Natural Language Engineering **16** (4): 391–415. © Cambridge University Press 2010
doi:10.1017/S1351324910000136

Class-based approach to disambiguating Levin verbs

JIANGUO LI[1] and CHRIS BREW[2]

[1]*Applied Research Center, Motorola, Schaumburg, IL 60196, USA*
email: jianguo.li@motorola.com
[2]*Department of Linguistics, Department of Computer Science and Engineering, The Ohio State University, Columbus, OH 43201, USA*
email: cbrew@ling.osu.edu

(*Received 15 July 2009; revised 28 February 2010; accepted 13 May 2010*)

Abstract

Lapata and Brew (*Computational Linguistics*, vol. 30, 2004, pp. 295–313) (hereafter LB04) obtain from untagged texts a statistical prior model that is able to generate class preferences for ambiguous Lewin (*English Verb Classes and Alternations: A Preliminary Investigation*, 1993, University of Chicago Press) verbs (hereafter Levin). They also show that their informative priors, incorporated into a Naive Bayes classifier deduced from hand-tagged data (HTD), can aid in verb class disambiguation. We re-analyse LB04's prior model and show that a single factor (the joint probability of class and frame) determines the predominant class for a particular verb in a particular frame. This means that the prior model cannot be sensitive to fine-grained lexical distinctions between different individual verbs falling in the same class.

We replicate LB04's supervised disambiguation experiments on large-scale data, using deep parsers rather than the shallow parser of LB04. In addition, we introduce a method for training our classifier without using HTD. This relies on knowledge of Levin class memberships to move information from unambiguous to ambiguous instances of each class. We regard this system as unsupervised because it does not rely on human annotation of individual verb instances. Although our unsupervised verb class disambiguator does not match the performance of the ones that make use of HTD, it consistently outperforms the random baseline model. Our experiments also demonstrate that the informative priors derived from untagged texts help improve the performance of the classifier trained on untagged data.

1 Introduction

Much research in lexical acquisition of verbs has concentrated on the relation between verbs and their argument frames. Theoretical linguists claim that the behavior of a verb, particularly with respect to the expression of arguments and the assignment of thematic roles is to a large extent driven by deep semantic regularities (Green 1974; Dowty 1991; Levin 1993; Goldberg 1995). To pick one example, Levin's (Levin 1993; hereafter Levin) leading hypothesis is that verb classes can be distinguished, at least in part, by patterns in the distribution of verb frames. Levin is particularly concerned with pairs of syntactic frames that express the same (or nearly the same) meaning (these pairs are often referred to as diathesis alternations). Her

https://doi.org/10.1017/S1351324910000136 Published online by Cambridge University Press

primary goal in undertaking extensive study of English verbs is to support theoretical claims about the interaction between different subsystems of the grammar (syntax and lexical semantics).

Levin's theoretical focus is therefore on verbs for which distribution of syntactic frames is a useful indicator of class membership, and, correspondingly, on classes which are relevant for such verbs. Levin's choices are determined primarily by the needs of her theoretical argument, so any coverage of verbs that are irrelevant to these needs is, from the perspective of natural language engineering, a happy accident.

In choosing to use Levin's verb classification, we therefore obtain a window on some (but not all) of the potentially useful semantic properties of some (but not all) of the verbs that a natural language system is likely to encounter. The verb classification can still help reduce redundancy in verb descriptions and enable generalizations across semantically similar verbs. This prospect has motivated a recent wealth of work on automatic acquisition of verb lexicons from corpus texts (Merlo and Stevenson 2001; Schulte im Walde 2000; Korhonen, Krymolowski and Marx 2003; Joanis, Stevenson and James 2007). Most of this work is concerned with the placement of a particular verb into a class as defined in Levin's verb taxonomy. Thus, the focus has been more on verb types than tokens. In this paper, we move beyond type-based verb classification and explore token-level disambiguation of Levin verbs. We demonstrate that Levin's classes can be used to achieve effective corpus-based disambiguation of verb instances, and that this can be done without benefit of sense-tagged corpora.

1.1 Levin's verb sense inventory

Levin classifies 3,104 English verbs into 191 verb classes, based on diathesis alternations. Of the 3,104 verbs she covers, 784 are listed as belonging to more than one class. As a result, a verb sense inventory falls out immediately from her classification. To the extent that her main hypothesis is correct, verbs in each class will share certain meaning components, thus constituting a distinctive verb sense. Verbs that belong to more than one class are considered ambiguous. There are two different types of ambiguous Levin verbs:

Pseudo-ambiguous: Verbs whose class membership can be guessed correctly solely on the basis of the subcategorization frame it occurs in. Consider for instance the verb *serve*, which is a member of four Levin classes: GIVE, FIT, MASQUERADE, and FULFILLING. Each of these classes can in turn license one of four distinct syntactic frames.

Example 1.1 GIVE: They can *serve* our guests a good meal. [NP1-V-NP2-NP3]
Example 1.2 FIT: The airline *serves* 164 destinations. [NP1-V-NP2]
Example 1.3 MASQUERADE: He *served* Napoleon as minister of the interior. [NP1-V-NP2-PP*as*]
Example 1.4 FULFILLING: She *served* an apprenticeship to a still-life photographer. [NP1-V-NP2-PP*to*]

https://doi.org/10.1017/S1351324910000136 Published online by Cambridge University Press

Genuinely-ambiguous: Verbs for which the frame information does not provide sure-fire cues for class disambiguation. An example is given in Example 2 about the verb *call* where it occurs in a double object frame, but belongs to the class DUB in Example 2.1 and GET in Example 2.2.

Example 2.1 DUB: He *called* John a fool. [NP1-V-NP2-NP3]
Example 2.2 GET: He *called* John a cab. [NP1-V-NP2-NP3]

Earlier research on token-level disambiguation of verbs has generally adopted the verb sense inventory of WordNet. WordNet is an online lexical database of English that currently contains approximately 120,000 sets of noun, verb, adjective, and adverb synonyms, each representing a lexicalized concept. A synset (synonym set) contains, besides all word forms that can refer to, a given concept, a definitional gloss and—in most cases—an example sentence. Words and synsets are organized in a hierarchical structure in which they are interrelated by means of lexical and semantic-conceptual links, respectively. WordNet is designed principally as a semantic network, and contains little syntactic information. Levin's verb sense inventory is different from that of WordNet in several respects. In particular, Levin's verb sense inventory employs different criteria to make sense distinctions of verbs. Palmer (2000) argues that computational lexicons should support sense distinctions based on three concrete criteria:

(1) different predicate argument structures;
(2) different selectional restrictions on verb arguments;
(3) co-occurrences with different lexical material, especially prepositions.

Levin's verb sense inventory meets Palmer's requirement for concreteness. It relies mainly on diathesis alternations and syntactic frame patterns as the criterion for discriminating verbs, and applies this criterion with great consistency. However, Levin's verb classification has its own limitations. For example, Levin's verb classification is certainly not intended as an exhaustive description of English verbs, their meanings or their behavior. A different grouping might, for example, have occurred if finer or coarser semantic distinctions were taken into account (Dang *et al.* 1998; Merlo and Stevenson 2001). As pointed out by Kipper, Dang and Palmer (2000), Levin classes also exhibit inconsistencies, and verbs are listed in multiple classes, some of which have conflicting sets of syntactic frames. Hence, some ambiguities may also arise as a result of accidental errors or inconsistencies, and some ambiguities may be missing due to incomplete entries. Despite its imperfection, Levin's verb classification has been shown to be valuable in aiding various Natural Language Processing (NLP) tasks including lexical resource construction (Korhonen 2002), semantic parsing (Shi and Mihalcea 2005), and semantic role labeling (Gildea and Jurafsky 2002).

