

Which are the Best Features for Automatic Verb Classification: a Corpus Study of Levin Verb Classification

Jianguo Li
Department of Linguistics
The Ohio State University
Columbus, Ohio 43201

Kirk Baker
Department of Linguistics
The Ohio State University
Columbus, Ohio 43201

Chris Brew
Department of Linguistics
The Ohio State University
Columbus, Ohio 43201

August 23, 2011

Abstract

Much research in lexical acquisition has concentrated on the hypothesis 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. We focus on the empirical basis for a central part of this research, Levin (1993), (which we abbreviate as Levin from now on). The core of Levin's work is a set of claims about the connections between meaning regularities and syntactic alternation patterns. These connections form the basis for a detailed classification of a large number of English verbs. In computational linguistics, Levin's classification has been shown to reduce redundancy in verb descriptions and to enable potentially useful generalizations across similar verbs. This has motivated a wealth of work on automatic acquisition of verb lexicons from labeled and unlabeled corpora. This is a clear demonstration of the transfer of useful ideas from theoretical linguistics to computational linguistics. Until now, there has been little transfer in the opposite direction.

In the present case study, we investigate the relation between Levin's approach, the theoretical motivations for her work and the corpus data. Concretely, we wish to discover maximally informative and effective features for automatic verb classification. We develop and evaluate a wide range of different feature sets, evaluating each against Levin-style verb classification. We perform automatic classification experiments using Bayesian Multinomial Regression (a state-of-the-art log-linear modeling framework which we found to outperform SVMs for this task) with the proposed feature spaces. Our experiments suggest that subcategorization frames, despite their theoretical credibility, are not the most effective basis for automatic classifiers. Rather, we find that the best-performing classifier use a combination of dependency information and simple co-occurrence features. This observation is consistent with the hypothesis that robust and practically useful extensions of Levin's classes will be based not only on alternation patterns but also on semantic similarities.

1 Introduction

Improved automatic text understanding requires detailed linguistic information about the words that comprise the text. Particularly crucial is the knowledge about predicates, typically verbs, which communicate both the event being expressed and how participants are related to the event, i.e., *who did what to whom*. Many different approaches to lexicon development (Pustejovsky, 1991; Lowe et al., 1997; Dorr, 1997) have been proposed for construction of verb lexicons. Although the

field of natural language processing (NLP) has yet to develop a clear consensus on guidelines for building a verb lexicon that is suitable for applications in NLP, the class-based construction of verb lexicons with explicitly stated syntactic and semantic information has proved to be an appealing approach. Such class-based construction of verb lexicons typically benefits from a long-standing linguistic hypothesis that there is a tight connection between the lexical meaning of a verb and its behavior. To be exact, many scholars hypothesize that the behavior of a verb, in particular with respect to the expression and interpretation of its arguments, is to a large extent driven by deep semantic regularities (Dowty, 1991; Goldberg, 1995; Levin, 1993; Pinker, 1989; Green, 1974), and thus measurements of verb frame patterns can perhaps be used to probe for linguistically relevant aspects of verb meanings.

In computational linguistics, it is common for variations in the predictive features used for a classification task to have a greater impact on performance than do changes in the machine learning algorithms used. The focus of the present article is on the relative merit of the different possible features that might be used to reproduce a Levin-style taxonomy. To explore this we make systematic use of state-of-the-art classification methods, and compare the results across feature sets. The benefits of automation include the ability to compare many versions of the taxonomy, varying the granularity and difficulty of the tasks presented for the classifier. It would be very costly to do such a comparison if the classification had to be created by human annotators, since each annotator could be used on only one version of the taxonomy.

We are interpreting difference in classifier performance as informative about the merits of the feature sets. This is risky if a different classifier would yield different judgements about relative values of the feature sets. For most of the present work we use Bayesian Multinomial Regression with a Gaussian prior, but we also demonstrate that an alternate choice of state-of-the-art classifier does not alter the general conclusions.

A second potential concern is that automatic classifiers cannot distinguish between genuine causal relationships and mere correlations. Thus, if we find that features other than those mentioned by the theoreticians are more effective, we cannot conclude that the claimed causal connections are absent. The other features could be acting as proxies for the real causes, and the classifiers may be choosing the proxies for linguistically uninteresting engineering reasons, such as those associated with data sparsity.

1.1 Levin Verb Classification

Levin has extensively studied the correspondence between verbal meaning and syntax. Levin 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 the distribution of syntactic frames is a useful indicator of class membership, and correspondingly, on classes which are relevant for such verbs. Verbs in each Levin class are assumed to share both a common semantics and a set of syntactic alternations.

Levin defines 78 diathesis alternations, and classifies 3,104 English verbs into 49 verb classes - partly divided into 191 sub-classes according to alternations the respective verbs undergo. The set of syntactic frames associated with a particular Levin class are not intended to be arbitrary, but they are supposed to reflect underlying semantic components that constrain allowable arguments. Levin demonstrates this point about the nature of lexical knowledge with an example about four verbs: *break*, *cut*, *hit*, and *touch*.

The verbs *break*, *cut*, *hit*, and *touch* are all transitive, taking two arguments expressed as subject

and object. However, they differ from each other with respect to their participation in diathesis alternation. In fact, the following three diathesis alternations make a perfect distinction among these four verbs:

Middle Alternation: In the middle alternation, the object of the transitive frame ((a) examples) and the subject of the intransitive frame ((b) examples) have the same semantic relation to the verb. Only *cut* and *break* can participate in the middle alternation.

- (1) a. Margaret cut the bread.
b. The bread cuts easily.
- (2) a. Janet broke the vase.
b. Crystal vases break easily.
- (3) a. Terry touched the cat.
b. *Cats touch easily.
- (4) a. Carla hit the door.
b. *Door frames hit easily.

The middle alternation is semantically correlated with the notion of causing a change of state. The verbs *cut* and *break* usually denote actions that entail a change of state, but *hit* and *touch* do not.

Conative Alternation: Unlike the middle alternation, in the conative alternation, the subject of the transitive and intransitive frame bears the same semantic relation to the verb. Only *cut* and *hit* can appear in the conative construction.

- (5) a. Margaret cut the bread.
b. Margaret cut at the bread.
- (6) a. Janet broke the vase.
b. *Janet broke at the vase.
- (7) a. Terry touched the cat.
b. *Terry touched at the cat.
- (8) a. Carla hit the door.
b. Carla hit at the door.

The meaning component relevant to the conative construction is the notion of motion. Only *cut* and *hit* involve motion component. In addition, there is no entailment that the action denoted by the verb was completed. For example, *cut* describes a series of actions directed at achieving the goal of separating some objects into pieces. However, it is possible for these actions to be performed without the end result being achieved (*Margaret cut at the bread, but did not make a dent in it*). Where *break* is concerned, the only aspect specified is the resulting change of state where the object becomes separated into pieces. If the result is not achieved, there are no attempted *breaking* actions that can still be recognized (**Janet broke the vase, but the vase was not broken*).

Body-part Possessor Ascension Alternation: This alternation is characterized by a change in the expression of a possessed body part. The possessed body part is expressed as the direct object, as in the (a) examples. In (b) examples, the possessor is expressed as the object, with the possessed body part expressed in a prepositional phrase. This alternation distinguishes *cut*, *hit*, and *touch* from *break*.

- (9) a. Margaret cut Bill’s arm.
 b. Margaret cut Bill on the arm.
- (10) a. Janet broke Bill’s finger.
 b. *Janet broke Bill at the finger.
- (11) a. Terry touched Bill’s shoulder.
 b. Terry touched Bill on the shoulder.
- (12) a. Carla hit Bill’s back.
 b. Carla hit Bill on the back.

The body-part possessor ascension alternation is related to the notion of contact. The three verbs *cut*, *touch*, and *hit* denote actions that necessarily involve contact. Although real-world event denoted by the verb *break* often involves contact, *break* is a pure change of state verb, and a notion of contact is not inherent in its meaning.

It becomes self-evident now that each verb *touch*, *hit*, *cut*, and *break* shows a distinct pattern of behavior with respect to these three alternations, as summarized in Table (1).

| Alternation | Meaning Component | <i>touch</i> | <i>hit</i> | <i>cut</i> | <i>break</i> |
|-------------------------------|-------------------|--------------|------------|------------|--------------|
| Conative | change of state | NO | YES | YES | NO |
| Body-Part Possessor Ascension | motion | YES | YES | YES | NO |
| Middle | contact | NO | NO | YES | YES |

Table 1: Pattern of alternation behavior of *touch*, *hit*, *cut*, and *break*

The patterns observed in Table (1) cannot simply be dismissed because they are linked to four different verbs. Corresponding to each of these four verbs are many other verbs that show the same pattern of behavior, as shown in Table (2).

| Verb Class | Example Verbs |
|-------------|--|
| TOUCH verbs | <i>pat, stroke, tickle, touch, ...</i> |
| HIT verbs | <i>bask, hit, kick, pound, ...</i> |
| CUT verbs | <i>cut, hack, saw, scratch, ...</i> |
| BREAK verbs | <i>break, crack, rip, shatter, ...</i> |

Table 2: Verb classes based on *touch*, *hit*, *cut*, and *break*

Levin verb classification has also inspired the creation of many other verb lexicons. For example, it has been incorporated into the computational lexicon of VerbNet (Kipper et al., 2000), which has

recently been further extended (Kipper et al., 2006). Other computational verb lexicons include FrameNet (Baker et al., 1998) and PropBank (Palmer et al., 2005). However, VerbNet is the only one that strictly follows Levin in assigning verbs to classes based on their syntactic behavior. We therefore refer to Levin and VerbNet as Levin-style verb classifications. The lexical semantic classification of verbs constructed this way has been shown to be important in supporting a wide range of NLP tasks, including lexical resource construction (Korhonen, 2002), natural language generation for machine translation (Swift, 2005), semantic parsing (Shi and Mihalcea, 2005), semantic role labeling (Gildea and Jurafsky, 2002) and information retrieval (Klavans and Kan, 1998).

While Levin verb classification has proved useful in supporting various NLP tasks, to adopt Levin verb classification is also to accept some limitations on the reach of its utility and applicability. Some of its limitations reflect the explicit design decisions:

1. Levin verb classification relies primarily on straightforward syntactic criteria. However, not all semantically interesting differences will have the appropriate reflexes in syntax. For instance, Levin verb classification does not treat the inherent lexical aspect of verbs (*aktionsart*). Levin even suggests that the connection between her verb classes and lexical aspect be carefully investigated since lexical *aspect* also plays an important role in determining verb behavior (Vendler, 1957; Dowty, 1979). Knowledge about the lexical *aspect* of verbs is important in aiding language understanding, such as textual entailment.
2. Levin chooses the verb classes because their members participate in certain diathesis alternations. This strategy has led to the omission of certain verbs and verb classes. To be exact, Levin classification has been restricted to verbs taking noun phrase (NP) and prepositional phrase (PP) complements and verbs taking sentential complements are for the most part ignored. Lately there have been great efforts to extend Levin verb classification by incorporating new classes (Korhonen and Briscoe, 2004).
3. Levin verb classification lacks a hierarchical organization that usually characterizes other hand-crafted lexical resources, such as WordNet and FrameNet. Levin leaves it as an open question whether a complete hierarchical organization of English verbs is desirable, but arranging verb classes into a structure that is linked by some relations (e.g. *Is-a*, *Causative-of*) is an obvious benefit that various NLP tasks can take advantage of, such as paraphrase, question-answering and inferencing (Baker et al., 1998).

1.2 Goals

Despite all the limitations due to its design decisions, Levin verb classification has motivated a recent wealth of work on automatic acquisition of verb lexicons from corpus texts. Recently there has been a substantial amount of work on automatically inducing Levin-style verb classification from corpus data. Most of these studies represent each Levin verb as a vector of some type of feature, usually extracted from corpora. Machine learning methods are then applied to discover semantic classes of verbs from these features. These studies differ from each other in several aspects:

- *Feature Set*. While most studies focus on using subcategorization frames (Schulte im Walde, 2000; Korhonen et al., 2003), others make attempts to incorporate lexical information into the feature space (Joanis and Stevenson, 2003; Joanis et al., 2007).

- *Number of Verbs and Verb Classes.* Most of these studies test a small set of verbs or a small number of verb classes. For example, Schulte im Walde (2000) uses 153 verbs and 30 classes, and Joanis et al. (2007) deal with 835 verbs and 15 classes from Levin.
- *Machine Learning Methods.* Earlier work adopts either a supervised or an unsupervised machine learning method for the discovery of verb classes. Supervised classification (e.g. decision trees, SVMs) automatically assigns verbs into one of the verb classes in an existing verb class taxonomy (Joanis et al., 2007), and unsupervised induction (e.g. *k*-means) infers a verb classification through clustering (Schulte im Walde, 2000).

The primary goal of this paper is to explore the relation between Levin’s approach, her theoretical motivations and corpus data. In particular, we aim to identify the most informative and effective features for automatic verb classification. Automatic verb classification helps avoid the expensive hand-coding of such information, but appropriate features must be identified and demonstrated to be effective:

- Levin classifies verbs based on their syntactic behavior. What are the most informative features in automatic verb classification?
- Levin pre-selects syntactic frames that display alternations. Are these frames important for classification?
- Levin classifies verbs based on whether a given verb participates in a particular alternation, without regard to frequency. Does the frequency information, derived from corpus data, benefit or hurt automatic verb classification?

