set for AVC. Although it achieves the highest accuracy on the 2-way classification, its accuracy drops drastically as *n* gets bigger, indicating that SCF does not scale as well as other feature sets when dealing with larger number of verb classes. On the other hand, the co-occurrence feature (**CO**), which is believed to convey only lexical information, outperforms **SCF** on every *n*-way classification when $n \geq 10$, suggesting that verbs in the same Levin classes tend to share their neighboring words.

- The three feature sets we propose that combine syntactic and lexical information generally scale well. Again, **DR** and **SCF+CO** generally outperform all other feature sets on all *n*-way classifications, except the 2-way classification. In addition, **ACO** achieves a better performance on every *n*-way classification than **CO**. Although **SCF** and **CO** are not very effective when used individually, they tend to yield the best performance when combined together.

- Again, **JOANIS07** does not match the performance of other feature sets that combine both syntactic and lexical information, but yields similar accuracy as **SCF**.

Experimental Task	No of Tasks	Random Baseline	Feature Set					
			SCF	DR	CO	ACO	SCF+CO	JOANIS07
2-way	1,028	50	**84.0**	83.4	77.8	80.9	82.9	82.4
5-way	100	20	71.9	76.4	70.4	73.0	**77.3**	72.2
10-way	100	10	65.8	**73.7**	68.8	71.2	72.8	65.9
20-way	100	5	51.4	65.1	58.8	60.1	**65.8**	50.7
30-way	100	3.3	46.7	56.9	48.6	51.8	**57.8**	47.1
40-way	100	2.5	43.6	54.8	47.3	49.9	**55.1**	44.2
48-way	1	2.2	39.1	51.6	42.4	46.8	**52.8**	38.9

Table 6: Experimental results for Levin48 (%)

6.4 Further Discussion

Previous studies on AVC have focused on using SCFs. Our experiments reveal that SCFs, at least when used alone, compare poorly to the feature sets that mix syntactic and lexical information. One explanation for the poor performance could be that we use all the frames generated by the CCG parser in our experiment. A better way of doing this would be to use some expert-selected SCF set. Levin classifies English verbs on the basis of 78 SCFs, which should, at least in principle, be good at separating verb classes. To see if Levin-selected SCFs are more effective for AVC, we match each SCF generated by the C&C CCG parser (**CCG-SCF**) to one of 78 Levin-defined SCFs, and refer to the resulting SCF set as **unfiltered-Levin-SCF**. Following studies on automatic SCF extraction (Brent, 1993), we apply a statistical test (Binomial Hypothesis Test) to the **unfiltered-Levin-SCF** to filter out noisy SCFs, and denote the resulting SCF set as **filtered-Levin-SCF**. We then perform the 48-way task (one of Levin48) with these two different SCF sets. Recall that using **CCG-SCF** gives us a macro-averaged recall of 39.1% on the 48-way task. Our experiments show that using **unfiltered-Levin-SCF** and **filtered-Levin-SCF** raises the accuracy to 39.7% and 40.3% respectively. Although a little performance gain has been obtained by using expert-defined SCFs, the accuracy level is still far below that achieved by using a feature set that combines syntactic and semantic information. In fact, even the simple co-occurrence feature (CO) yields a better performance (42.4%) than these Levin-selected SCF sets.

7 Conclusion and Future Work

We have performed a wide range of experiments to identify which features are most informative in AVC. Our conclusion is that both syntactic and lexical information are useful for verb classification. Although neither **SCF** nor **CO** performs well on its own, a combination of them proves to be the most informative feature for this task. Other ways of mixing syntactic and lexical information, such as **DR**, and **ACO**, work relatively well too. What makes these mixed feature sets even more appealing is that they tend to scale well in comparison to **SCF** and **CO**. In addition, these feature sets are devised on a general level without relying on any knowledge about specific classes, thus potentially applicable to a wider range of class distinctions. Assuming that Levin's analysis is generally applicable across languages in terms of the linking of semantic arguments to their syntactic expressions, these mixed feature sets are potentially useful for building verb classifications for other languages.

For our future work, we aim to test whether an automatically created verb classification can be beneficial to other NLP tasks. One potential application of our verb classification is parsing. Lexicalized PCFGs (where head words annotate phrasal nodes) have proved a key tool for high performance PCFG parsing, however its performance is hampered by the sparse lexical dependency exhibited in the Penn Treebank. Our experiments on verb classification have offered a class-based approach to alleviate data sparsity problem in parsing. It is our goal to test whether this class-based approach will lead to an improved parsing performance.

8 Acknowledgments

This study was supported by NSF grant 0347799. We are grateful to Eric Fosler-Lussier, Detmar Meurers, Mike White and Kirk Baker for their valuable comments.