In this paper, we focus on token-level disambiguation of Levin verbs. Our main reason is that successful disambiguation against this classification will credibly offer a number of benefits to NLP:

- At the syntactic level, accurate class information provides partial information about the subcategorization frames and alternations that are allowed. This,

https://doi.org/10.1017/S1351324910000136 Published online by Cambridge University Press

for example, could be used to improve the quality of subcategorization data automatically acquired from corpora (Korhonen 2002; Korhonen and Preiss 2003).

- At the semantic level, knowing a token's class helps determine the thematic roles of its arguments (Merlo and Stevenson 2001; Swier and Stevenson 2004). Information about the thematic roles of a verb's arguments is a key factor in the performance of semantic analysers. Such analysers, enriched with lexical semantic information, can then be employed in applications such as question answering (Shen and Lapata 2007).

1.2 A class-based approach to disambiguating Levin verbs

In this paper, the task is to disambiguate an instance of an ambiguous Levin verb, given knowledge of its frame and surrounding context. Throughout, we assume the availability of an automatic parser capable of detecting frames with sufficient accuracy. We explore various options for this component, and previous work has covered others.

Formally, the disambiguation task is as follows. Given a Levin verb v and a particular frame f and a set of Levin classes $C(v, f) = \{c_1, \ldots, c_k\}$, which is compatible with v and f we must choose class label $c_i \in C$. For example, in Examples 2.1 and 2.2, when the verb *call* occurs in a double object frame, we want to determine whether it is a DUB verb or GET verb. The disambiguation of pseudo-ambiguous verbs is trivial if f has been accurately determined, so we do not discuss it further.

Word Sense Disambiguation (WSD) is usually cast as a problem in supervised learning, where a word sense disambiguator is induced from hand-tagged data (HTD). The context within which an ambiguous word occurs is typically represented by a set of linguistically motivated features from which a learning algorithm induces a representative model that performs the disambiguation task. One classifier that has been extensively used is the Naive Bayes classifier. This is described by (1):

$$Posterior = \frac{Prior \times Likelihood}{Evidence} \tag{1}$$

In WSD, since the denominator is constant for all senses, the problem is reduced to finding the sense that maximizes the value of the numerator, which consists of two parts, prior and likelihood. Prior represents how likely a given sense is to occur in general, which corresponds to the proportion of each sense in a sense-tagged corpus. Likelihood amounts to the probability of a sense given some specific context.

In disambiguating Levin verbs, Lapata and Brew (2004) (hereafter LB04) estimate an informative prior over Levin verb classes for an ambiguous verb in a particular frame, training on untagged texts. Their prior model is able to generate a class preference for an ambiguous verb. To compute the likelihood, LB04 uses contextual features (e.g. word collocation, word co-occurrence) extracted from a small hand-tagged corpus. The contribution of LB04 is to highlight the importance for WSD of a suitable prior derived from untagged text:

https://doi.org/10.1017/S1351324910000136 Published online by Cambridge University Press

- A prior model derived from untagged texts can help find with reasonable accuracy the predominant sense of an ambiguous word. Knowing the predominant sense of a target ambiguous word is valuable as the first sense heuristic which usually serves as a baseline for supervised WSD systems outperforms many of these systems which take the surrounding contexts into account. McCarthy *et al.* (2004) have also recently demonstrated the usefulness of a suitable prior in WSD. They use parsed data to find distributionally similar words to a target ambiguous word in SENSEVAL and then use the associated similarity scores to discover the predominant sense for that word. One benefit of both LB04 and McCarthy's method is that the predominant senses can be derived without relying on HTD, which may not be available for every domain and text type. This is important because the frequency of the senses of words depends on the genre and domain of the text under consideration.
- A suitable prior model derived from untagged texts can also help improve the performance of a classifier over a uniform prior. This is exactly what is shown in LB04. However, although the informative priors in LB04 are derived from untagged texts, the likelihood is deduced from HTD. However, if we do have a hand-tagged corpus, then we might as well derive an empirical prior from it. Relying on HTD to estimate the likelihood defeats half of the purpose of deriving an informative prior from untagged data.

The primary goal of this paper is to explore the feasibility of disambiguating Levin verbs using only untagged data. Although supervised systems trained on hand-tagged corpora typically achieve a better WSD performance than their unsupervised alternatives, their applicability is limited. First, creation of hand-labeled corpora is costly and time-consuming. Next, supervised WSD systems can be only applied to those words and senses to which sense-labeled training data are available. Last but not least, the distribution for word senses is often skewed and likely to vary across domains and genres of the text under consideration. A WSD system trained on a hand-tagged corpus of financial news is unlikely to perform well in disambiguating texts of sports reports.

For these reasons, we propose a class-based approach to WSD, one that completely avoids the need for HTD. When the information about a verb is not immediately available, our system will fall back on information about the class of which the given verb is a member. We will illustrate the way this idea works using a Naive Bayes classifier, which allows us to derive both the prior and posterior probabilities from untagged data. For the prior, we adapt our previous work. LB04 proposes an informative prior that is able to choose predominant classes for Levin verbs from unlabeled data. The key assumption of this prior model is that semantic class determines the subcategorization distribution of its members independently of their identity. We have found that the description of LB04's prior model is more complex than it needs to be. Some of the described parameters are actually irrelevant to the ultimate outcome of the decision process. It turns out that a single component, the joint probability of class and frame, determines the predominant class. This leads to a much simpler reformulation of LB04's prior model.

https://doi.org/10.1017/S1351324910000136 Published online by Cambridge University Press

For the likelihood, we introduce a method for collecting training data from only untagged data. The insight is that the contexts of unambiguous members of a verb class can be used to estimate the likely behavior of their ambiguous co-members of the same class. In practice, this amounts to a working hypothesis that verbs in the same Levin class tend to share their neighboring words.

Finally, it is straightforward to combine the estimates of prior and likelihood, both of which have been created without recourse to tagged data, into a full system for disambiguation.

2 Deriving informative priors

In WSD, the heuristic of choosing the most common and predominant sense can be difficult to beat. This is because the distribution of senses of a word is often skewed. Recent research on WSD has found that the first sense heuristic can often deliver results competitive with supervised approaches based on local context (Hoste *et al.* 2002; McCarthy *et al.* 2004). In addition, even the supervised approaches which show superior performance to the first sense heuristic often make use of this heuristic as a component (Hoste, Kool and Daelemans 2001). While the first sense heuristic derived from HTD such as SemCor is clearly beneficial, there is still a strong case for obtaining the predominant sense from untagged data so that a WSD system can be tuned to the genre or domain under consideration. LB04 and Merlo, Joanis and Henderson (2005) (hereafter Merlo05) both extend Levin's verb inventory to a statistical prior model that is able to generate class preferences for ambiguous Levin verbs, without relying on HTD.