To this end, a series of supervised classification experiments, representing a wide range of class distinctions, are conducted to test the general applicability and scalability of each feature set as we attempt to classify more English verbs into a larger number of verb classes. In particular, we include a classification task that involves 48 Levin classes and 1,307 verbs. This is, to the best of our knowledge, the largest-scale investigation on automatic classification of English verbs.

2 Verb Features for Automatic Verb Classification

In this section, we summarize the different features used in the previous studies on automatic verb classification, and identify the deficiency of these features with respect to the task of automatic verb classification. To overcome their deficiency, we propose a wider range of feature sets that effectively combines syntactic and lexical information, which we believe are both essential for automatic verb classification.

2.1 Verb Features Previously Used

Early work on automatic verb classification has generally adopted one of the two approaches for collecting statistical, corpus-based features to capture verb behaviors as described in Levin.

Subcategorization Frames (SCF): Subcategorization frames are obviously relevant to alternation behaviors. It is therefore unsurprising that much of the work on verb classification has adopted them as features (Schulte im Walde, 2000; Brew and Schulte im Walde, 2002; Schulte im

Walde and Brew, 2002; Korhonen et al., 2003). These features are general, in that they apply equally to any verb classes. However, using subcategorization information alone leads to the loss of semantic distinctions. Levin verb classification has been mainly concerned with verbs taking NP and PP complements. It is not surprising that prepositions should play an important role in defining relevant subcategorization frames. However, only knowing the identity of prepositions is not always enough. 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 (13), 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.

- (13) 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 (14).

- (14) 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 a certain extent (15), these features do not generally improve classification performance (Schulte im Walde, 2000; Joanis, 2002).

- (15) 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 automatic verb classification. 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 requires 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.

Merlo and Stevenson (2001) perform a classification task involving three optionally transitive verb classes, including unergative, unaccusative, and object-drop. However, these three classes exhibit similarities with respect to their argument structure, in that they all can be used both as transitive and intransitive (Table 3). Therefore, argument structure alone does not distinguish these verb classes, and subcategorization information needs to be refined by thematic relations. Merlo and Stevenson define verb features based on linguistic heuristics that describe thematic relations

between subject and object in transitive and intransitive verb usage. These features are obtained using heuristics for transitivity, causativity, animacy, and syntax. Take the feature of animacy as an example (see Table 3). Unaccusative verbs have a *theme* subject in the intransitives, hence lower use of animate subjects than unergative and object-drop verbs. Although these features are effective for the three classes they select, they are limited in their applicability because they are devised specific to the particular class distinctions investigated. New deep linguistic analysis is needed when more classes are involved.

| Class | Trans. | Example | Subject | Object |
|--------------|--------|---|---------------|--------|
| unergative | intr. | The horse raced past the barn. | agent | |
| | trans. | The jockey raced the horse past the barn. | causative ag. | agent |
| unaccusative | intr. | The butter melted in the pan. | theme | |
| | trans. | The cook melted the butter in the pan. | causative ag. | theme |
| object-drop | intr. | The boy played. | agent | |
| | trans. | The boy played soccer. | agent | theme |

Table 3: Thematic role assignment for optionally transitive verb classes Merlo and Stevenson (2001)

Joanis et al. (2007) report an extension of Merlo and Stevenson’s work by defining a general feature space that includes 224 core features. They analyze the possible alternations that are primary determinants of the verb classification. Since they analyze verb class distinctions at a general level, they need only do the linguistic analysis once, rather than having to do an individual analysis for every set of classes that they want to distinguish. By basing their features directly on the alternation themselves, rather than on the existing classes, they have devised a general feature space which in principle is useful for any Levin-style verb classification task. The features they use fall into four different groups: *syntactic slots*, *slot overlaps*, *tense*, *voice and aspect*, and *animacy of NPs*.

- *Syntactic Slots*: These features encode the frequency of the syntactic positions, and are considered an approximation to subcategorization frames. The syntactic slots they considered include mainly NP and PP complements.
- *Slot Overlaps*: Using syntactic slot information alone misses potentially important properties of alternations, since verbs from different classes may occur in the same syntactic frames, but undergo different mappings of their arguments to the positions. Slot overlap features are designed 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, the 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. For example, animacy is definitely relevant to the agent/theme distinction.

Joanis et al. (2007) show 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, their results also show that this general feature space does not generally improve the classification accuracy over subcategorization frames (see Table 4).

| Experimental Task | General Feature Space | SCF |
|--------------------------|-----------------------|------|
| average 2-way | 83.2 | 80.4 |
| average 3-way | 69.6 | 69.4 |
| average (≥ 6)-way | 61.1 | 62.8 |

Table 4: Results (macro-averaged recall %) on automatic verb classification from Joanis et al. (2007)

2.2 Integration of Syntactic and Lexical Information

Earlier attempts at combining syntactic and lexical information, as described in the previous section, have achieved little success in automatic verb classification. In the following, we propose several different ways for combining syntactic and lexical information.

Dependency Relations (DR): Recall that subcategorization frames are limited as verb features in the properties of verb behaviors they tap into. Lexicalized frames, with potentially improved discriminatory power, suffer from increased exposure to data sparsity. Our way to overcome data sparsity is to break lexicalized frames into dependency relations. Dependency relations contain both syntactic and lexical information (16).

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

However, since we augment prepositional phrases with the head nouns selected by prepositions, as in PP(*with:fork*), the data sparsity problem still exists. We therefore break all prepositional phrases in the form PP(preposition:noun) into two separate dependency relations: PP(preposition) and PP-noun, as shown in (17).

- (17) a. SUBJ(*I*), PP(*with*), PP-*fork*
 b. SUBJ(*I*), PP(*with*), PP-*friend*
 c. SUBJ(*I*), PP(*with*), PP-*confidence*

Although dependency relations have proved effective in a wide range of automatic acquisition of lexical information, such as WSD (McCarthy et al., 2004), construction of lexical semantic space (Pado and Lapata, 2007), and detection of polysemy (Lin, 1998), their utility in automatic verb classification still remains untested.

Co-occurrences (CO): CO features mostly convey lexical information and are generally considered to contain little syntactic information and not particularly sensitive to argument structures (Rohde et al., 2004). Levin considers verbs in each of her classes *syntactic synonyms*, but not *lexical synonyms*. In other words, verbs in the same Levin class are substitutable for each other

in the frame set that the whole class licenses. However, they should not be assumed to be always substitutable for each other in exactly the same textual contexts. At first blush, co-occurrences do not qualify as good candidates for capturing verb behaviors as described in Levin. However, it seems reasonable to assume that the meaning components shared by verbs in a given Levin class may be correlated with the neighboring words with which they occur. Consider the class of DRIVE, which includes verbs such as *drive*, *fly*, and *row*. These verbs all describe events in which a *driver* takes a *passenger* to a *destination* by means of a *vehicle*.

- (18) a. The car drives to the city center.
b. The woman drives to Chicago.
c. The son drives a blue car.
d. The boy drives his father to downtown.

Obviously, it is not always acceptable to substitute *drive* with *fly* or *row* in (18) without creating a semantic anomaly. Lexical words fulfilling the role of *vehicle* certainly vary from verb to verb (*car* for *drive*, *plane* for *fly* and *boat* for *row*). Nevertheless, there should be a great overlap in lexical words that can fulfill other participant roles (e.g. *destination*). Hence, it is possible that 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-occurrences (ACO): Earlier work on lexical acquisition that uses CO features is only interested in words that reflect lexical semantics only. As a result, they consider it necessary to reduce the influence due to other factors, such as syntax. One key source of syntactic information in CO is function words, such as determiners, prepositions, and auxiliary verbs. The patterns with which a word co-occurs with these function words are considered to primarily reflect syntactic type, with very little bearing on lexical semantics. It is therefore a common practice to eliminate these function words from the feature space. However, some of the function words, prepositions in particular, have been demonstrated to carry a great amount of syntactic information that is closely related to the lexical meaning of Levin verbs (Schulte im Walde, 2000; Brew and Schulte im Walde, 2002; Joanis et al., 2007). We therefore propose to adapt the conventional CO features by keeping all prepositions in the context window of a target Levin verb. Consider the example sentences in (18) again. The participant role of *destination* for *drive*, *fly*, *row* is usually headed by a preposition denoting direction or path, such as *to*, *into*, and *across*. In addition, although verbs tend to put a strong selectional preference on their nominal arguments, they are not really selective about the identity of verbs in their verbal arguments (e.g. infinitive, sentential complement). We, therefore, decide to replace all verbs in the neighboring contexts of each target verb with their POS tags, an attempt to approximate argument structures like infinitive and sentential complement. ACO features defined this way integrate at least some degree of syntactic information into a feature space that is dominated by lexical information.

ScfCo: Another way to combine syntactic and lexical information is to use SCF and CO together. Since they are complementary to each other in terms of the kind of information they convey, a feature set combining them together has the potential for yielding better results than either of them used alone.

| |
|--|
| <i>Two men broke the door with a hammer.</i> |
| (det door_4 the_3) |
| (dobj _ broke_2 door_4) |
| (det hammer_7 a_6) |
| (dobj with_5 hammer_7) |
| (iobj broke_2 with_5) |
| (ncsubj broke_2 men_1 _) |
| (det men_1 two_0) |

Table 5: Example of grammatical relations generated by the CCG parser

3 Experiment Design

3.1 Corpus

To collect each type of feature, we use the Linguistic Data Consortium’s English Gigaword Corpus (LDC 2003), which consists of samples of recent newswire text data collected from four distinct international sources of English newswire. Although it is a newswire corpus, it covers a wide spectrum of subjects, and therefore should avoid any strong domain-specific verb usage.

3.2 Feature Extraction

We evaluate six different feature sets for their effectiveness in automatic verb classification: SCF, CO, ACO, DR, ScfCo, and JOANIS07. SCF contains mainly syntactic information, whereas CO encodes lexical information. The other four feature sets represent four different ways of combining syntactic and lexical information. In the following, we describe how each feature set is extracted from the Gigaword Corpus.

SCF and **DR**: A brief summary of the process for extracting SCF and DR features is given below:

- Parse every input sentence using the CCG parser (Clark and Curran, 2007). Table (5) lists the grammatical relations for the sentence *Two men broke the door with a hammer*.
- Build a lexicalized frame for the verb *break*. This is done by matching each grammatical label onto one of the traditional syntactic constituents:¹ NP1(*men*)-V-NP2(*door*)-PP(*with:hammer*).
- Construct a SCF and a set of DRs based on the lexicalized frame. The SCF is created by simply combining all syntactic constituents in the lexicalized frame: NP1-V-NP2-PP*with*. The set of DRs is obtained by breaking the lexicalized frames: [SUBJ(*man*), OBJ(*door*), PP(*with*), PP-*hammer*].

For our initial experiment, we do not assume knowledge of the Levin-defined SCF set. Instead, we take all SCFs proposed by the CCG parser, and use them as features for automatic verb classification. Additional experiments will be performed in section 6.4 using Levin-defined SCFs as features to investigate whether expert-defined SCFs are necessary for automatic verb classification.

CO: CO features are collected in the following way:

¹For example, we match the grammatical label *ncsubj* to the syntactic constituent NP1.

- Use a flat 4-word window, meaning that the 4 words to the left/right of each target Levin verb are considered candidates for CO features. We choose a relatively small window because smaller windows have been shown to work well for verbs in various lexical acquisition tasks (Leacock et al., 1998; Rohde et al., 2004).
- Eliminate any words that are in a stop word list, which consists of about 150 function words including mainly prepositions, determiners, and punctuation (Rohde et al., 2004).
- Lemmatize each word using the English lemmatizer as described in Minnen et al. (2000), and use lemmas as features instead of words.

ACO: We adapt the conventional CO features in the following ways to incorporate some degree of syntactic information into the feature set:

- Keep all prepositions.
- Replace all verbs in the neighboring contexts of each target verb with their POS tags.
- Keep the words in the left window only if it is tagged as a noun or pronoun. In our task, contents of left windows are supposed to represent subjects of a particular target verb, and in English nouns and pronouns are more likely to fill the slots for subjects.

ScfCo: We simply combine SCF and CO features together.

JOANIS07: This feature set consists of 224 features in four categories, as shown in Table (6).

| Feature Category | Number of Features |
|------------------|--------------------|
| syntactic slots | 77 |
| use of pronouns | 6 |
| slot overlap | 41 |
| passive | 2 |
| POS of the verb | 6 |
| aux, modal, adv | 13 |
| derived forms | 3 |
| animacy of NPs | 76 |

Table 6: Feature categories with the number of features of each type (Joanis et al., 2007)

We briefly describe how these features are extracted in our experiments. The extraction process follows Joanis et al. (2007) whenever we can, but deviates as needed.

- *Syntactic Slot Features:* We break each subcategorization frame into individual syntactic slots, and gather a count over each of the following syntactic slots - subject, direct and indirect object, and prepositional phrase - independently of their occurrences with other slots. For prepositional phrases, we include a separate feature for each of 51 high frequency prepositions, as well as 19 groups of closely related prepositions (e.g. one group consists of *between*, *in between*, *among*, *amongst*, *amid* and *amidst*). See Joanis (2002) for the full list of prepositions and groups. Levin includes a number of alternations in which a syntactic slot

contains a specific word or class of words, in particular pronouns. To approximate these uses, we add features to count reflexive pronouns (*-self*) in the object position; *it* in the subject, intransitive subject and direct object positions; and *there* in the subject and transitive subject positions.