References

Brent, M. (1993). From grammar to lexicon: Unsupervised learning of lexical syntax. *Computational Linguistics*, 19(3):243–262.

Brew, C. and Schulte im Walde, S. (2002). Spectral clustering for German verbs. In *Proccedings of the 2002 Conference on EMNLP*, pages 117–124.

Burgess, C. and Lund, K. (1997). Modelling parsing constraints with high-dimentional context space. *Language and Cognitive Processes*, 12(3):177–210.

Chang, C. and Lin, C. (2001). LIBSVM: A library for support vector machines. http://www.csie.ntu.edu.tw. cjlin/libsvm.

Clark, S. and Curran, J. (2007). Formalism-independent parser evaluation with CCG and Depbank. In *Proceedings of the 45th Annual Meeting of ACL*, pages 248–255.

Dowty, D. (1991). Thematic proto-roles and argument selection. *Language*, 67:547–619.

Gildea, D. and Jurafsky, D. (2002). Automatic labeling of semantic role. *Computational Linguistics*, 28(3):245–288.

Goldberg, A. (1995). *Constructions*. University of Chicago Press, Chicago, 1st edition.

Green, G. (1974). *Semantics and Syntactic Regularity*. Indiana University Press, Bloomington.

Habash, N., Dorr, B., and Traum, D. (2003). Hybrid natural language generation from lexical conceptual structures. *Machine Translation*, 18(2):81–128.

Joanis, E. (2002). Automatic verb classification using a general feature space. Master's thesis, University of Toronto.

Joanis, E., Stevenson, S., and James, D. (2007). A general feature space for automatic verb classification. *Natural Language Engineering*, 1:1–31.

Korhonen, A. (2002). *Subcategorization Acquisition*. PhD thesis, Cambridge University.

Korhonen, A. and Briscoe, T. (2004). Extended lexical-semantic classification of english verbs. In *Proceedings of the 2004 HLT/NAACL Workshop on Computational Lexical Semantics*, pages 38–45, Boston, MA.

Korhonen, A., Krymolowski, Y., and Collier, N. (2006). Automatic classification of verbs in biomedical texts. In *Proceedings of the 21st International Conference on COLING and 44th Annual Meeting of ACL*, pages 345–352, Sydney, Australia.

Korhonen, A., Krymolowski, Y., and Marx, Z. (2003). Clustering polysemic subcategorization frame distributions semantically. In *Proceedings of the 41st Annual Meeting of ACL*, pages 48–55, Sapparo, Japan.

Lapata, M. and Brew, C. (2004). Verb class disambiguation using informative priors. *Computational Linguistics*, 30(1):45–73.

Levin, B. (1993). *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press, Chicago, 1st edition.

Lin, D. (1998). Automatic retrieval and clustering of similar words. In *Proceedings of the 17th Internation Conference on COLING and 36th Annual Meeting of ACL*.

Madigan, D., Genkin, A., Lewis, D., and Fradkin, D. (2005). Bayesian Multinomial Logistic Regression for Author Identification. *DIMACS Technical Report*.

McCarthy, D., Koeling, R., Weeds, J., and Carroll, J. (2004). Finding predominant senses in untagged text. In *Proceedings of the 42nd Annual Meeting of ACL*, pages 280–287.

Merlo, P. and Stevenson, S. (2001). Automatic verb classification based on statistical distribution of argument structure. *Computational Linguistics*, 27(3):373–408.

Minnen, G., Carroll, J., and Pearce, D. (2000). Applied morphological processing of English. *Natural Language Engineering*, 7(3):207–223.

Pado, S. and Lapata, M. (2007). Dependency-based construction of semantic space models. *Computional Linguistics*, 33(2):161–199.

Rohde, D., Gonnerman, L., and Plaut, D. (2004). An improved method for deriving word meaning from lexical co-occurrence. http://dlt4.mit.edu/ dr/COALS.

Schulte im Walde, S. (2000). Clustering verbs semantically according to alternation behavior. In *Proceedings of the 18th International Conference on COLING*, pages 747–753.

Schulte im Walde, S. (2003). Experiments on the choice of features for learning verb classes. In *Proceedings of the 10th Conference of EACL*, pages 315–322.

Swier, R. and Stevenson, S. (2004). Unsupervised semantic role labelling. In *Proceedings of the 2004 Conference on EMNLP*, pages 95–102.

Tsang, V., Stevenson, S., and Merlo, P. (2002). Crosslinguistic transfer in automatic verb classification. In *Proceedings of the 19th International Conference on COLING*, pages 1023–1029, Taiwan, China.