2.1 LB04's prior model

2.1.1 Description of LB04's prior model

LB04's prior model views the choice of a class for a polysemous verb in a given frame as a maximization of the joint probability $P(c, f, v)$, where v is an ambiguous verb subcategorizing for the frame f with Levin class c:

$$P(c, f, v) = P(v) \times P(f|v) \times P(c|f, v) \tag{2}$$

The estimation of $P(c|f, v)$ relies on the frequency of $F(c, f, v)$, which could be obtained if a parsed corpus annotated with semantic class information were available. Lacking such a corpus, LB04 assumes that the semantic class determines the subcategorization patterns of its members independently of their identity (3):

$$P(c|f, v) \approx P(c|f) \tag{3}$$

By applying Bayes' rule, $P(c|f)$ is rewritten as

$$P(c|f) = \frac{P(f|c) \times P(c)}{P(f)} \tag{4}$$

Substituting (4) into (2), LB04 expresses $P(c, f, v)$ as

$$P(c, f, v) \approx \frac{P(v) \times P(f|v) \times P(f|c) \times P(c)}{P(f)} \tag{5}$$

https://doi.org/10.1017/S1351324910000136 Published online by Cambridge University Press

Thus, for a verb v in a frame f, LB04 finds its predominant class c according to the following equation:

$$c_i = arg\max_i P(c_i, f, v) \approx arg\max_i \left(\frac{P(v) \times P(f|v) \times P(f|c_i) \times P(c_i)}{P(f)} \right) \quad (6)$$

The model's outcome is considered correct if it agrees with the most frequent verb class as found in the annotated corpus sample. Consider again the verb *call* which in a double object frame is ambiguous between the class DUB and GET. According to LB04's prior model, DUB is the most likely class for *call* when it occurs in a double object frame. Out of 100 instances of the verb *call* in a double object frame, 93 are manually assigned the DUB class, 3 assigned the GET class, and 4 are parsing mistakes. In LB04's evaluation, this outcome is considered correct since human annotation of a small test set also reveals a preference for the DUB class. LB04 reports that the prior model achieves an accuracy of 73.0 per cent , in contrast to a baseline preference accuracy of 46.2 per cent.[1] LB04 suggests that misclassifications are due mainly to the independence assumption that semantic class determines the subcategorization patterns of its members independently of their identity ($P(c|f,v) \approx P(c|f)$). For example, LB04's prior model selects BUILD as the predominant class for the verb *cook* when it occurs in a double object frame. Although it is generally the case that BUILD verbs (e.g. *make*, *assemble*, *build*) are more frequent in double object frames than PREPARE verbs (e.g. *bake*, *roast*, *boil*), the situation is reversed for *cook*.

2.1.2 Reformulation of LB04's prior model

In (6), we can drop $P(v)$, $P(f|v)$ and $P(f)$ since they do not vary with the choice of the class c and therefore can be ignored. Consequently, (6) can be rewritten as:

$$arg\max_i P(c_i, f, v) \approx arg\max_i (P(f|c_i) \times P(c_i)) \quad (7)$$

In other words, it is the value of $P(f|c)$ and $P(c)$ that determines the most likely class for a given verb. In LB04, $P(f|c)$ and $P(c)$ are estimated as

$$P(f|c) = \frac{F(c,f)}{F(c)} \quad (8)$$

$$P(c) = \frac{F(c)}{\sum_i F(c_i)} \quad (9)$$

Combining (8) and (9), the value that determines the predominant class for a given verb in a given frame is calculated as

$$\frac{F(c,f)}{F(c)} \times \frac{F(c)}{\sum_i F(c_i)} = \frac{F(c,f)}{\sum_i F(c_i)} \quad (10)$$

[1] In LB04, the baseline is obtained as follows: for a verb v that belongs to $\{c_1, c_2, \ldots, c_n\}$ in a frame f, choose the class c_i as the predominant class where $F(c_i) > F(c_j)$ for all j such that $1 \leq j \leq n$ and $i \neq j$. See Section 2 of LB04 for discussion on how $F(c)$ is estimated.

https://doi.org/10.1017/S1351324910000136 Published online by Cambridge University Press

Table 1. *Estimation of $F(c, f, v)$ in LB04 (LB04 page 52)*

GIVE	F(GIVE, NP1-V-NP2-NP3, v)	F(GIVE, NP1-V-NP2-PP*to*, v)
feed	$\frac{98}{2}$	$\frac{40}{2}$
give	25,705	7,502
lend	343	648
rent	$\frac{6}{2}$	10
pass	$\frac{181}{3}$	$\frac{256}{3}$
serve	85	$\frac{58}{2}$

The value of the denominator $\sum_i F(c_i)$ is independent of c, so it can be dropped. It is only a normalizing constant to ensure that we have a probability function. So $F(f, c)$ is the only quantity that affects the predominant class. Thus, in our implementation, we find the predominant class c for a verb v in a frame f according to the following equation:

$$c_i = arg \max_{c_i} F(c_i, f) \tag{11}$$

Following LB04, we obtain $F(c, f)$ by summing over all occurrences of verbs that are members of class c and attested in the corpus with frame f:

$$F(c, f) = \sum_i F(c, f, v_i) \tag{12}$$

For each individual verb v, $F(c, f, v)$ is estimated as follows, where $|C|$ is the number of verb classes a verb v belongs to when occurring in a frame f:

$$F(c, f, v) = \frac{F(f, v)}{|C|} \tag{13}$$

For verbs that are members of only one class, $F(c, f, v)$ is just the number of times these verbs have been attested in the corpus with a given frame. LB04's approach to estimating this quantity for polysemous verbs is to evenly divide $F(c, f, v)$ between the classes that allow v to have the frame f. Table 1 shows the estimation from the British National Corpus (BNC) of the frequency $F(c, f, v)$ for six verbs that are members of the GIVE class. Consider, for example, the verb *feed*, which is a member of four Levin classes: GIVE, GORGE, FEEDING, and FIT. Of these four classes, only GIVE and FEEDING license the double object and dative frame. This is why the co-occurrence frequency of *feed* with these two frames is divided by two. In contrast, the verb *give* only belongs to one Levin class: GIVE. Hence, the co-occurrence frequency is equivalent to the number of times the verb *give* has been attested with the double object or dative frame in the corpus.

Our step-by-step reformulation of LB04's prior model has revealed that the only value that matters to the decision on the predominant class of a verb v in a frame f is $F(c, f)$. This conclusion follows from the independence assumption LB04 makes as described in (3). Based on our reformulation of LB04's prior model, the

https://doi.org/10.1017/S1351324910000136 Published online by Cambridge University Press

Table 2. *Frequency of six classes with NP1-V-NP2*

Rank	Class	*F*(Class,NP1-V-NP2)
1	CONT. LOCATION	70,471
2	ADMIRE	66,352
3	HURT	12,730
4	WIPE MANNER	10,294
5	ASSESS	9,872
6	PUSH–PULL	9,828

identity of the verb should play no role in LB04's estimation of the predominant class. In particular, two different verbs that are members of the same classes must be estimated to have the same predominant class. For example, both *miss* and *support* are ambiguous between the class ADMIRE and CONT. LOCATION when occurring in a transitive frame. Since *F*(CONT. LOCATION, NP1-V-NP2) is greater than *F*(ADMIRE, NP1-V-NP2) in the untagged BNC (see Table 2), the model selects CONT. LOCATION as the predominant class for both verbs. However, the prevalence in the manually annotated corpus data from the BNC suggests that CONT. LOCATION is the preferred class for *miss* while ADMIRE is the preferred class for *support*. The independence assumption makes it impossible for the model to select the right predominant class for both *miss* and *support*, since there is nothing in the model that can distinguish between them.