- *Slot Overlap Features*: We consider the overlap between each pair of slots that corresponds to an alternation used by Levin to characterize the classes. For each alternation in which a semantic argument occurs in one slot in one usage of the verb, and in a different slot in the alternant usage, we add a feature with the measure of overlap in noun (lemma) usage between these two slots for the verb. For example, given the alternation exemplified by *The sky cleared/The clouds cleared from the sky*, we add an overlap feature for the subject slot and the object-of-from slot.²
- *Tense, Voice and Aspect Features*: We first include a set of features that encodes the proportion of occurrences of the six POS tags that verbs can take in the Penn Treebank tagset: VB, VBP, VBZ, VBG, VBD, and VBN. We further augment the feature space to count verb uses which occur with an adverb or with each specified auxiliary or modal: *have, be, will, can, could, might, must, would, should, and may*. We also include a set of features that measures the frequency of a verb used as a noun or as an adjective (in both past participle and present participle forms).
- *Animacy Features*: We consider the animacy of each of our syntactic slots. Merlo and Stevenson (2001) use a simple linguistic heuristic to estimate the animacy features. They count as animate all personal pronouns other than *it*. Joanis et al. (2007) additionally include proper NPs labelled as PERSON by the chunker they use. The CCG parser we use for extracting syntactic information does not provide such information, we therefore follow Merlo and Stevenson (2001) in using personal pronouns to heuristically estimate the counts of animacy features.

Table (7) provides an example of how each feature set is represented using the sentence *Two men broke the door with a hammer*.

3.3 Verbs and Verb Classes

We conduct a series of classification experiments involving two separate sets of verbs and verb classes, both of which are based on Levin:

Joanis15: Joanis et al. (2007) choose 15 Levin verb classes that exhibit a range of syntactic and semantic distinctions, to evaluate the robustness of the general feature space they have proposed. To this end, they manually select pairs (or, in two cases, triples) of classes to represent a range of distinctions that exist among the classes in general. For example, some of the pairs/triples are

²Let S_x be the set of lemmas occurring in slot X for a given verb, MS_x the multiset of lemmas occurring in slot X , and w_x the number of times lemma w occurs in MS_x . Then we define the overlap between slots A and B as follows:

$$overlap(A, B) = \frac{\sum_{w \in S_A \cap S_B} \max(w_A, w_B)}{|MS_A| + |MS_B|} \quad (1)$$

For example, if a, a, a, b, b, c is the multiset of lemmas occurring as subject of a verb, and a, b, b, b, b, d, d, e is the multiset of lemmas occurring as direct object of the same verb, then $overlap(\text{subject}, \text{object}) = (3+4)/(6+8) = 0.5$. For more discussion, see Joanis (2002).

| Feature Set | Features |
|-------------|---|
| SCF | NP1-V-NP2-PP <i>with</i> |
| DR | SUBJ(<i>man</i>), OBJ(<i>door</i>), PP <i>with</i> , PP- <i>hammer</i> |
| CO | <i>two, man, door, hammer</i> |
| ACO | <i>two, man, door, with, hammer</i> |
| ScfCo | NP1-V-NP2-PP <i>with, two, man, door, with, hammer</i> |
| JOANIS07 | SUBJ(<i>man</i>), OBJ(<i>door</i>), PP <i>with</i> , VBD ANIMATE(SUBJ), INANIMATE(OBJ), INANIMATE(PP <i>with</i>) |

Table 7: An example of representations of each feature set

| Verb Class | Levin Class Number | Number of Verbs | Examples |
|--------------------------|--------------------|-----------------|------------------------|
| BENEFACTIVE | 26.1, 26.3 | 49 | <i>bake, buy</i> |
| RECIPIENT | 13.1, 13.3 | 33 | <i>give, offer</i> |
| ADMIRE | 31.2 | 39 | <i>admire, adore</i> |
| AMUSE | 31.1 | 157 | <i>amuse, amaze</i> |
| RUN | 51.3.2 | 85 | <i>amble, run</i> |
| SOUND-EMISSION | 43.2 | 63 | <i>bang, click</i> |
| LIGHT-SUBSTANCE-EMISSION | 43.1, 43.4 | 41 | <i>radiate, flash</i> |
| CHEAT | 10.6 | 30 | <i>cheat, deplete</i> |
| STEAL-REMOVE | 10.5, 10.1 | 50 | <i>steal, dismiss</i> |
| WIPE | 10.4.1, 10.4.2 | 42 | <i>wipe, clear</i> |
| SPRAY-LOAD | 9.7 | 42 | <i>spray, load</i> |
| FILL | 9.8 | 65 | <i>fill, plug</i> |
| PUTTING | 9.1-6 | 59 | <i>place, mount</i> |
| CHANGE-OF-STATE | 45.1-4 | 200 | <i>broaden, mellow</i> |
| OBJECT-DROP | 26.1, 26.3, 26.7 | 64 | <i>blend, toss</i> |

Table 8: Joanis15 - Verb classes, their Levin class numbers, the number of experimental verbs in each class, and example verbs in each class

syntactically dissimilar, while others show little syntactic distinction across the classes. Table (8) lists the 15 classes, along with their Levin class numbers and the number of verbs in each class.³ These 15 verb classes involve 835 verbs.

These 835 experimental verbs are selected as follows: It starts with a list of all the verbs in the selected Levin classes, but removes any verb that did not occur at least 100 times in the BNC. Because they assign a single class label to each verb in our experiments, they remove any verb that belongs to more than one of the classes in their selected pairs/triples of classes. They also remove any verb that they deem to be overly polysemous (belonging to six or more Levin classes). With these 15 selected classes, Joanis et al. (2007) perform 11 classification tasks of varying difficulty. The classification tasks are summarized as follows:

Basic classification tasks: Below we provide a discussion on the pairs/triples of classes along with the syntactic and semantic distinctions among them:

- 1) BENEFACTIVE versus RECIPIENT:
Mary baked /a cake for Joan/Joan a cake.

³Note that some of the 15 classes selected by Joanis et al. (2007) are actually obtained by merging two or more original Levin classes.

Mary gave /a cake to Joan/Joan a cake.

These two alternations differ mainly in the preposition used.

- 2) ADMIRE versus AMUSE:

I admire Jane.

Jane amuses me.

These two classes differ in the mapping of semantic roles to syntactic positions. For ADMIRE verbs, the EXPERIENCER argument is the subject, while the STIMULUS argument is the object. The mapping is reversed for AMUSE verbs.

- 3) RUN versus SOUND-EMISSION:

*Kids ran in the room./*The room run with kids.*

*Birds sang in the trees./*The trees sang with birds.*

The two classes differ in some of the prepositional alternations they allow.

- 4) LIGHT-SUBSTANCE-EMISSION versus SOUND-EMISSION:

The jewels sparkled./The fountain gushed.

The hinges squeaked.

These classes allow most of the same alternations, with each allowing one or two alternant forms that others do not allow.

- 5) CHEAT versus STEAL-REMOVE:

*I cheated /Jane of her money/*the money from Jane.*

*I stole /*Jane of her money/the money from Jane.*

These semantically related classes differ in the prepositional alternants they allow.

- 6) WIPE versus STEAL-REMOVE:

Wipe /the dust/the dust from the table/the table.

*Steal /the money/the money from the bank/*the bank.*

The classes here generally allow the same syntactic frames, but differ in possible semantic role assignment. For example, although both classes allow a transitive frame, the LOCATION argument can appear as the direct object of WIPE, but not of STEAL-REMOVE.

- 7) SPRAY-LOAD versus FILL versus PUTTING:

I loaded /hay on the wagon/the wagon with hay.

*I filled /*hay on the wagon/the wagon with hay.*

*I put /hay on the wagon/*the wagon with hay.*

These classes also differ in prepositional alternants. However, the options for SPRAY-LOAD overlap with both of the other two classes.

- 8) RUN versus CHANGE-OF-STATE versus OBJECT-DROP:

The trainer jumped the lion through the hoop./The lion jumped through the hoop.

The chef melted the butter in the pan./The butter melted in the pan.

Mary baked a cake for Joan./Mary baked for Joan.

These are the three classes of Merlo and Stevenson (2001). All are optionally intransitive but assign different semantic roles to their arguments.

Additional multiway tasks: In addition to the eight basic classification tasks, Joanis et al. (2007) also add the following three multiway tasks to explore how well their general feature space scales to multiple class distinctions:

- 9) 6-way task involving CHEAT, STEAL-REMOVE, WIPE, SPRAY-LOAD, FILL and PUTTING, all of which undergo similar alternations of locative arguments.
- 10) 8-way task that adds to the above 6-way task, plus the classes RUN and SOUND-EMISSION, which also undergo locative alternations.
- 11) 14-way task including all the classes (except BENEFACTIVE, which is a subset of OBJECT-DROP).

Levin48: Earlier work on automatic classification of English verbs has focused on a small set of verbs or a small number of verb classes. For example, Schulte im Walde (2000) uses 153 English verbs in 30 Levin classes, and Merlo and Stevenson (2001) deal with only 59 verbs in 3 classes. Joanis et al. (2007), the largest investigation of English verb so far, take on 835 verbs in 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 experiment, we aim for a larger scale experiment.

We select our experimental verb classes and verbs as follows: We start with all 191 Levin 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 experiment. This process yields 48 classes involving about 1,307 monosemous verbs. In our experiments, we test the applicability of each feature set to distinctions among up to 48 classes. To our knowledge, this is, by far, the largest investigation on English verb classification. The size of most classes falls in the range from 10 to 30, with a couple of classes having sizes over 100. Table (9) lists these 48 classes, along with Levin class numbers and the number of verbs in each class.

If we could exhaust all the possible m -way ($2 \leq m \leq 48$) classification tasks with the 48 Levin classes we investigate, it will certainly 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 extremely huge for intermediate values of m . For example, there will be $C_{48}^{20} = 10^{44}$ classification tasks when $m = 20$. That is too many for us to handle. For this reason, we set m to be 2, 5, 10, 20, 30, 40, or 48. For the 2-way classification, we perform all the possible $C_{48}^2 = 1,028$ tasks. For the 48-way classification, there is only one possible task. We randomly select 100 m -way tasks for each $m = 5, 10, 20, 30, 40$. We believe that these tasks represent a wide range of class distinctions, and will therefore give us a reasonably good idea of whether a particular feature set is generally applicable and scales well to multiple way distinctions.

4 Machine Learning Methods

4.1 Preprocessing Data

We represent the semantic space for verbs as a matrix of frequency, where each row corresponds to a Levin verb and each column represents a given feature. We construct a semantic space with each

| Verb Class | Levin Class Number | Number of Verbs | Examples |
|-----------------------|--------------------|-----------------|-----------------------------|
| ADMIRE | 31.2 | 33 | <i>adore, dislike</i> |
| AMALGAMATE | 22.2 | 27 | <i>amalgamate, conjoin</i> |
| AMUSE | 31.1 | 170 | <i>affront, amaze</i> |
| APPOINT | 29.1 | 10 | <i>appoint, ordain</i> |
| BEGIN | 55.1 | 10 | <i>commence, start</i> |
| BUTTER | 9.9 | 47 | <i>bread, caulk</i> |
| CAPTAIN | 29.8 | 26 | <i>captain, emcee</i> |
| CARVE | 21.2 | 17 | <i>dent, mangle</i> |
| CHANGE-OF-STATE | 45.4 | 159 | <i>age, blacken</i> |
| CHARACTERIZE | 29.2 | 29 | <i>class, employ</i> |
| CHEAT | 10.6 | 30 | <i>bereave, cure</i> |
| CONJECTURE | 29.5 | 11 | <i>conjecture, mean</i> |
| CONTIGUOUS-LOCATION | 47.8 | 15 | <i>border, overhang</i> |
| CONTRIBUTE | 13.2 | 14 | <i>disburse, relinquish</i> |
| COOKING | 45.3 | 14 | <i>broil, percolate</i> |
| CORRESPOND | 36.1 | 42 | <i>bargain, commiserate</i> |
| CRANE | 40.3.2 | 11 | <i>clench, hunch</i> |
| CREATE | 26.4 | 12 | <i>concoct, mint</i> |
| DECLARE | 29.4 | 12 | <i>assume, prove</i> |
| DESTROY | 44 | 11 | <i>decimate, ruin</i> |
| DISASSEMBLE | 23.3 | 14 | <i>sunder, unleash</i> |
| DUB | 29.3 | 10 | <i>consecrate, rule</i> |
| FILL | 9.8 | 46 | <i>clutter, deluge</i> |
| FUTURE-HAVING | 13.3 | 11 | <i>assign, promise</i> |
| GET | 13.5.1 | 14 | <i>cash, order</i> |
| GIVE | 13.1 | 11 | <i>give, sell</i> |
| JUDGMENT | 33 | 49 | <i>applaud, chastise</i> |
| LIGHT-EMISSION | 43.1 | 15 | <i>flare, glitter</i> |
| MANNER-OF-SPEAKING | 37.3 | 14 | <i>drawl, stammer</i> |
| MARVEL | 31.3 | 26 | <i>bask, fret</i> |
| NONVERBAL-EXPRESSION | 40.2 | 18 | <i>frown, guffaw</i> |
| PIT | 10.7 | 14 | <i>core, poll</i> |
| POCKET | 9.10 | 33 | <i>bank, bin</i> |
| REMOVE | 10.1 | 21 | <i>discharge, eradicate</i> |
| RUMMAGE | 35.5 | 12 | <i>forage, rummage</i> |
| RUN | 51.3.2 | 77 | <i>bowl, flit</i> |
| SAY | 37.7 | 14 | <i>blab, recount</i> |
| SEARCH | 35.2 | 14 | <i>dredge, scavenge</i> |
| SEND | 11.1 | 10 | <i>hand, ship</i> |
| SHAKE | 22.3 | 13 | <i>baste, jumble</i> |
| SOUND-EMISSION | 43.2 | 35 | <i>chime, crackle</i> |
| SPANK | 18.3 | 11 | <i>clobber, spank</i> |
| SPATIAL-CONFIGURATION | 47.6 | 10 | <i>loungue, sag</i> |
| SPRAY/LOAD | 9.7 | 28 | <i>daub, mound</i> |
| STEAL | 10.5 | 25 | <i>confiscate, filch</i> |
| TAPE | 22.4 | 32 | <i>button, gum</i> |
| VEHICLE-NAMES | 51.4.1 | 18 | <i>boat, motor</i> |
| WIPE-MANNER | 10.4.1 | 10 | <i>distill, suction</i> |