2.2 Merlo05's prior model

Merlo05 explores the consequences of two ideas: first, they want a method where verb identity matters; second, they want to give a more explicit role to alternations. Merlo05 also models the preference of a verb for a given class based on subcategorization frames. The chosen parameters and their estimation differ in detail from LB04, but the task is the same and the approach is similar. Merlo05's prior model tries directly to maximize the conditional probability $P(c|v,f)$.

$$P(c|v,f) = \frac{P(c|v) \times P(f|c,v)}{P(f|v)} \tag{14}$$

Merlo05 makes a different independence assumption. It assumes that all verbs of a given class have a similar distribution of subcategorization frames, which means $P(f|c,v) \approx P(f|c)$. Based on this assumption, $P(c|v,f)$ is approximated as:

$$P(c|v,f) \approx \frac{P(c|v) \times P(f|c)}{P(f|v)} = \frac{P(c|v) \times P(f|c)}{\sum_{c'} P(c',f|v)} = \frac{P(c|v) \times P(f|c)}{\sum_{c'} P(c'|v) \times P(f|c')} \tag{15}$$

Thus, Merlo05 finds the predominant class c for a verb v in a frame f according to the following equation:

$$c_i = arg\max_{c_i} \frac{P(c_i|v) \times P(f|c_i)}{\sum_{c'} P(c'|v) \times P(f|c')} \tag{16}$$

In Merlo05, $P(c|v)$ and $P(f|c)$ are estimated using the verb-frame pairs found in the BNC parsed with a parser as described in Henderson (2003). As in LB04, if the

https://doi.org/10.1017/S1351324910000136 Published online by Cambridge University Press

verb-frame pair class is unambiguous, it is counted as an occurrence of that class. If it is ambiguous, the count is split uniformly across the different classes for the verb (see Table 1). In Merlo05, $P(c|v)$ and $P(f|c)$ are estimated using add-one smoothing, as follows:

$$P(c|v) = \frac{1 + \sum_f F(c,f,v)}{\sum_{c'}(1 + \sum_f F(c',f,v))} \tag{17}$$

$$P(f|c) = \frac{1 + \sum_v F(c,f,v)}{\sum_{f'}(1 + \sum_v F(c,f',v))} \tag{18}$$

Recall that our reformulation of LB04's prior model reveals that the identity of a verb is irrelevant in determining the preferred class for it. In fact, LB04 argues that this independence assumption is responsible for misclassifications made by its prior model. In Merlo05's prior model, the identity of each individual verb does play some role in that the model has to estimate $P(c|v)$. It is therefore worthwhile testing on our data whether the introduction of the verb identity into Merlo05's prior model can overcome the deficiency attributed to the independence assumption in LB04.

2.3 Experiments on prior models

Implementation of LB04 and Merlo05's prior models requires extraction of Levin-defined subcategorization frames from corpora.

LB04 uses a parsed version of the whole BNC made with GSearch (Corley *et al.* 2001), a tool that facilitates the search of arbitrary POS-tagged corpora for shallow syntactic patterns. It uses a chunk grammar for recognizing the verbal complex, NPs and PPs, and applies GSearch to extract tokens matching frames specified in Levin. A set of linguistic heuristics are applied to the parser's output in order to filter out unreliable cues (Lapata 1999).

Merlo05 uses a parsed version of 75 per cent of the BNC made with the parser described in Henderson (2003) and extracts all reduced parsed sentences. The reduced parsed sentences consist of subject, direct object, indirect object, and first prepositional phrase. In extracting frames, it does not make an extensive argument-adjunct distinction, but removes temporal NPs that would have been misidentified as direct or indirect object.

Our implementation uses two sets of frames acquired from the whole BNC using two different statistical parsers:

- We parse the whole BNC with Charniak's parser (Charniak 2000) and extract frames specified in Levin.
- We obtain the frame set from Schulte im Walde (2000). This frame set is acquired from the whole BNC using a head-entity parser described in Carroll and Rooth (1998).

It should be noted that these two frame sets do not make any distinction between argument and adjunct. We implement both LB04's model and Merlo05's model using in each case these two separate sets of frames.

https://doi.org/10.1017/S1351324910000136 Published online by Cambridge University Press

Table 3. *Test data from LB04*

Frame	Number of Verb Types
NP1-V-NP2-NP3	12
NP1-V-NP2-PP*to*	16
NP1-V-NP2	36

Table 4. *Type accuracy for LB04 and Merlo05's prior models (%)*

Implementation	Our implementation		As reported in LB04
Parser	Charniak	Carroll–Rooth	GSearch
LB04	57·8	60·9	74·1
Merlo05	46·8	46·8	NA
Baseline	39·7±0·01		46·2

2.3.1 Test data

We obtain our test data from Mirella Lapata. See LB04 page 57 for a detailed discussion on how their test data is selected. This data set, summarized in Table 3, consists of 5,078 ambiguous verb tokens involving 64 verb types and 3 frame types.[2]

2.3.2 Results of the prior models' performance

We report the results of our implementation of LB04 and Merlo05's prior models on two evaluation metrics: accuracy by verb type and accuracy by verb token.

Verb Type: LB04 measures the performance of its prior model using accuracy by verb type. This accuracy is the percentage of verb types for which the prior model correctly selects the predominant class. Again, the outcome of the prior model is considered correct if the class selected by the prior model agrees with the most frequent class found in the manually annotated corpus sample. Table 4 provides a summary of the results for our implementation of LB04 and Merlo05's prior models. For comparison purposes, we also include the results as reported in LB04. We also compute a baseline by randomly selecting a class out of all the possible classes for a given verb in a particular frame.[3]

Table 4 shows that our implementation of LB04's prior model achieves a better performance (using either set of frame frequency) than the baseline. However, it fails to match the accuracy level reported in LB04. Our implementation of Merlo05's prior model produces type level results more than 10 per cent worse than our implementation of LB04.

[2] The test data we use here is not identical to that used in LB04. It has undergone both additional corrections and systematic adjustments before being released to us.

[3] We replicate this random selection 100 times and average the results to produce the numbers in the table.

https://doi.org/10.1017/S1351324910000136 Published online by Cambridge University Press

Table 5. *Token accuracy for LB04 and Merlo05's prior models (%)*

Implementation Parser	Our implementation		As reported in Merlo05 Henderson
	Charniak	Carroll–Rooth	
LB04	58·7	59·1	46·4
Merlo05	44·4	44·8	45·3
Baseline	37·9 ± 0·005		41·1

Verb Token: Merlo05 measures the performance of their implementation of LB04 and Merlo05's prior model using accuracy by verb token. This accuracy is the percentage of verb tokens for which the prior model is correct in assigning a class. Following Merlo05, we select the class with the highest probability for all occurrences of a verb according to a prior model. This is also known as the first sense heuristic. For example, our implementation of LB04's prior model assigns a higher probability to the DUB class than the GET class for the verb *call* when it occurs in a double object frame. All 96 instances of the verb *call* in double object frames in our test data are therefore assigned the DUB class.

In Merlo05, the test data consists of 3,680 sentences, which corresponds to a subset of the 5,059 sentences provided by Mirella Lapata. Among these, they randomly select 1,840 sentences for test. To compare our results with those reported in Merlo05, we also randomly selected 1,840 sentences from the whole set of 5,078 sentences we obtained from Mirella Lapata. However, we repeat this selection 100 times, which gives us 100 sets of test data. Our final result on verb token accuracy is obtained by averaging the results on these 100 sets of test data. We also provide a random baseline for verb tokens by selecting a class out of all the possible classes for each verb token in a particular frame.[4]

Table 5 shows that according to Merlo05's implementation, neither of the two prior models improves much over their random baseline (only about 4–5 per cent improvement), and LB04's prior model is only slightly better than Merlo05's prior model. Our implementation shows that LB04's prior model is almost 15 per cent better than Merlo05's regardless of the frame sets used. This suggests that the way that identity of individual verbs is introduced into Merlo05's prior model fails to overcome the deficiency of the independence assumption made in LB04.[5] In addition,

[4] Again, we replicate this random selection 100 times and the result reported in Table 5 is obtained by averaging.