Table 9: Levin48 - Verb classes, their Levin class numbers, the number of experimental verbs in each class, and example verbs in each class

feature set. Except for JONAS07 which only contains 224 features, all the other feature sets lead to a very high-dimensional space. Table (10) lists the total number of features in each feature set. For instance, the semantic space with CO features contains over one million columns, which is too large and cumbersome.

| Feature Set | SCF | CO | ACO | DR |
|--------------------|--------|-----------|---------|---------|
| Number of features | 57,317 | 1,042,792 | 772,942 | 849,894 |

Table 10: Total number of features in each feature set extracted from the English Gigaword Corpus

Feature Selection: One way to avoid these high-dimensional spaces is to assume that most of the features are irrelevant. This is 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; Schutze, 1998). Rohde et al. (2004) have found it is simple and effective to discard columns on the basis of feature frequency. Columns representing low-frequency features tend to be noisier because they only involve few examples. We therefore follow Rohde et al. (2004) in using the most frequent n columns (features) for feature selection.

Data Normalization: A major problem with high-dimensional semantic space is that high frequency or higher variance columns can potentially contribute disproportionately to the distance measure, relative to the amount of information they convey, and can thus dominate the classification (Hsu et al., 2003). This is particularly problematic if a high frequency column is due to systematic errors made by NLP tools which are used for feature extraction. In order to reduce the undue influence of outlier features, normalization techniques are usually employed to help reduce the range of extreme values while having little effect on others. In our experiments, we adopt *correlation* for data normalization, a method proposed in Rohde et al. (2004):

$$w'_{vf} = \frac{T \times w_{vf} - \sum_j w_{vj} \times \sum_i w_{if}}{((\sum_j w_{vj} \times (T - \sum_j w_{vj})) \times (\sum_i w_{if} \times (T - \sum_i w_{if})))^{1/2}} \quad (2)$$

$$T = \sum_i \sum_j w_{ij}$$

The raw frequency w_{vf} of a verb v co-occurring with a feature f is replaced with the coefficient of correlation w'_{vf} . The correlation between the occurrence of verb v and feature f expresses the tendency of whether verb v occurs more or less often with feature f than it does in general. However, Rohde et al. (2004) also show that improved performance can be achieved by maneuvering the correlation normalization in the following ways:

- Eliminate all of the negative correlations. When using correlation for data normalization, the normalized value (w'_{vf}), will range from -1 to 1. However, it turns out that knowing the identity of anti-correlated features is not as helpful as knowing the identity of the positively correlated ones. To illustrate this point, imagine that you were asked to guess the word *dog*. Would you rather be told 10 words associated with the mystery word (e.g. *cat, bone, paw, collar*) or 100 words that have nothing to do with the mystery word (e.g. *whilst, missile, suitable, cloud*)? It should be obvious that the 10 positively correlated words would be much more helpful.

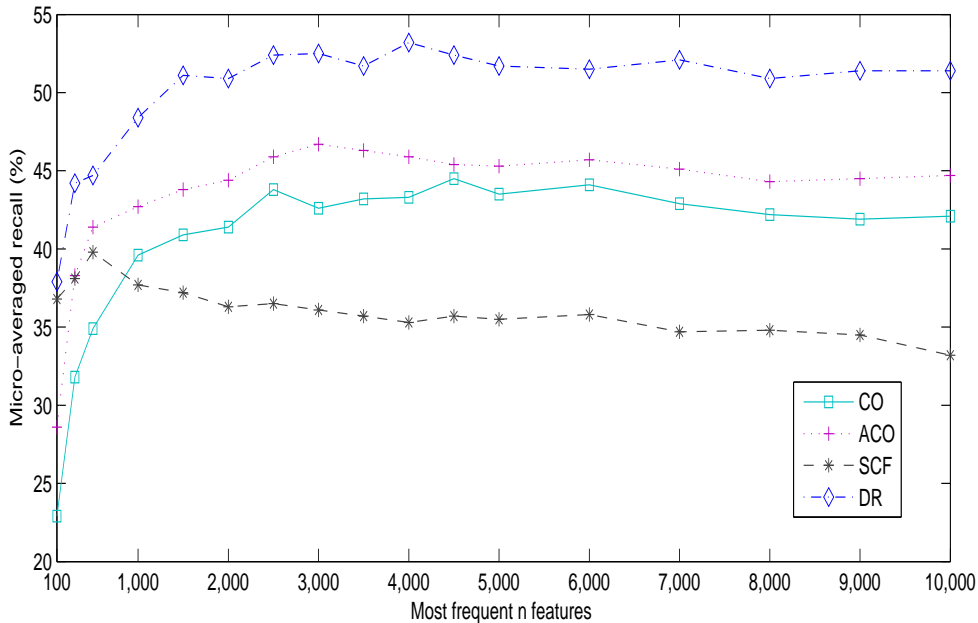


Figure 1: Classification results on the training data of the 48-way classification task

- Rather than using the straight positive correlation values, their square roots are used. This is not a terribly principled maneuver, but it has the beneficial effect of magnifying the importance of the many small values relative to the few large ones.

In sum, to preprocess data, we first select the most frequent n features, and then normalize it using the correlation method. To find the optimal value of n , we set n to be 100, 300, 500, and any value between 500 and 5,000 at each interval of 500, and between 5,000 to 10,000 at each interval of 1,000. In our experiments, for each task and feature set, we select the value of n that offers the best accuracy on the preprocessed training set according to k -fold stratified cross validation.⁴

We choose to use no more than 10,000 features in our classification tasks. This is based on the results from the classification experiments on the training data when we try to determine the optimal value for n to be used on the test data. Take the 48-way classification task as an example. It is one of the Levin48 tasks that involves all the 48 verb classes and 1,307 verbs we investigate. In the 48-way classification experiment to be detailed in section 5.3, we randomly select 9 verbs from each of the 48 classes as training data, and repeat this process 10 times. This gives us 10 different splits of training/test data. For each split, we need to find the value for n that offers the best accuracy on the training data, and this value will be used on the corresponding test data. To do this, we perform a classification task using 9-fold stratified cross-validation on each of the 10 training sets. The accuracy presented in Figure (1) is obtained by averaging the results from the classification experiments on these 10 training sets.

As shown in Figure (1), CO, ACO and DR show similar patterns of performance as the number of features selected increases. To be exact, they all exhibit a sharp jump in accuracy as the number of

⁴10-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.

features goes beyond 1,000. In fact, for each individual feature set, roughly equivalent performance is obtained from using anywhere between 2,000 to 10,000 features, with the performance declining gradually as the number of features approaches 10,000. SCF, on the other hand, reaches its accuracy peak when only 500 features are used, and its performance drops gradually as more SCF features are selected. On the basis of this result, we are reasonably confident that using more than 10,000 features is unlikely to lead to improved performance.

4.2 Classifier

We perform supervised classification on the verbs and verb classes we select. Earlier work on supervised automatic verb classification has employed classifiers such as decision trees (Merlo and Stevenson, 2001), and SVMs (Joanis et al., 2007). In our experiments, we perform the classification tasks using Bayesian Multinomial Logistic Regression (BMR), an efficient log-linear modeling framework (Madigan et al., 2005), which we found outperforms SVMs for this task.

BMR performs the so-called 1-of- k classification. It has been demonstrated to be very efficient with multi-way classification tasks, including text categorization, and author identification (Madigan et al., 2005; Genkin et al., 2004). These tasks all involve the handling of a large number of features and extremely sparsely populated matrices, which characterize the data we have for automatic verb classification.

To begin with, let $\mathbf{x} = [x_1, \dots, x_j, \dots, x_d]^T$ be a vector of feature values characterizing a verb to be classified. We encode the fact that a verb belongs to a class $k \in 1, \dots, K$ by a K -dimensional 0/1 valued vector $\mathbf{y} = (y_1, \dots, y_K)^T$, where $y_k = 1$ and all other coordinates are 0. Multinomial logistic regression is a conditional probability model of the form:

$$p(y_k = 1 | \mathbf{B}, \mathbf{x}) = \frac{\exp(\beta_k^T \mathbf{x})}{\sum_{k_i} \exp(\beta_{k_i}^T \mathbf{x})} \quad (3)$$

The model is parameterized by the matrix $\mathbf{B} = [\beta_1, \dots, \beta_K]$, where each column of \mathbf{B} is a parameter vector of feature weight corresponding to one of the classes k : $\beta_k = [\beta_{k1}, \dots, \beta_{kd}]^T$. That is:

$$\begin{bmatrix} \beta_{11} & \cdots & \beta_{k1} & \cdots & \beta_{K1} \\ \vdots & & \vdots & & \vdots \\ \beta_{1p} & \cdots & \beta_{kp} & \cdots & \beta_{Kp} \\ \vdots & & \vdots & & \vdots \\ \beta_{1P} & \cdots & \beta_{kP} & \cdots & \beta_{KP} \end{bmatrix} \quad (4)$$

Classification of a new observation is based on the vector of conditional probability estimates produced by the model. Usually the model simply assigns the class with the highest conditional probability estimate:

$$\hat{y}(\mathbf{x}) = \arg \max_k p(y_k = 1 | \mathbf{x}) \quad (5)$$

Given a training set $\{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_i, \mathbf{y}_i), \dots, (\mathbf{x}_n, \mathbf{y}_n)\}$, estimates of the values of \mathbf{B} are obtained via the method of maximum likelihood. Maximum likelihood estimation involves choosing

values of \mathbf{B} that are most consistent with the sample data, e.g. the likelihood of \mathbf{B} given the training data set is maximized. Maximum likelihood estimation of the parameters \mathbf{B} is equivalent to minimizing the negated log-likelihood:

$$l(\mathbf{B}|D) = - \sum_i [\sum_k y_{ik} \beta_k^T \mathbf{x}_i - \ln \sum_k \exp(\beta_k^T \mathbf{x}_i)] \quad (6)$$

Bayesian approaches to logistic regression involve specifying a distribution on \mathbf{B} that reflects prior beliefs about likely values of the parameters. The most widely used approach is to impose a univariate Gaussian prior with mean 0 and variance σ_{kj}^2 on each parameter β_{kj} . By specifying a mean of 0 for each Gaussian, we encode our prior belief that β_{kj} will be near 0. The variances of the Gaussians, σ_{kj}^2 , are positive constants we must specify. A small value of σ_{kj}^2 represents a prior belief that β_{kj} is close to zero, while larger value represents less confidence in this. However, the Gaussian prior, while favoring values of β_{kj} near 0, does not favor them being exactly equal to 0. Hence, the estimates of all or most β_{kj} will be nonzero. Another common choice of prior is the Laplacian, which favors values of \mathbf{B} to be 0. The basic idea behind this way of specifying priors is that in the typical classification setting involving large data sets in a high dimensional feature space, it is reasonable to expect that many of the features are redundant or noisy, and only a small subset are most important for classification. A feature which is a strong predictor of a single class will tend to get a large β_{kj} for that class, and a \mathbf{B} of 0 for most other classes.

Genkin et al. (2004) have found in their text categorization experiments that using a Laplacian prior systematically outperforms using a Gaussian prior. Our experiments, on the other hand, show that using a Gaussian prior consistently yields better results.⁵ The results we report below are obtained by using the Gaussian prior.

5 Results and Discussion

5.1 Evaluation Metrics

To compare the performance of each individual feature set in automatic verb classification, we adopt two evaluation metrics: macro-averaged recall and micro-averaged recall.