[5] Merlo05 does do not give any explanation for this observed inferiority of their model in comparison to LB04's. We suspect that the following two factors may have worked against Merlo05's prior model: (i) In Merlo05, $P(c|v)$ and $P(f|c)$ are estimated using add-one smoothing (see (17) and (18)). It is possible that add-one smoothing is a poor choice for this task since it tends to give too much probability mass to unseen events. (ii) Merlo05's prior model has more parameters to estimate, and in most cases these parameters can only be approximated. To be more specific, to find the predominant class c for a verb v when it occurs in a frame f, LB04's prior model only involves the estimation of the particular frame f under consideration ($F(c, f, v)$). Merlo05's prior model, on the other hand, requires the estimation of all Levin-defined frames in which a verb v occurs when it is a member of a class c, as shown in (18).

https://doi.org/10.1017/S1351324910000136 Published online by Cambridge University Press

there is a discrepancy between the performance of Merlo05's implementation of LB04 and ours. There may be differences in the data,[6] but the difference in performance is too large for this to be the only reason. We are confident that our implementation of LB04's prior model is correct and suspect that the confusing formulation in LB04 may have led Merlo05's implementation of LB04's model to be incorrect, with a loss of performance.

2.4 Impact of parse quality

As far as LB04's prior model is concerned, there is a discrepancy between our results and those reported in LB04. LB04 reports much higher performance. We discuss differences in the data and implementation below, but again suspect that the reason may be algorithmic. Since we implement LB04's prior model correctly, this would mean that their performance may have been obtained using an implementation that, while not matching the description in their paper, nonetheless has an empirical advantage in performance.

One major difference between our implementation of LB04's prior model and the implementation as described in LB04 is the quality of the subcategorization frame information that is used to estimate the prior. LB04 uses a custom built grammar and a shallow syntactic parser (GSearch) to acquire frames from the BNC, whereas we use full-scale statistical parsers (Charniak, Carroll-Rooth). At first blush, it is reasonable to expect full parsers to produce more accurate frames. However, there are reasons that, in retrospect, lead us to believe that the frame set used in LB04 may be more accurate than what is obtained from full parsers. These reasons include:

- Charniak's parser does not differentiate between NP arguments and NP adjuncts. This injects noise into the frame data. For instance, many instances of NP1-V-NP2 frame are mistaken for instances of NP1-V-NP2-NP3 (e.g. *I fed* [*the boy*] [*yesterday*]), thus polluting the data for double object frame. However, LB04 employs a set of linguistic heuristics to filter out these unreliable cues (Lapata 1999).
- LB04 treats compound nouns specially, using a likelihood-ratio based filter (Dunning 1993), operating on the output of GSearch to distinguish between compound nominals and sequences that should be parsed as a pair of noun phrases. Empirically, Charniak's parser, whatever else it may have learned from the Treebank that it is trained on, has not learned to effectively segment and disambiguate nominal compounds in the way we need (e.g. *Some also offer* [*free bus*] [*service*]).

To investigate how the quality of the acquired frames affects the performance of LB04's prior model, we, following Lapata (1999), develop a process which assesses whether the cues derived from the output of Charniak's parser are true instances of

[6] Merlo05's implementation of LB04's prior model is different from ours in terms of the data used: (i) The annotated data are not identical even though we both obtained the data from Mirella Lapata; (ii) Our implementation is based on the frames extracted from the whole BNC, while Merlo et al's implementation makes use of 75 per cent of the BNC.

https://doi.org/10.1017/S1351324910000136 Published online by Cambridge University Press

the frame types under consideration.[7] However, this cleaning process does not have any real impact on the prior model's performance. Thus, the lower performance of our implementation of LB04's prior model is probably not due to the false frame cues generated by Charniak's parser. LB04's prior model is based on the assumption that the semantic class determines the subcategorization patterns of its members independently of their identity. This assumption allows LB04's prior model to approximate $P(c|f,v)$ with $P(f|c)$. LB04 uses GSearch and we adopt Charniak's parser for acquiring Levin-defined frames. We suspect that the frame set generated by GSearch matches this approximation better than the frame set proposed by Charniak's parser, which leads to a more accurate prior model as reported in LB04.

3 Disambiguating Levin verbs using untagged data

In Section 2, we have focused on deriving an informative prior model of the distribution of Levin classes without relying on annotated data (hereafter IPrior). Our implementation of LB04's prior model shows that this model infers the right class for genuinely ambiguous Levin verbs 57.8 per cent of the time without taking the local context of their occurrences into account. This is much lower than the 74.1 per cent accuracy reported in LB04. We want to investigate whether the IPrior we obtain, which is less accurate, can still improve the disambiguation accuracy when combined with contextual features. However, our primary goal is to obtain local contextual information from untagged data, which will enable us to build a verb sense disambiguation system that completely avoids the needs for HTD.

3.1 Classifier

Following LB04, we employ a Naive Bayesian classifier for our disambiguation task. Although the Naive Bayesian classifier is simple, it is quite efficient and has shown good performance on WSD. Another reason for using a Naive Bayes classifier is that it is easy to incorporate the prior information, and we want to test whether an IPrior can help improve the performance of the Naive Bayes classifier. Within a Naive Bayes approach, the choice of the predominant class for an ambiguous verb v when occurring in a frame f given its context can be expressed as

$$C(f,v) = arg\max_{c_i}(P(c_i|f,v)\prod_{k=1}^{n}P(a_k|c_i,f,v)) \tag{19}$$

where $C(f,v)$ represents the predominant class for an ambiguous verb v when occurring in a frame f. $P(c_i|f,v)$ is the prior probability of the ambiguous verb v belonging to class c_i when occurring in frame f, and $\prod_{k=1}^{n}P(a_k|c_i,f,v)$ is the likelihood probability which can be estimated from the training data simply by counting the co-occurrence of feature a_k with class c, verb v, and frame f. For

[7] See Lapata (1999) for details regarding the linguistic heuristics and statistical tests for eliminating false cues of the target frame types.

https://doi.org/10.1017/S1351324910000136 Published online by Cambridge University Press

features that have zero counts, we use add-k smoothing , where k is a number less than one.

3.2 Automatic generation of training data

LB04 shows that an IPrior helps improve the performance of a classifier over a uniform prior (UPrior) when combined with contextual features. Although the IPrior is derived from untagged data, the likelihood is actually induced from a small HTD set, making it still dependent on HTD.

Merlo05 attempts to derive the likelihood probability from untagged data, taking advantage of the diathesis alternation described in Levin. The disambiguation information in not defined by the adjacent text alone, but rather by the relationship between the local contexts in which a target ambiguous Levin verb occurs and other sentences in which the same verb appears. If another sentence is found which could represent an alternation with the local context, then that is taken as evidence for the classes which licenses that alternation. Merlo05 first reduces sentences to a vector of slot-filler pairs, which they call a *reduced parsed sentence*. For instance, Example 3.1 is reduced to Example 3.2, and Examples 4.1 to 4.2. In constructing reduced parsed sentences, Merlo05 only keeps the information regarding subject(SUBJ), object(OBJ), indirect object (IOBJ), and the first prepositional phrase (PP) and the head noun selected by the preposition.