Macro-averaged recall: 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 good 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:

$$recall = \frac{\text{no. of test verbs in class } c \text{ correctly labeled}}{\text{no. of test verbs in class } c} \quad (7)$$

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

⁵We suspect that this is due to the differences in the predictive strength of the features employed. In the content-based text categorization, for each class, there are usually a few features (content words) that are strong indicators of the topics in which they occur. For instance, in the Reuters-RCV1 classes, words like *soccer*, *cup*, *match*, *game*, *played*, *league* are all strong predictors of the class SPORTS, whereas words like *election*, *party*, *polls*, *voters*, *candidate*, *campaign* are very indicative of the class ELECTION. In contrast, the features used for automatic verb classification are usually not strong predictors of a single class. As a matter of fact, each individual feature is likely to be associated with many verb classes. Take the SCF features as an example, more than 20 Levin verb classes license the double object frame (NP1-V-NP2-NP3).

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

C : a set of verb classes
 c : an individual verb class
 $|C|$: the number of verb classes

(8)

Micro-averaged recall: Micro-averaged recall treats each verb type equally, so that a verb class with many test verbs is mostly likely to dominate the micro-averaged recall. However, micro-averaged recall is a good indicator of the quality of classification across all verb types. Micro-averaged recall is defined by:

$$\text{micro-averaged recall} = \frac{\text{total no. of test verb types correctly assigned}}{\text{total no. of test verb types}}$$
(9)

5.2 Results on Joanis15

Joanis et al. (2007) perform 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 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 are obtained by averaging the results of the 10 trials. For each trial, each feature set is trained and tested on the same training/test split.

The macro-averaged recall and micro-averaged recall for each of the 11 classification tasks are presented in Tables (11) and (12) respectively. For comparison purposes, we also provide a chance baseline and the macro-averaged recall reported in Joanis et al. (2007)⁶. A few points are worth noting:

- Despite their evident effectiveness in supporting Levin’s intuition, SCF, at least when used alone, is not the most effective feature set for automatic verb classification. Our experiments show that the performance achieved by using SCF is generally worse than using the feature sets, ScfCo and DR in particular, which mix syntactic and lexical information. To be more specific, when measured by macro-averaged recall, SCF loses to ScfCo and DR on 9 out of the 11 tasks. Its performance fairs a little better when measured on micro-averaged recall. However, it still loses to ScfCo on 8 out of 11 tasks, and to DR on 6 tasks. SCF even loses to the simplest feature set, CO on 4 tasks regardless of the evaluation metric used, including the most difficult 14-way task.
- Although there is not a clear winner on these 11 classification tasks, the two feature sets (DR, ScfCo) we propose that combine syntactic and lexical information generally perform

⁶Joanis et al. (2007) only use macro-averaged recall. It should also be noted that experiments in Joanis et al. (2007) are different from ours in several aspects: 1. They use a chunker for feature extraction; 2. They extract verb features from the BNC; 3. They use SVMs for classification. 4. They perform only one trial on each classification task.

| Experimental Task | Random Baseline | Joanis 2007 | Feature Set | | | | | |
|--|-----------------|-------------|-------------|-------------|-------------|------|-------------|----------|
| | | | SCF | DR | CO | ACO | ScfCo | JOANIS07 |
| 2-way | | | | | | | | |
| Benefactive/Recipient | 50 | 86.4 | 88.7 | 88.9 | 88.0 | 89.3 | 90.6 | 88.9 |
| Admire/Amuse | 50 | 93.9 | 96.5 | 97.7 | 91.9 | 90.8 | 96.6 | 96.6 |
| Run/Sound-Emission | 50 | 86.8 | 86.4 | 89.7 | 92.1 | 90.2 | 90.6 | 87.1 |
| Light/Sound-Emission | 50 | 75.0 | 76.8 | 91.2 | 86.8 | 89.9 | 89.2 | 82.1 |
| Cheat/Steal-Remove | 50 | 76.5 | 77.9 | 81.4 | 73.1 | 75.8 | 77.8 | 76.4 |
| Wipe/Steal-Remove | 50 | 80.4 | 84.8 | 80.9 | 79.1 | 79.6 | 84.4 | 83.9 |
| 3-way | | | | | | | | |
| Spray-Load/Fill/Putting | 33.3 | 65.6 | 72.8 | 72.7 | 61.0 | 66.8 | 73.8 | 69.6 |
| Run/State-Of-Change/Object-Drop | 33.3 | 74.2 | 74.8 | 77.7 | 76.5 | 77.6 | 80.6 | 75.5 |
| \geq 6-way | | | | | | | | |
| Cheat/Steal-Remove/Wipe/ Spray-Load/Fill/Putting | 16.7 | 64.3 | 64.7 | 65.4 | 56.3 | 59.9 | 65.0 | 64.3 |
| Run/Sound-Emission/Cheat/ Steal-Remove/Wipe/Fill/ Spray-Load/Putting | 12.5 | 61.7 | 62.5 | 66.1 | 57.7 | 61.3 | 66.9 | 63.1 |
| 14-way (all except Benefactive) | 7.1 | 58.4 | 56.9 | 65.9 | 57.7 | 59.6 | 66.6 | 57.2 |

Table 11: Experimental results (macro-averaged recall) for Joanis15 (%)

| Experimental Task | Random Baseline | Feature Set | | | | | |
|--|-----------------|-------------|-------------|-------------|------|-------------|-------------|
| | | SCF | DR | CO | ACO | ScfCo | JOANIS07 |
| 2-way | | | | | | | |
| Benefactive/Recipient | 50 | 90.1 | 89.8 | 87.2 | 89.7 | 90.5 | 90.3 |
| Admire/Amuse | 50 | 96.3 | 97.1 | 92.5 | 90.1 | 96.8 | 95.2 |
| Run/Sound-Emission | 50 | 84.8 | 89.1 | 92.1 | 88.9 | 90.1 | 87.1 |
| Light/Sound-Emission | 50 | 64.8 | 96.1 | 93.5 | 93.8 | 93.5 | 87.8 |
| Cheat/Steal-Remove | 50 | 73.7 | 72.4 | 60.4 | 65.2 | 70.4 | 71.4 |
| Wipe/Steal-Remove | 50 | 81.7 | 76.3 | 74.9 | 73.8 | 78.2 | 82.1 |
| 3-way | | | | | | | |
| Spray-Load/Fill/Putting | 33.3 | 76.9 | 74.0 | 56.0 | 68.7 | 75.9 | 67.9 |
| Run/State-Of-Change/Object-Drop | 33.3 | 76.7 | 79.1 | 77.2 | 80.2 | 82.5 | 77.5 |
| \geq 6-way | | | | | | | |
| Cheat/Steal-Remove/Wipe/ Spray-Load/Fill/Putting | 16.7 | 65.5 | 64.6 | 47.0 | 55.7 | 66.1 | 63.0 |
| Run/Sound-Emission/Cheat/ Steal-Remove/Wipe/Fill/ Spray-Load/Putting | 12.5 | 64.8 | 67.8 | 55.1 | 61.4 | 67.4 | 64.2 |
| 14-way (all except Benefactive) | 7.1 | 54.7 | 64.5 | 55.0 | 58.7 | 64.9 | 56.3 |

Table 12: Experimental results (micro-averaged recall) for Joanis15 (%)

better than those feature sets (SCF, CO) that only include syntactic or lexical information, suggesting that they are both 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. all ≥ 6 -way tasks), exhibiting the tendency to scale well with a larger number of verb classes and verbs. Another feature set we propose, ACO, which retains some function words in the feature space to preserve a certain degree of 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 the keys to improved automatic verb classification.

- Although JOANIS07 also combines syntactic and lexical information, its performance is not comparable to that of other feature sets that also include syntactic and lexical information. In fact, SCF and JOANIS07 yield similar accuracy on most classification tasks in this experiment, which agrees with the findings in Joanis et al. (2007) (compare Table (4) with (11) and (12)).

5.3 Results on Levin48

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

The results (both macro-averaged recall and micro-averaged recall) for Levin48 are plotted in Figure (2), which offers a clear view of the general applicability and scalability of each feature set:

- Results from Levin48 reconfirm that SCF is not the most effective feature set for automatic verb classification. For both evaluation metrics the accuracy achieved by using SCF drops drastically as m gets bigger, indicating that SCF does not scale as well as other feature sets when dealing with a larger number of verbs and verb classes. On the other hand, CO, which is deemed to convey only lexical information, generally slightly outperforms SCF on m -way classification when $m \geq 10$. This result seems to suggest that verbs in the same Levin class tend to share their neighboring words.
- Unlike Joanis15 where there is no clear winner, the two feature sets (ScfCo, DR) consistently perform better than other feature sets, especially those that only contain syntactic or semantic information (SCF, CO). ScfCo and DR scale much better as we try to classify more verbs into more classes. In addition, ACO achieves better performance than the conventional CO on almost every m -way classification. All these results help reinforce the idea that both syntactic and lexical information are useful in automatic verb classification.
- Again, JOANIS07 does not match the performance of other feature sets that combine syntactic and lexical information, but yields similar accuracy as SCF, just like what we have observed for the experiments with Joanis15. Again, both syntactic and lexical information are helpful in automatic verb classification, but the key is how to effectively mix them together.

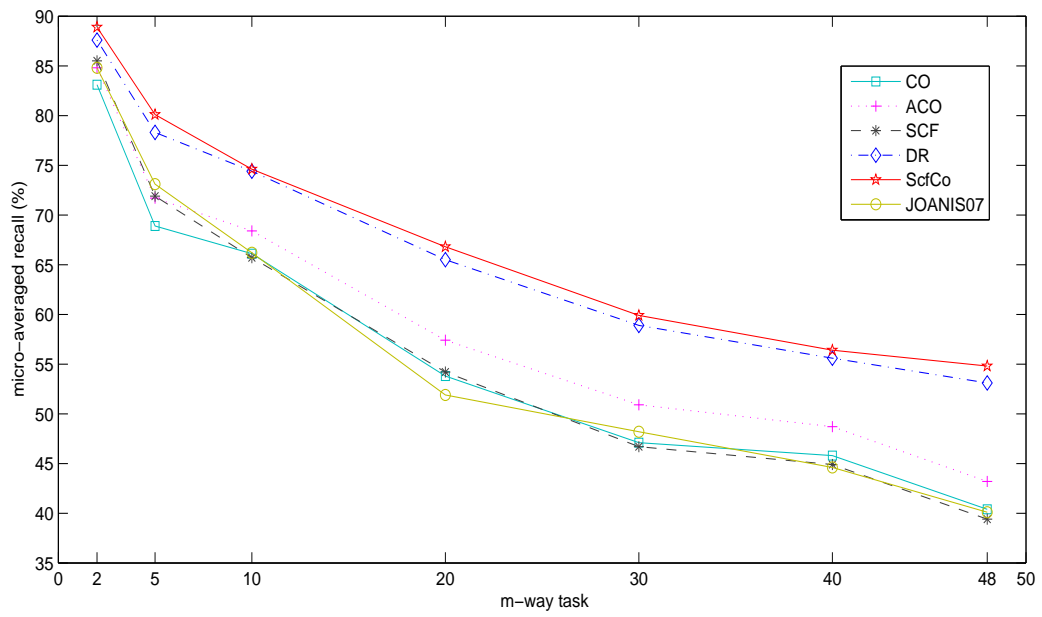
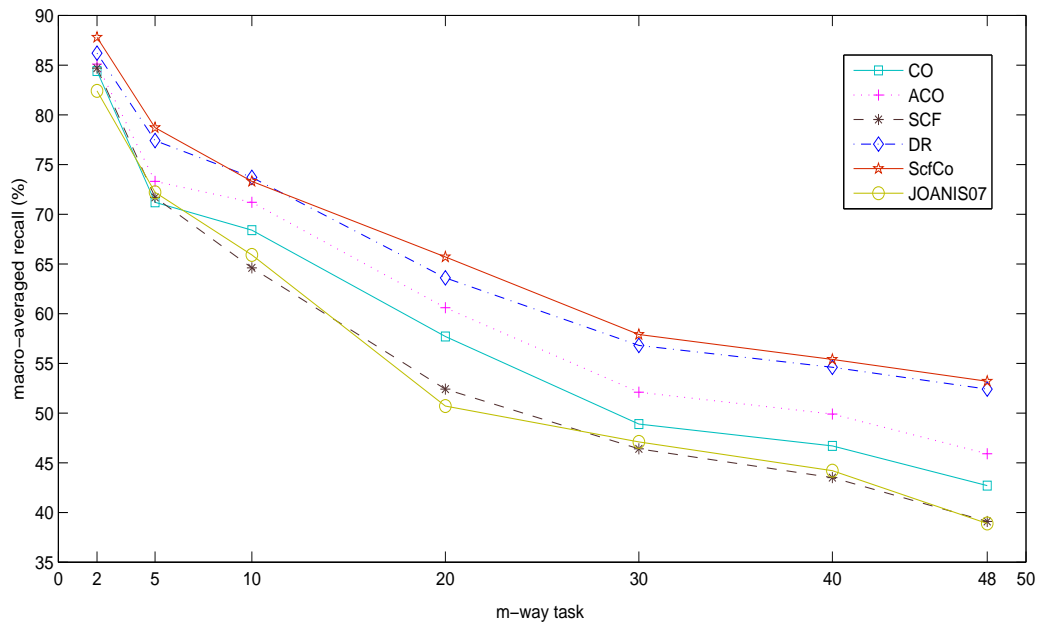


Figure 2: Classification results for Levin48

| Feature Selection | Feature Set | | | | |
|-------------------|-------------|-----------|-----------|-----------|-----------|
| | SCF | DR | CO | ACO | ScfCo |
| frequency | 39.8/40.7 | 52.4/53.1 | 42.7/40.9 | 45.9/43.2 | 53.2/54.8 |
| variance | 39.7/40.5 | 52.7/53.3 | 41.3/40.1 | 44.8/42.7 | 52.2/53.7 |

Table 13: Results (macro-averaged and micro-averaged recall) on the 48-way classification using frequency or variance for feature selection (%)

6 Further Discussion

We have performed a series of classification tasks, representing a wide range of class distinctions, with six different feature sets. Our conclusion is that subcategorization frames, at least on their own, are not very effective for automatic verb classification. In addition, both syntactic and lexical information prove useful in automatic verb classification. This conclusion is drawn on the classification experiments based on the following settings:

- *Feature Selection*: We choose the most frequent n features from each feature set.
- *Normalization Method*: We adopt the correlation method for data normalization.
- *Classifier*: We use BMR for classification, a 1-of- k classification method.
- *SCF Set*: We select all the frame types proposed by the CCG parser.