Example 3.1 *John called Tom a taxi.*
Example 3.2 VERB = *call*, SUBJ = *John*, OBJ = *taxi*, IOBJ = *Tom*, PP = *null*

Example 4.1 *John called a taxi for Tom.*
Example 4.2 VERB = *call*, SUBJ = *John*, OBJ = *taxi*, IOBJ = *null*, PP*for* = *Tom*

Recall that the verb *call* is ambiguous between the class DUB and GET when occurring in double object frame. According to Levin, GET verbs licence both double object frame and benefactive frame, whereas DUB verbs only allow for double object frame. This means that *call* is not ambiguous when it occurs in benefactive frame. Hence, the contextual information extracted from benefactive frames in which the verb *call* occurs can be utilized to disambiguate *call* when it appears in double object frame. This enables Merlo05 to disambiguate each individual occurrence of ambiguous Levin verbs without any HTD. However, Merlo05 shows that using diathesis alternation fails to improve the performance of the verb sense disambiguator. We suspect that this may be due to data sparsity. A verb that licenses a particular alternation may prefer to occur in one of the variant frames over the other. Statistics based on the dis-preferred variant frame are likely to be unreliable, because the variant frame is so rare.

In our experiment, we adopt a different method for generating training data from untagged texts. Levin has classified her verbs primarily according to their syntactic behavior. In contrast, many researchers have shown that words with similar contextual features, typically neighboring words, are also semantically similar (Lin 1998; Schutze 1998; Pado and Lapata 2003; Rohde, Gonnerman and Plaut 2004). Faced with these two different approaches to identifying semantically similar verbs,

https://doi.org/10.1017/S1351324910000136 Published online by Cambridge University Press

Table 6. *Verbs in the DUB and GET class*

Class	Ambiguous verbs	Unambiguous verbs
DUB	*call, make, vote*	*anoint, baptize, brand, christen consecrate, crown, decree, dub name, nickname, pronounce, rule stamp, style, term*
GET	*call, find,leave, vote*	*book, buy, cash, catch charter, choose, earn, fetch gain, gather, hire, keep order, phone, pick, pluck procure, pull, reach, rent reserve, save, secure, shoot slaughter, steal, win*

a question of interest is whether verbs in a particular Levin class tend to share their neighboring words. The design principles of Levin's classification imply that verbs in the same class should often be substitutable in the same set of syntactic frames, though not necessarily in exactly the same local contexts. However, Li and Brew (2008) have shown that the simple co-occurrence features work as well as subcategorization frames in inducing Levin-style verb classification from corpora, and a combination of co-occurrence and frame features tend to yield the best results. This seems to suggest that verbs belonging to the same Levin class share, at least to a certain extent, their local contextual features, represented by neighboring words. Based on this observation, we decide to use the contextual information of unambiguous verbs in a Levin class to disambiguate ambiguous ones in that class. This allows us to build a verb sense disambiguator that completely avoids the need for HTD.

To perform verb class disambiguation without relying on HTD, we will train our verb class disambiguator on the examples containing only unambiguous Levin verbs. Consider the verb *call* again. It is ambiguous between the class DUB and GET when it occurs in a double object frame. However, most verbs in these two classes are not ambiguous, as shown in Table 6. For an unambiguous verb, we know for sure the class it belongs to without even examining the sentence in which it occurs. To disambiguate *call* in a double object frame, we collect all the examples that are identified as double object frame and contain an unambiguous DUB verb, and use these examples as the training data for the class DUB. Training data for the class GET is generated in the same way. The likelihood probability can then be induced from these automatically generated training data.

Our approach for automatically generating training data is similar to that of Yarowsky (1992) and Leacock *et al.* (1998) in that a knowledge base is needed, but it differs in other respects, notably in the properties of the knowledge base we choose, and the way training data is identified. Our disambiguation task is targeted at the verb sense inventory in Levin and Yarowsky (1992) and Leacock *et al.* (1998) at the word sense inventory in Roget's thesaurus and WordNet respectively.

https://doi.org/10.1017/S1351324910000136 Published online by Cambridge University Press

Both Roget's thesaurus and WordNet link words through various lexical semantic relations, such as synonym, antonym, and hypernym. The working hypothesis is that words related through these lexical semantic relations are supposed to share their neighboring words. Our method for collecting training data is based on a similar working hypothesis that verbs in the same Levin classes tend to share their neighboring words, although Levin does not make such a claim. It is very easy and straightforward for Yarowsky (1992) and Leacock *et al.* (1998) to identify the training data in the corpus they use. They only need to pick the sentences that contain an unambiguous word, which stands in some lexical semantic relation to the ambiguous word under consideration. Because our verb sense disambiguator is targeted at ambiguous Levin verbs in a particular frame, identification of training data relies on parsers. Even though we have applied linguistic heuristics to the frames acquired by Charniak's parser, the training data constructed this way is likely to be polluted since some false instances of the target frames will be included. Hence the training data we use could have a negative effect on the performance of the verb class disambiguator.

3.3 Feature space

As demonstrated in previous research on WSD (Lee and Ng 2002; McCarthy *et al.* 2004), a feature space that combines both lexical (e.g. lexical co-occurrences or collocation) and syntactic information (e.g. dependency relations, or syntactic categories) tends to yield a better performance on WSD than either of them used alone.

In our experiments, we do not seek a combined feature set, but adopt the simple lexical collocation features. One reason that we do not explore syntactic features in this experiment is that our task is only concerned with Levin verbs that are ambiguous in a particular subcategorization frame. All occurrences of each ambiguous verb inhabit the same subcategorization frame in our training and test data. Another reason is that LB04 has shown that collocation features are very effective for this disambiguation task.

Following LB04, we consider 12 types of lexical collocation. Collocation and co-occurrence features are similar, except that for collocation features, the position in which a neighboring word occurs relative to target verbs matters. Examples of our collocations for the verb *call* are illustrated in Table 7. The L columns in the table indicate the number of words to the left of the ambiguous words, and the R columns the number of words to the right. So for example, the collocation L2R4 represents two words to the left and four words to the right of the target ambiguous word. Collocations are represented as lemmas obtained by lemmatizing context words using the English lemmatizer as described in Minnen, Carroll and Pearce 2000.

3.4 Results and discussion

We use the same test data from our experiments on prior models, which consists of 5,078 ambiguous verb tokens involving 64 verb types and 3 frame types.

https://doi.org/10.1017/S1351324910000136 Published online by Cambridge University Press

Table 7. *Collocation features for disambiguating Levin verbs*

L	R	Example	L	R	Example
0	1	**call** you	1	1	will **call** you
1	0	will **call**	1	2	will **call** you a
0	2	**call** you a	2	1	I will **call** you
2	0	I will **call**	1	3	will **call** you a big
0	3	**call** you a big	3	1	Perhaps I will **call** you
3	0	Perhaps I will **call**	2	4	I will **call** you a big fool

3.4.1 Models

We compare the performance of six different models. Recall that within a Naive Bayes approach, the choice of the predominant class c for an ambiguous verb v when occurring in a frame f given its context can be expressed as

$$C(f,v) = arg\max_{c_i}(P(c_i|f,v)\prod_{k=1}^{n} P(a_k|c_i,f,v)) \tag{20}$$

For example, we could be trying to select c from $\{\mathbf{GET}, \mathbf{DUB}\}$ given the knowledge that $v =$ **call** and $f =$ **ditransitive**, and the information that the surrounding features a_k include the word "cab", among others.

The six models we test differ from each other in whether the prior is derived from HTD and whether the likelihood is induced from the HTD.