Below, we propose additional experiments to investigate how experimental results might be affected by a change in one of these settings. In particular, we want to see whether it still holds, with a change in experiment setting, that a mixed feature set combining syntactic and lexical information is still more effective than a feature set that contains only syntactic or lexical information. All the experiments below are performed on the 48-way classification task.

6.1 Feature Selection

In our experiments, we select features based on their frequency. Burgess and Lund (1997) suggest that when dealing with high-dimensional space, features can be reduced by eliminating all but the n features with the highest variance. In this way, it is hoped that the more informative columns are retained. We therefore conduct an additional experiment by using only the n features with the highest variance from each feature set. Again, we set n to be 100, 300, 500, and then any value between 500 and 5,000 at an interval of 500, and between 5,000 and 10,000 at an interval of 1,000. All other settings remain unchanged.

As shown in Table (13), using variance for feature selection does not have a significant effect on the accuracy achieved by each feature set. While using variance for feature selection yields a better result for DR than using frequency, it actually results in a decrease in accuracy for SCF, CO, ACO, and ScfCo. However, the change in either case is slight, and more importantly, ScfCo and DR consistently perform better than SCF and CO regardless of the method used for feature selection.

6.2 Normalization Technique

Following Rohde et al. (2004), we adopt correlation for data normalization to reduce the undue influence of outlier features. Table (14) lists a few alternative methods for data normalization which have been used when dealing with high-dimensional space.

| | |
|--------|--|
| row | $w'_{vf} = \frac{w_{vf}}{\sum_j w_{vj}}$ |
| column | $w'_{vf} = \frac{w_{vf}}{\sum_i w_{if}}$ |
| length | $w'_{vf} = \frac{w_{vf}}{\sum_j w_{vj}^2}^{1/2}$ |

Table 14: Normalization techniques

In this experiment, instead of correlation, we normalize the frequency matrix for each feature set with the three alternative normalization methods. Again, all other settings remain unchanged. Rohde et al. (2004) show that high-dimensional lexical semantic space normalized with correlation achieves better results on various lexical judgment tasks than with the other three methods. In our initial experiment, we obtain similar results. The three alternative normalization methods achieve an accuracy far below that obtained by the correlation method. However, the correlation method used in Rohde et al. (2004) is further maneuvered by using only the square root of the positive correlations, which they argue has the beneficial effect of magnifying the importance of the many small values relative to the few large ones. With our data set, most values obtained after row, column and length normalization tend to be very small, so we choose to use the square root of the row-, column- and length-normalized values. The results are summarized in Table (15).

| Normalization | Feature Set | | | | |
|---------------|-------------|-----------|-----------|-----------|-----------|
| | SCF | DR | CO | ACO | ScfCo |
| correlation | 39.8/40.7 | 52.4/53.1 | 42.7/40.9 | 45.9/43.2 | 53.2/54.8 |
| row | 39.8/41.4 | 54.7/56.9 | 42.3/40.1 | 47.4/44.8 | 53.7/55.1 |
| column | 34.3/34.1 | 44.6/46.7 | 39.9/35.8 | 40.4/36.2 | 45.4/46.8 |
| length | 37.7/38.8 | 46.0/45.8 | 37.9/35.1 | 42.4/39.7 | 47.1/46.5 |

Table 15: Results (macro-averaged /micro-averaged recall) on the 48-way classification using different data normalization methods (%)

Using the square roots has different effect on row, column and length normalization. While the accuracy achieved by using column and length normalization are still far below that achieved by using correlation, using row normalization actually renders slightly better results for all feature sets except CO than using correlation normalization. However, it is worth noting that regardless of the normalization method used, ScfCo and DR still consistently outperform SCF and CO.

6.3 Classifier

SVMs are a popular technique for classification (Cortes and Vapnik, 1995), and have been shown to be an effective and reliable classifier in various tasks, such as text categorization, POS tagging,

and WSD. In fact, Joanis et al. (2007) adopt SVMs for their automatic verb classification tasks. In this experiment, we will investigate whether ScfCo and DR still produce a better classification result than SCF and CO when a different classifier is used. Instead of BMR, we use SVMs for classification. Again, all other settings remain unchanged.

To begin with, given a training set of instance-labeled pairs (\mathbf{x}_i, y_i) , $i = 1, \dots, l$ where $\mathbf{x}_i \in R_n$ and $y \in \{1, -1\}^l$, SVMs require the solution of the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w}, b, \zeta} \quad & \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^l \zeta_i \\ \text{subject to} \quad & y_i (\mathbf{w}^T \phi(\mathbf{x}_i) + b) \geq 1 - \zeta_i \\ & \zeta_i \geq 0 \end{aligned} \tag{10}$$

Here training vectors \mathbf{x}_i are mapped into a higher dimensional space by the function ϕ . Then SVM finds a linear separating hyperplane with the maximal margin in this higher dimensional space. A test example is classified depending on the side of the hyperplane it lies in. Furthermore, a kernel function can be used to reduce the computational cost of training and testing in high dimensional space. If the training examples are not perfectly separable, a regularization parameter C can be used to control the trade-off between achieving a large margin and a low training error. We use the LIBSVM library for support vector machines described in Chang and Lin (2001), with the default setting recommended by Hsu et al. (2003):

- We use standard C-SVC Cost-Based Support Vector Classification model.
- We use radial basis function (RBF) kernel, $e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2}$
- LIBSVM involves tuning two parameters while using RBF kernels: C , and γ . For each task and feature set, we consider a range of values:
 - $C = 2^{-5}, 2^{-3}, \dots, 2^{15}$
 - $\gamma = 2^{-15}, 2^{-13}, \dots, 2^3$

and select the combination of C and γ that offers the best accuracy on the preprocessed training set according to 10-fold cross-validation.

| Classifier | Feature Set | | | | |
|------------|-------------|-----------|-----------|-----------|-----------|
| | SCF | DR | CO | ACO | ScfCo |
| BMR | 39.8/40.7 | 52.4/53.1 | 42.7/40.9 | 45.9/43.2 | 53.2/54.8 |
| SVMs | 40.7/41.9 | 48.6/51.0 | 41.1/40.2 | 42.8/41.4 | 50.2/52.4 |

Table 16: Results (macro-averaged and micro-averaged recall) on the 48-way classification using BMR and SVMs for classification (%)

The results from using BMR and SVMs are given in Table (16). Overall, SVMs do not perform as well as BMR on this task, although a slightly better result is obtained for SCF. Again, ScfCo and DR are still the most effective feature sets for our task regardless of the classifier (BMR or SVMs) used.

6.4 SCF Set

This section is devoted to a discussion on the effect of the choice of a SCF set on automatic verb classification. In particular, we will closely examine the relationship between Levin’s claim, her theoretical motivations and corpus data.

6.4.1 Subcategorization Information

First, Levin classifies English verbs based primarily on their syntactic behavior. Which are the most informative features in automatic verb classification?

Our experiments have revealed that SCFs, at least when used alone, compare poorly to those feature sets that mix syntactic and lexical information. However, one explanation for the poor performance of SCFs could be that we use all the frame types generated by the CCG parser. How will the classification results be affected if we adopt a frame set hand-selected with linguistic expertise?

6.4.2 Levin-defined Subcategorization Frames

Next, Levin pre-selects frames that display alternations. Are these frames important for automatic verb classification?

As discussed before, earlier work on automatic verb classification has preferred to use SCFs as verb features. However, they differ in the frame sets used. Like us, Schulte im Walde (2000) considers all 7,444 frame types generated by the parser she uses, but decides to limit the SCF set to the 88 frames which appear at least 2,000 times in the training corpus. Korhonen et al. (2003) use a set of 160 frames which incorporates those found in ANLT (Boguraev et al., 1987) and COMLEX (Grishman et al., 1994). In her classification, Levin defines 78 frames, which should, at least in principle, be good at separating verb classes. To see if Levin-selected frames are more effective for automatic verb classification, we perform an additional experiment using only the frames pre-selected by Levin. However, we do not assume any prior knowledge about whether a verb licenses a particular frame. In practice, we match each SCF generated for a Levin verb by the CCG parser (CCG-SCF) to one of the 78 Levin-selected frames regardless of whether the verb licenses that frame according to Levin. The matching process proceeds as follows: given a frame generated for a verb by the CCG parser, we first check if it is one of the 78 frames selected by Levin. If it is not, we remove the last element from the frame and check again. This process continues until a matching frame is found. We refer to frames derived this way as UnfilteredLevinSCF. We also apply a statistical test (relative frequency) to the UnfilteredLevinSCF to filter out noisy SCFs, and denote the resulting SCF set as FilteredLevinSCF. We then perform the 48-way classification task with three different SCF sets, and the results are presented in Table (17).

| SCF Set | | |
|-----------|--------------------|------------------|
| CCG-SCF | UnfilteredLevinSCF | FilteredLevinSCF |
| 39.8/40.7 | 39.7/41.0 | 40.5/41.1 |

Table 17: Results (macro-averaged/micro-averaged recall) on the 48-way task using Levin-selected frames (%)

Although a small gain in performance has been achieved by using Levin-selected frames, the accuracy is still far below that achieved by using a feature set that combines syntactic and semantic information. In fact, even the simple CO features yield a slightly better performance (42.1/41.5) than these expert-selected SCFs.

6.4.3 Frequency Information

Last, Levin classifies verbs based on whether a verb participates in a particular alternation, without regard to frequency. Does frequency information derived from the corpus benefit or hurt automatic verb classification?

To answer this question, we transform each raw count matrix by replacing any non-zero counts with 1. In the converted matrix, 0 means that a particular frame does not co-occur with a given verb, and 1 otherwise. Again, we perform the 48-way classification task with three different SCF sets, and the results are summarized in Table (18).

| Accuracy | SCF Set | | |
|-----------|-----------|--------------------|------------------|
| | CCG-SCF | UnfilteredLevinSCF | FilteredLevinSCF |
| binary | 37.7/39.5 | 35.6/36.4 | 38.2/39.7 |
| frequency | 39.8/40.7 | 39.7/41.0 | 40.5/41.1 |

Table 18: Results (macro-averaged/micro-averaged recall) on the 48-way task using Levin-selected frames with or without frequency information (%)

Table (18) shows that replacing frequency information with binary features actually leads to a drop in classification accuracy, suggesting that frequency information derived from the corpus benefits automatic verb classification.

6.4.4 Analysis

A clear picture emerging from these additional experiments is that the SCF set pre-selected by Levin is no more useful for automatic verb classification than the frame set proposed by parsers. This finding agrees with the results reported in Joanis et al. (2007). Joanis et al. (2007) compare their general feature space to subsets of their own features (called the Levin-defined subsets) which are hand-selected through an analysis of the classes in Levin. For each class, they systematically identify the subset of features indicated by the class description in Levin. For each experimental task, the Levin-derived subset is the union of these subsets of features for all the classes in the task. Table (19) lists the Levin-derived subset of these features for the BENEFACTIVE/RECIPIENT classification task.

The results from using the general feature space or Levin-derived subset of features are given in Table (20). It is clear that the general feature space actually performs slightly better than Levin-defined subsets.

Overall, the experiments we have performed all seem to suggest that subcategorization frames, at least when used alone, are not the most effective features for automatically deriving Levin-style verb classification despite their evident effectiveness in supporting Levin’s initial intuition. However, this should not be taken as an argument against Levin’s claim and theoretical motivations. We believe that there are a few factors that may be responsible for the results we have observed:

| Features | Class | |
|----------|--|---|
| | BENEFACTIVE | RECIPIENT |
| Slot | dir.object, ind.object PPfor, PPfrom PPinto, PPout of | dir.object, ind.object PPto, PPwith PP-groupbehind, PP-groupnear |
| Overlap | (subj, dir.object) (intr.subject, dir.object) (trans.subject, intr.subject) (trans.subject, PPfrom) (tran.subject, PPout of) (dir.object, PPfrom) (dir.object, PPinto) (dir.object, PPout of) | (subject, dir.object) (intr.subject, dir.object) (dir.object, PPwith) (ind.object, PPto) |
| Animacy | subject, trans.subject, dir.object ind.object, PPfor, PPinto, PPout of | subject, trans.subject, dir.object ind.object, PPto |

Table 19: Levin-derived subsets of features for BENEFACTIVE/RECIPIENT task (Joanis et al., 2007). (%)

| Experimental Task | Random Baseline | Feature Set | |
|-------------------------------|-----------------|-----------------------|---------------|
| | | General Feature Space | Levin-derived |
| Average (2-way tasks) | 50.0 | 83.2 | 78.8 |
| Average (3-way tasks) | 33.3 | 69.9 | 68.4 |
| Average (\geq 6-way tasks) | 12.1 | 61.1 | 57.4 |

Table 20: Experimental results using the general feature space and Levin-derived feature set (%)

- *There exist some discrepancies between Levin’s SCF definition for a particular verb and what is observed in the corpus data:* For a particular Levin verb, some SCFs, listed in Levin, fail to appear in the corpus data and some SCFs, attested in the corpus data, are not listed in Levin. For example, although the verb *deny* licenses the frame NP1-V-NP2-INF according to Levin, it does not co-occur with this frame in the corpus data. To the contrary, *deny* is attested with the frame NP1-V-NP2-PPto in the corpus data though it is not supposed to license this frame according to Levin.
- *SCFs are noisy features:* Both the parsing and filtering processes are prone to producing errors in automatically extracting SCFs from corpora, which may well have some negative effects on the results of the classification tasks.
- *SCFs are semantically impoverished features:* However, we believe that the primary reason for the ineffectiveness of SCFs in automatic verb classification is that SCFs lack information crucial for making lexical distinctions. In section 2.1, the example NP1-V-PPwith is provided to illustrate that SCFs, without any regard for lexical information, can easily obscure the different underlying meanings associated with the surface frames. The importance of lexical information in automatic verb classification gets highlighted by the experiment outcome that feature sets combining syntactic and lexical information generally improve over SCFs. For

instance, like SCFs, extraction of DRs also depend on parsers, and therefore also contain some noise in them. However, using DRs outperforms SCFs consistently on automatic verb classification. It is reasonable to attribute the effectiveness of DRs and ineffectiveness of SCFs in automatic verb classification to the presence of lexical information in DRs. As a matter of fact, lexical information does play a part in Levin verb classification even though syntactic alternations are the primary criteria. For example, Levin has to rely on her lexical knowledge as a linguist to make the distinction between the different underlying meanings of the frame NP1-V-PP*with*. Furthermore, such lexical knowledge also comes into play when several Levin classes can not be properly distinguished by the frame set they take. To give an example, the BUTTER, DESTROY, ILLUSTRATE, and PELT classes are all defined, according to Levin, by the frame set {NP1-V-NP2, NP1-V-NP-PP*with*}. To complicate the matter even more, the PP*with* can all be interpreted as *instrument* when verbs in these four classes take the frame NP1-V-NP-PP*with*. Apparently, verbs in these four classes cannot be successfully separated into coherent classes without the help of lexical knowledge. Unfortunately, such lexical knowledge is not immediately available in automatic verb classification. We therefore approximate this lexical knowledge by integrating into the feature space the lexical information in the simple form of neighboring words. In light of the classification results, this method helps overcome the deficiency of SCFs in achieving better classification performance.

7 Related Work

In this section, we give a broad overview of the related work on automatic verb classification. We start with papers on automatic classification of English verbs, followed by papers targeted at verbs in other languages or specific domains.

7.1 English Verbs

Merlo and Stevenson (2001) present an automatic classification of three types of English intransitive verbs including *unergatives*, *unaccusatives*, and *object-drop*. They select 20 verbs from each verb class, for a total of 60 verbs. However, verbs in these three selected classes show similarities with respect to their argument structure in that they can all be used as transitives and intransitives. Therefore, syntactic cues alone cannot effectively distinguish the classes. Merlo and Stevenson define five linguistically-motivated verb features that describe the thematic relations between subject and object in transitive and intransitive usage. These features are collected from an automatically tagged corpus (primarily Wall Street Journal). Each verb is represented as a five-feature vector on which a decision tree classifier is trained. They achieve 69.8% accuracy for a task with a baseline of 33.3%, and an expert-based upper bound at 86.5%.

The experiments conducted in Merlo and Stevenson (2001) yield promising results on automatic verb classification, but deep linguistic expertise is required to identify the five verb features, which are crucial for the success of the classification experiments. The need for such linguistic expertise limits the applicability of the method because these features are designed specifically for the particular class distinctions investigated, and are unlikely to be effective when applied to other classes. Later work has proposed an analysis of possible class distinctions exhibited by Levin verbs, which generalizes Merlo and Stevenson’s features to a larger space of features that potentially cover any verb classes (Joanis, 2002; Joanis and Stevenson, 2003; Joanis et al., 2007). The features they use

fall into four different groups: *syntactic slots*, *slot overlaps*, *tense*, *voice*, and *aspect*, and *animacy of NPs*. These features are extracted from the BNC using the chunker described in Abney (1991). The feature space they design is *general* - potentially applicable to any class distinction among Levin classes; *broad* - it taps into various semantic features of Levin verbs; and *inexpensive* - requiring no more than a POS tagger and a chunker. Joanis et al. (2007) present experiments on classification tasks involving 15 verb classes and 835 verbs from Levin using SVMs with the proposed feature space. Their experiments achieve a rate of error reduction ranging from 48% to 88% over a chance baseline across classification tasks of varying difficulty. In particular, using the general feature space they devise yields classification accuracy comparable to or even better than using the feature sets manually selected for each particular task.

Schulte im Walde (2000) clusters 153 English verbs into 30 Levin classes. 103 verbs only have a single sense, 35 verbs have two senses, 9 verbs have three senses, and 6 verbs have four senses. Each verb is represented by distributions over subcategorization frames extracted from the BNC using a robust statistical parser (Carroll and Rooth, 1998). An unsupervised hierarchical clustering is employed, which assigns each verb to only one verb class. Schulte im Walde investigates the linguistic conditions crucial for automatic verb classification by evaluating the effectiveness of subcategorization frames in three different forms: 1) syntactic frames, which are relevant to capturing argument alternations (e.g. NP-V-PP); 2) prepositions, which are able to distinguish, e.g., directions from locations (e.g. NP-V-PP*into*, NP-V-PP*on*); and 3) selectional preferences, which encode participant roles (e.g. NP(PERSON)-V-PP*on*(LOCATION)). Using Levin verb classification as a basis for evaluation, 61% of the verbs are correctly classified into semantic classes. The best clustering result is achieved when using subcategorization frames enriched with PP information. Adding selectional preferences actually decreases the clustering performance. This is probably due to the data sparsity that result from the encoding of selectional preferences.

Korhonen et al. (2003) present an investigation on English verb classification, but concentrate more on polysemous verbs. They employ as a gold standard an extended version of Levin verb classification constructed by Korhonen (2003).⁷ 110 test verbs are chosen, most of which belong to more than one verb class. They obtain subcategorization frame frequency information from the BNC which is extracted automatically using the parser described in Briscoe and Carroll (1997), and then apply two clustering methods: 1) a simple hard method that collects the nearest neighbor of each verb, and 2) an iterative method based on information bottleneck. Neither of these clustering methods allows the assignment of a single verb to multiple verb classes. A novel aspect of their investigation is that it reveals the impact that polysemy tends to have on the clustering results. For example, polysemous verbs with a clear predominant sense and those with similar regular polysemy⁸ are frequently classified together, but verbs with irregular polysemy tend to resist any classification and are likely to be assigned to singleton clusters.

⁷This incorporates Levin’s classes, 26 additional classes by Dorr (1997), and 57 new classes for verb types not covered comprehensively by Levin and Dorr.

⁸A verb displays regular polysemy if it shares its full set of Levin class memberships with at least one other verb. Dang et al. (1998) defines regular polysemy as intersective Levin classes obtained by grouping together subsets of existing Levin classes with at least two overlapping members. Such intersective classes have more coherent sets of syntactic frames and associated semantic components. Likewise, a verb displays irregular polysemy if it does not share its full set of Levin class memberships with any other verbs.

7.2 Cross-linguistic Investigation

For many languages, no Levin-style lexical resources for verbs have been yet created. Assuming that Levin’s analysis is generally applicable across languages in terms of the linking of semantic arguments to their syntactic expressions, a substantial amount of work has been done to derive Levin-style verb classification for various languages including German (Schulte im Walde and Brew, 2002; Brew and Schulte im Walde, 2002; Schulte im Walde, 2003, 2006), Japanese (Oishi and Matsumoto, 1997), Portuguese (Dang et al., 1998), and Italian (Merlo et al., 2002).

Schulte im Walde (2006) proposes clustering experiments on German verbs. She uses a statistical parser as described in Schmid (2000) to extract subcategorization information from a large collection of German newspaper corpora. Similar to Schulte im Walde (2000), linguistic conditions are assessed which are crucial for automatic verb classification: 1) syntax frames; 2) syntax frames with PP information; and 3) syntax frames with selectional preferences. An unsupervised clustering, *k*-means algorithm, is used to perform an automatic induction of verb classes. A full range of experiments are conducted to investigate the technical parameters for the *k*-means clustering that are suitable for automatic verb classification, such as initialization and distance measures. The clustering methodology is tuned on a small set of verbs and then adapted to a larger-scale verb set including 883 German verbs and 43 verb classes. Experimental results show that PP information improves the accuracy of verb clustering, and enriching hand-picked syntactic slots with selectional preferences yields the best clustering performance.

Oishi and Matsumoto (1997) also use unsupervised learning to automatically cluster Japanese verbs into semantic classes. Like Joanis et al. (2007), they use a combination of syntactic frames and aspectual features for classification. They first classify 835 Japanese verbs along two dimensions: thematic and aspectual. In the thematic dimension, patterns of case marking particles, which can be obtained by gathering syntactic argument structures of verbs in a corpus, are adopted as features. In the aspectual dimension, adverbs, which indicate various event types, are used as features. They then combine the results into a network of 38 classes using linguistic knowledge and semi-automated processing. Examination of the resulting verb classification validates the existence of close relations between verb meaning and verb behavior.

7.3 Domain-Specific Study

All the studies described in sections 7.1 and 7.2 have concentrated on deriving general-purpose verb classifications, like Levin and VerbNet. Although these general-purpose verb classifications have benefited various NLP tasks, they are not suitable for domain specific applications because verb meanings and behaviors are likely to vary across domains. To our knowledge, Korhonen et al. (2006) is the only study that deals with domain-specific verb classification.

Korhonen et al. (2006) propose an experiment on automatic clustering of English verbs in biomedical texts. To gain an initial idea of the differences between the biomedical classification and the general language classification, they examine the extent to which verbs and their frequencies differ in biomedical and general language texts. To do this, they compare the distribution of verbs in their biomedical corpus against that in the BNC. The Spearman rank correlation between the 1,165 verbs that occurred in both corpora is a weak one: 0.37. To evaluate the resulting classification from their experiments, they manually create 50 verb classes as the gold standard for the biomedical domain. Among the manually-created classes, 33 (out of 50) classes are biomedical, 17 (out of these 33) classes are unrelated to any Levin classes and 16 (out of these 33) bear vague resemblance

to Levin classes; 17 (out of 50) are of the general class, but 4 of them are absent from Levin classification. They show that the methodology developed in Korhonen et al. (2003) for deriving general-purpose verb classifications works equally well for the biomedical domain. This study helps highlight the importance of building and tuning lexical information specific to different domains.

8 Summary

8.1 Best Features for Automatic Verb Classification

Complex information about verbs is needed to support semantic processing of predicates and determine how participants are related. Despite the existence of several manually created verb lexicons, automatic verb classification is still necessary for extending existing verb lexicons, building and tuning lexical information specific to different domains, and bootstrapping verb lexicons for new languages. However, appropriate verb features must be identified and demonstrated to be effective for automatic verb classification. In this chapter, we have performed a wide range of experiments to assess the linguistic conditions that are crucial for automatic verb classification. Our experiments, at least to our knowledge, comprise the largest investigation on automatic classification of English verbs. While earlier work has focused on a small number of verbs or a small set of verb classes, we have tested the applicability of different features up to the distinction of 48 Levin classes involving about 1,307 verbs. We believe that this wide range of classification experiments helps shed light on the applicability and scalability of different feature sets.

- Both syntactic and lexical information are useful for automatic verb classification. SCFs, although supporting Levin’s initial intuition, prove inadequate in separating Levin classes in automatic verb classification due to their disregard of lexical information. Earlier attempts at integrating lexical information into the feature space fail to improve over using SCFs. In this study, we propose several ways for combining syntactic and lexical information, which turn out to be more effective for automatic verb classification. Although neither syntactic (SCF) nor lexical (CO) features perform well on their own, a combination of them proves to be very informative for this task. Other ways of mixing syntactic and lexical information (DR and ACO), work relatively well too. What makes these mixed feature sets even more appealing is that they tend to scale well when attempting to classify more verbs into a larger number of classes.
- Levin pre-selects 78 frames in supporting her verb classification. These frames should, at least in principle, be good at distinguishing verb classes. If expert-selected frames perform much better, then it may be unavoidable to use linguistic expertise when high performance is needed. However, we have shown that expert-selected frames fail to significantly improve over parser-proposed frames. In fact, all mixed feature sets we propose that include both syntactic and lexical information consistently outperform frames pre-selected by Levin. Even the simple CO features yield a slightly better performance. All these results indicate that expert-selected frames are not necessary for high performance for automatic verb classification.
- The simple CO features are generally considered to convey only lexical information and are not sensitive to argument structures. It is therefore surprising to observe that CO features perform as well as SCF features. As a matter of fact, CO features exhibit a greater potential for scaling well to a larger number of class distinctions in comparison to SCF features. This

result seems to suggest that verbs in the same Levin class tend to share their neighboring words even though they are not intended to be substitutable for each other in exactly the same local contexts.