- **Prior**
  - EPrior: The empirical priors derived from HTD. In our experiment, the empirical priors are derived from the training examples. EPrior is estimated as $P(c_i|f,v)$.
  - IPrior: The informative priors derived from untagged texts as described in Section 1.2. Since LB04 assumes that semantic class determines the distribution of subcategorization frames independently of individual verbs, the identity of each ambiguous verb does not play a role in determining the predominant class for a verb in a particular frame. As a result, IPrior is estimated as $P(c_i|f)$.
  - UPrior: The uniform priors.
- **Likelihood**
  - HTD: The classifier is trained on HTD. In our experiment, the classifier is trained and tested using 10-fold stratified cross-validation on the 5,078 annotated examples. The likelihood is estimated as $\prod_{k=1}^{n} P(a_k|c_i,f,v)$.
  - NHTD: The classifier is trained without using HTD. In our experiment, the training data consists of all the examples containing only unambiguous verbs. The classifier is tested on all 5,078 annotated examples. Since we utilize the contextual features of unambiguous verbs in a Levin class to which a target ambiguous verb belongs, the contextual features of each

https://doi.org/10.1017/S1351324910000136 Published online by Cambridge University Press

Table 8. *Estimation of Naive Bayes classifiers for six different models*

Model	$C(v,f) =$
EPrior+HTD	$arg\max_{c_i}(P(c_i\|f,v)\prod_{k=1}^{n}P(a_k\|c_i,f,v))$
IPrior+HTD	$arg\max_{c_i}(P(c_i\|f)\prod_{k=1}^{n}P(a_k\|c_i,f,v))$
UPrior+HTD	$arg\max_{c_i}(\prod_{k=1}^{n}P(a_k\|c_i,f,v))$
EPrior+NHTD	$arg\max_{c_i}(P(c_i\|f,v)\prod_{k=1}^{n}P(a_k\|c_i,f))$
IPrior+NHTD	$arg\max_{c_i}(P(c_i\|f)\prod_{k=1}^{n}P(a_k\|c_i,f))$
UPrior+NHTD	$arg\max_{c_i}(\prod_{k=1}^{n}P(a_k\|c_i,f))$

ambiguous verb have no part in training a Naive Bayesian classifier. As a result, the likelihood is estimated as $\prod_{k=1}^{n}P(a_k|c_i,f)$.

We experiment with six models: **EPrior+HTD**, **IPrior+HTD**, **UPrior+HTD**, **EPrior+NHTD**, **IPrior+NHTD**, and **UPrior+NHTD**.[8] The estimation of the six models are summarized in Table 8.

For the purpose of comparison, we also report the performance of three different baseline models:

- **Random Baseline (RB)**: We randomly select a class from all those that are compatible with the given verb and frame. Selection is based on a uniform distribution. The RBs for all three frame types are all slightly lower than 40 per cent.
- **IPrior Baseline (IPB)**: We select the class whose IPrior is the largest of the available possibilities.
- **EPrior Baseline (EPB)**: We select the class whose EPrior is the largest of the available possibilities.

In addition, we also include the upper bound (**UB**) reported in LB04 on disambiguation performance. This is obtained by measuring how well two human judges agree with each other (percentage of agreement) on the class assignment task. The UBs for all three frame types are around 90 per cent.

3.4.2 Results

The results of our class disambiguation experiments for each individual frame are summarized in Figures 1, 2 and 3 respectively. A similar pattern in disambiguation accuracy is observed for three different frames:

- When the classifier is trained on HTD, a higher performance is achieved by using the IPrior (IPrior+HTD) than the UPrior (UPrior+HTD). This agrees

[8] We also estimate a prior from the unambiguous examples only, but its performance is about the same as the IPrior.

https://doi.org/10.1017/S1351324910000136 Published online by Cambridge University Press

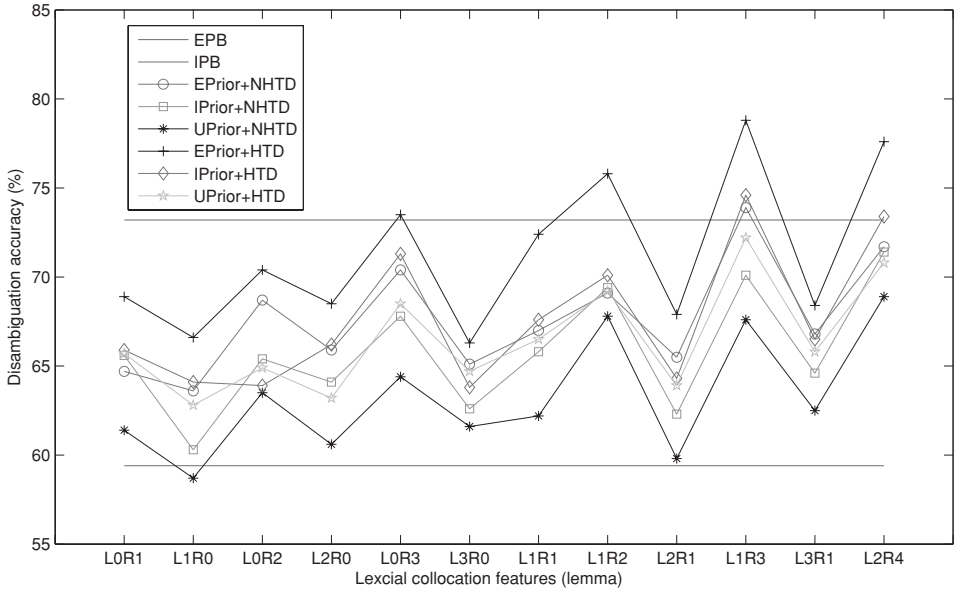


Fig. 1. Verb sense disambiguation accuracy for the frame NP1-V-NP2-NP3.

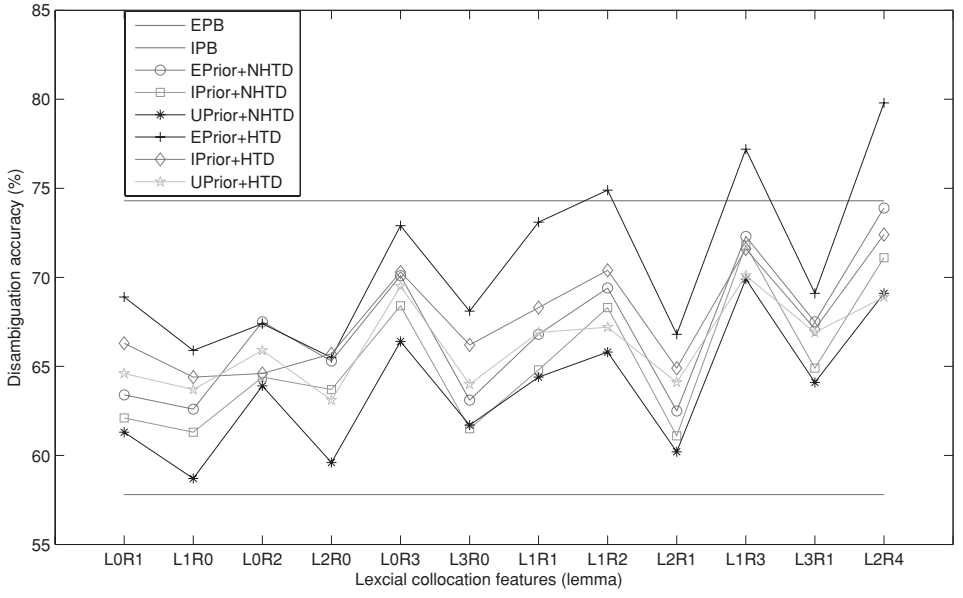


Fig. 2. Verb sense disambiguation accuracy for the frame NP1-V-NP2-PP*to*.