The feature sets we have proposed are devised on a general level without relying on any knowledge about specific classes, thus having the potential to be 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.

8.2 Linguistic Theory and Automatic Lexical Acquisition

Like many other tasks of automatic lexical acquisition, our experiments on automatic verb classification have been guided by linguistic theories in lexical semantics. However, if a theory-guided lexical acquisition task fails to reveal, or worse yet, contradicts what linguistic theories have predicted, is it due to the deficiencies of the theories, of the data, or of the techniques? This question probably needs to be answered on a task-by-task basis, but we believe that experiments on automatic lexical acquisition can generally provide great insights into linguistic theories.

First, information extracted from corpus data can be used to test linguistic theories by taking into account realistic examples that do not constitute idealizations of linguistic phenomena. Linguistic theories generally disregard many of the questions arising in the process of automatic lexical acquisition as issues that should come under the rubric of linguistic performance. However, these problems are usually more wide-spread and significant than originally thought, therefore deserving an adequate explanation from linguistic theories. Take the *dative* alternation as an example. Lexical semantic theories, like the Proto-Role Hypothesis proposed in Dowty (1991), predict that verbs of PREVENTION or POSSESSION (*cost*, *deny*) are not allowed to occur in the dative frame (NP1-V-NP2-PP_{to}). Early research on lexical semantics has described this class as occurring in only the double-object frame (NP1-V-NP2-NP3) (Levin, 1993; Green, 1974). However, contrary to the prediction made by linguistic theories, a large-scale acquisition of subcategorization frames has revealed that this class occurs in both frames. Linguistic theories can not simply dismiss the occurrence of this class in dative frames as sporadic errors, since the frequency ratio between this class occurring in dative frames and double object frames is about 1:9.⁹ This observation seems to indicate that the mapping between semantic classes of dative verbs and alternative syntactic constructions should rest on probabilistic biases rather than strict categories, which is in agreement with the claim made by Manning (2003) that a linguistic theory should allow for a non-categorical representation and explanation.

Second, corpus data can also be used to refine or extend linguistic theories. Recent research in computational lexical acquisition (Dang et al., 1998) has suggested that verb classes are not simply the product of automatic applications of a set of rules about participation in alternations, but are at least partially semantically motivated. In Levin verb taxonomy, it seems to be implied that all of the distinctions made are of equal validity. However, some alternation patterns do not reflect the inherent semantics of the verb very well. An example of this is Levin’s verb classes of social interaction: CORRESPOND, MARRY and MEET verbs (Baker and Ruppenhofer, 2002).

⁹This has been checked by hand. A few examples: i) *It will cost jobs to the Californians in the televised profession.*
ii) *The employer has denied a leave to the teacher.*

| Alternation | Example |
|-------------------------------|--|
| Collective Subject NP | <i>The couple bantered/married/met</i> |
| Simple Reciprocal Alternation | <i>Pat bantered/*married/met with Kim</i> <i>Pat and Kim bantered/married/met</i> |
| Understood Reciprocal Object | <i>Pat *bantered/married/met Kim</i> <i>Pat and Kim bantered/married/met</i> |

Table 21: Verbs of social interaction

These three classes are defined syntactically by alternations reflective of the notion of *reciprocity* (see table 21). Unfortunately, the alternations that Levin describes as characteristic of this verb class are not in fact diagnostic of *reciprocity*. As a result, verbs that show similar alternation patterns, but denote actions of the participants that are not directed at each other, can be potentially misclassified into this class, such as *jog*, *eat*. A strict reliance on syntactic alternations for classification leads Levin to posit some broad, heterogeneous, and semantically very abstract classes. It is at least worth considering whether it makes sense to bring in more semantic knowledge for fine-tuning such broad classes into a few classes that are more homogeneous. For example, among Levin’s CORRESPOND verbs, some are focused on verbal communication, such as *argue*, *bicker*, *chat*, and *gossip*, others more on actions, such as *compete*, *struggle* and *contest*.

Two possible lessons about the relationship between theoretical and computational linguistics could be drawn from this. On the one hand, modestly, we could conclude that computational linguists should be wary of over-interpreting theoretical claims as prescriptions for engineering success. On the other hand, less modestly, the computational linguists may wish to point to the merits of a systematic and exhaustive engineering exploration as tool for exposing deficiencies and incompletenesses in the theory.

Last, lexical information acquired from corpus data can also contribute to the discovery of linguistic facts that complement the linguist’s introspection. According to the theory, verbs in the same Levin class are supposed to have very similar alternation behavior. We observe this, as predicted. But we also observe some similarities not predicted by the theory, namely that verbs in the same class also exhibit non-random contextual similarities frequently taken as indicative of a degree of synonymy. These observations are not in conflict with the theory, but should moderate inflated expectations about its usefulness and general applicability. While Levin’s theoretical argument has been extremely influential in computational linguistics, it should not be mistaken for a prescription for engineering success. Rather, it should be taken at face value, as a well-constructed argument, carefully supported by suitably marshalled evidence, for the relevance and importance of a set of theoretical ideas. In order to produce effective technology, theoretical insights must be complemented by careful engineering, thorough exploration and analysis of a wide range of possible designs.

It is not the direct concern of the computational linguist whether corpus-based insights into the evidential basis for linguistic theories have any impact on future theory, but even insights that the theoreticians will probably ignore can aid NLP tasks. Had we not noticed that different members of Levin classes share contextual features, we would not have considered the possibility that the contexts associated with the unambiguous members can be treated as training material for the classifier developed in (Li and Brew, 2007).

References

- Abney, S. (1991). Parsing by chunks. *Principle based parsing*.
- Baker, C., Fillmore, F., and John, L. (1998). The Berkeley FrameNet Project. In *Proceedings of the 17th International Conference on COLING and the 36th Annual Meeting of the ACL*, pages 86–90, Montreal, Canada.
- Baker, C. and Ruppenhofer, J. (2002). FrameNet’s frames and Levin’s verb classes. In *Proceedings of the 28th Annual Meeting of the Berkeley Linguistics Society*, pages 27–38, Berkeley, CA.
- Boguraev, B., Briscoe, E., Carroll, J., Carter, D., and Grover, C. (1987). The derivation of a grammatically-indexed lexicon from the Longman Dictionary of Contemporary English. In *Proceedings of the 25th Annual Meeting of ACL*, pages 193–200.
- Brew, C. and Schulte im Walde, S. (2002). Spectral clustering for German verbs. In *Proceedings of the 2002 Conference on EMNLP*, pages 117–124.
- Briscoe, T. and Carroll, J. (1997). Automatic extraction of subcategorization from corpora. In *Proceedings of the 5th ACL Conference on Applied Natural Language Processing*, pages 356–363.
- Burgess, C. and Lund, K. (1997). Modelling parsing constraints with high-dimensional context space. *Language and Cognitive Processes*, 12(3):177–210.
- Carroll, G. and Rooth, M. (1998). Valence induction with a head-lexicalized PCFG. In *Proceedings of the 1998 Conference on EMNLP*, pages 58–63.
- 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.
- Cortes, C. and Vapnik, V. (1995). Support vector network. *Machine Learning*, 20:273–297.
- Dang, H., Kipper, K., Palmer, M., and Rosenzweig, J. (1998). Investigating regular sense extensions based on intersective Levin classes. In *Proceedings of the 17th International Conference on COLING and 36th Annual Meeting of ACL*, pages 293–299, Montreal, Quebec, Canada.
- Dorr, B. (1997). Large-scale dictionary construction for foreign language tutoring and interlingual machine translation. *Machine Translation*, 12(4):1–55.
- Dowty, D. (1979). *Word Meaning and Montague Grammar*. Reidel, Dordrecht.
- Dowty, D. (1991). Thematic proto-roles and argument selection. *Language*, 67:547–619.
- Genkin, A., Lewis, D., and Madigan, D. (2004). Large-scale Bayesian Logistic Regression for text categorization. *DIMACS Technical Report*.
- 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.
- Grishman, R., Macleod, C., and Meryers, A. (1994). Complex syntax: Building a computational lexicon. In *Proceedings of the 15th International Conference on COLING*, pages 268–272.
- Hsu, C., Chang, C., and Lin, C. (2003). A practical guide to support vector classification. <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- Joanis, E. (2002). Automatic verb classification using a general feature space. Master’s thesis, University of Toronto.
- Joanis, E. and Stevenson, S. (2003). A general feature space for automatic verb classification. In *Proceedings of the 2003 Conference of EACL*, pages 163–170.
- Joanis, E., Stevenson, S., and James, D. (2007). A general feature space for automatic verb classification. *Natural Language Engineering*, 1:1–31.
- Kipper, K., Dang, H. T., and Palmer, M. (2000). Class-based construction of a verb lexicon. In *AAAI/IAAI*, pages 691–696.
- Kipper, K., Korhonen, A., Ryant, N., and Palmer, M. (2006). A large-scale extension of VerbNet with novel verb classes. In *Proceedings of the 12th EURALEX International Congress*, Turin, Italy.
- Klavans, J. and Kan, M. (1998). The role of verbs in document analysis. In *Proceedings of the 17th International Conference on COLING*, pages 680–686, Montreal, Canada.
- Korhonen, A. (2002). *Subcategorization Acquisition*. PhD thesis, Cambridge University.
- Korhonen, A. (2003). Extending Levin’s classification with new verb classes. Unpublished manuscript.
- 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 64–71, Sapporo, Japan.
- Leacock, C., Chodorow, M., and Miller, C. (1998). Using corpus statistics and WordNet relations for sense identification. *Computational Linguistics*, 24(1):147–165.
- Levin, B. (1993). *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press, Chicago, 1st edition.

- Li, J. and Brew, C. (2007). Disambiguating Levin verbs using untagged data. In *Proceedings of Recent Advances in Natural Language Processing*, Borovets, Bulgaria.
- Lin, D. (1998). Automatic retrieval and clustering of similar words. In *Proceedings of the 17th International Conference on COLING and 36th Annual Meeting of ACL*, pages 768–774.
- Lowe, J., Baker, C., and Fillmore, C. (1997). A frame-semantic approach to semantic annotation. In *Proceedings of 1997 SIGLEX Workshop/ANLP97*.
- Madigan, D., Genkin, A., Lewis, D., and Fradkin, D. (2005). Bayesian Multinomial Logistic Regression for author identification. *DIMACS Technical Report*.
- Manning, C. (2003). Probabilistic syntax. *Probabilistic Linguistics*, pages 289–341.
- 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.
- Merlo, P., Stevenson, S., Tsang, V., and Allaria, G. (2002). A multilingual paradigm for automatic verb classification. In *Proceedings of the 40th Annual Meeting of the ACL*, pages 207–214, Philadelphia, PA.
- Minnen, G., Carroll, J., and Pearce, D. (2000). Applied morphological processing of English. *Natural Language Engineering*, 7(3):207–223.
- Oishi, A. and Matsumoto, Y. (1997). Detecting the organization of semantic subclasses of Japanese verbs. *International Journal of Corpus Linguistics*, 2(1):65–89.
- Pado, S. and Lapata, M. (2007). Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199.
- Palmer, M., Gildea, D., and Kingsbury, P. (2005). The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.
- Pinker, S. (1989). *Learnability and Cognition: The Acquisition of Argument Structure*. MIT Press, Cambridge, MA.
- Pustejovsky, J. (1991). The generative lexicon. *Computational Linguistics*, 17(4).
- Rohde, D., Gonnerman, L., and Plaut, D. (2004). An improved method for deriving word meaning from lexical co-occurrence. manuscript.
- Schmid, H. (2000). Lopar: Design and implementation. <http://citeseer.ist.psu.edu/schmid00lopar.html>.
- 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.

- Schulte im Walde, S. (2006). Experiments on the automatic induction of German semantic verb classes. *Computational Linguistics*, 32(2):159–194.
- Schulte im Walde, S. and Brew, C. (2002). Inducing German semantic verb classes from purely syntactic subcategorization information. In *Proceedings of the 40th Annual Meeting of ACL*, pages 223–230.
- Schutze, H. (1998). Automatic word sense disambiguation. *Computational Linguistics*, 24(1):97–124.
- Shi, L. and Mihalcea, R. (2005). Put pieces together: Combining FrameNet, VerbNet and WordNet for robust semantic parsing. In *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing*, pages 100–111.
- Swift, M. (2005). Towards automatic verb acquisition from VerbNet for spoken dialog processing. In *Proceedings of the Interdisciplinary Workshop on the Identification and Representation of Verb Features and Verb Classes*, pages 115–120.
- Vendler, Z. (1957). Verbs and times. *Philosophical Review*, 56:143–160.