with what is shown in LB04. However, using the IPrior (IPrior+HTD) does not match the performance of using the EPrior (EPrior+HTD). This suggests that it is better to derive a prior from a hand-tagged corpus if such a corpus is available even though it is likely to be small in size.

https://doi.org/10.1017/S1351324910000136 Published online by Cambridge University Press

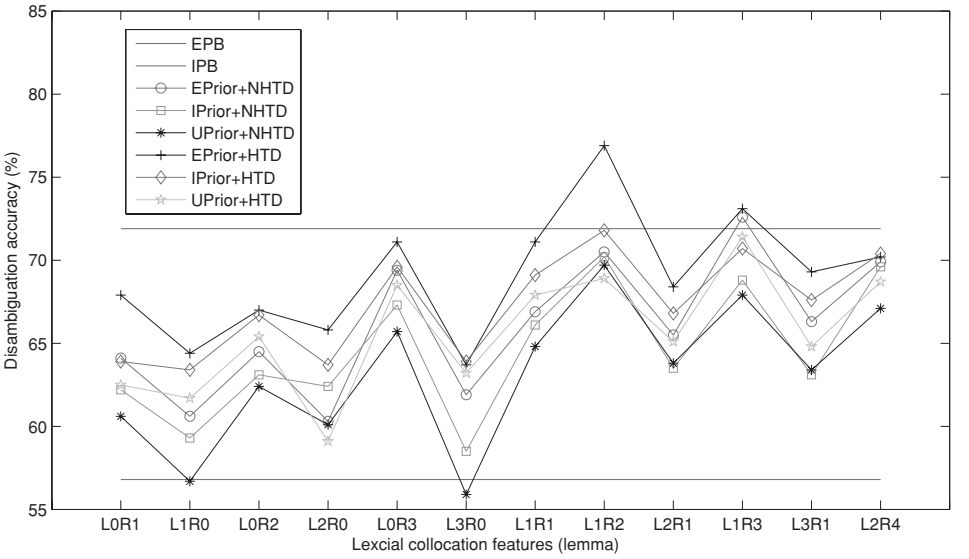


Fig. 3. Verb sense disambiguation accuracy for the frame NP1-V-NP2.

- When the classifier is trained using untagged data (NHTD), neither using the UPrior (UPrior+NHTD) nor using the IPrior (IPrior+NHTD) performs better than their counterparts that are trained on HTD (UPrior+HTD, and IPrior+HTD). However, they both (UPrior+NHTD and IPrior+NHTD) consistently outperform the random baseline (RB), suggesting that our verb class disambiguator using only untagged data can be used in the absence of a hand-tagged corpus. In addition, using the EPrior with the classifier trained on untagged data (EPrior+NHTD) yields a performance comparable to that achieved by the supervised model with the UPrior (UPrior+HTD), indicating that a more accurate prior helps more with the disambiguation task, and is therefore more desirable to obtain. Furthermore, using the IPrior (IPrior+NHTD) outperforms using the UPrior (UPrior+NHTD). This suggests that the IPrior derived from untagged data, though not as accurate as EPrior, can still help improve the performance of the classifier.

The selection of the collocation window size and direction also has a significant effect on the results. However, it is the direction of the collocation window that matters most. For the three frame types we test, the collocation features to the right of target verbs prove to be much more useful than those to the left. As shown in Figures 1, 2 and 3, the highest accuracy is usually achieved using the collocation feature L1R2, L1R3 and L2R4 while using collocation feature L1R0 and L2R0 tends to give much lower disambiguation accuracy. This pattern is common to all the models (supervised and unsuperivsed) we experiment with. To put such results into a more linguistic prospective, collocation features within the right window of ambiguous verbs are likely to cover direct and indirect objects, and collocation features within the left window correspond to subjects. The disambiguation results

https://doi.org/10.1017/S1351324910000136 Published online by Cambridge University Press

we obtain seem to indicate that at least for Levin verbs that are ambiguous in the three frames we test, it is the object argments that makes the sense distinctions. This certainly agrees with the examples given in Examples 2.1 and 2.2. When the verb *call* occurs in a double object frame, it is the direct object (*fool* and *cab*) that determines its correct sense.

Overall, for the task of disambiguating Levin verbs, using a hand-tagged corpus generally outperforms using untagged data. Among the six models we test, EPrior+HTD and IPrior+UHTD deserve some special attention. EPrior+HTD is the best model when a hand-tagged corpus is available, and IPrior+UHTD yields the best performance when only untagged data is allowed. As indicated by the disambiguation results, the best performance achieved for all three frame types by EPrior+HTD is about 5–7 per cent higher than IPrior+UHTD. This result for disambiguating Levin verbs is compatible with the trend prevalent in WSD with SENSEVAL (McCarthy *et al.* 2004). We suspect that the following two factors contribute to the outcome we have observed:

- For our class-based approach which does not need any HTD, we have to rely on a statistical parser for identifying the target frames in constructing the training data. The training data obtained this way is noisy in that some false instances of the target frames are included, although we apply linguistic heuristics to improve the quality of the acquired frames. In contrast, the supervised models depend on hand-tagged training data, which has been examined by human annotators, and should be free of any false instances of the target frames.
- Levin claims that verbs belonging to the same classes are not intended to be substitutable for each other in the exact local contexts. Although our experiments on automatic verb classification suggest that verbs in the same Levin class tend to share their neighboring words (Li and Brew 2008), it should not be expected that the local contextual information, collected from examples containing unambiguous verbs in a class, will agree perfectly with that collected from examples containing target ambiguous verbs.

Nevertheless, our verb sense disambiguator built on untagged data is effective, and can be used in the absence of HTD.

4 Conclusion and future work

In this paper, we investigate the feasibility of disambiguating Levin verbs using only untagged data. Levin's verb classification provides a verb sense inventory where each class corresponds to one unique verb sense. Taking advantage of this design, we adopt a class-based approach to building a verb sense disambiguation system, which completely avoids the need for expensive HTD. This class-based approach is driven by the idea that when the information about a verb is not sufficient or immediately available, it is feasible to fall back on the information about the verb class to which this verb belongs to.

https://doi.org/10.1017/S1351324910000136 Published online by Cambridge University Press

Our first experiment is about inducing an informative prior for Levin verbs from untagged data. LB04 extends Levin's verb classification to a statistical prior model that is able to generate class preferences for a particular ambiguous Levin verb. To build a model from untagged data only, LB04 makes a hypothesis that the semantic class determines the subcategorization patterns of its members independently of their identity ($P(c|f,v) \approx P(c|f)$). We provide a clearer reformulation of LB04's prior model, which helps pinpoint the only value that determines the most predominant class for a given ambiguous Levin verb. This value is the frequency of all the verbs in a Levin class that occurs with a particular frame ($F(c,f)$). This is a direct consequence of the independence assumption made in LB04's prior model.

In our second experiment, we build a Naive Bayesian classifier to disambiguate each individual occurrence of an ambiguous Levin verb in context. We derive the prior information based on our reformulation of LB04's prior model. LB04 learns the likelihood probability from the local contextual features extracted from a small amount of HTD. Unlike LB04, we adopt a method for extracting contextual features from untagged data. Our working hypothesis is that verbs in the same Levin class tend to share their neighboring words, which allows us to use contextual features extracted from examples containing unambiguous verbs from a Levin class to disambiguate the ambiguous ones in that class ($\prod_{k=1}^{n} P(a_k|c_i,f,v) \approx \prod_{k=1}^{n} P(a_k|c_i,f)$).

This class-based approach we take to disambiguating Levin verbs allows us to build verb sense disambiguation systems without any help of HTD. Thus, it can be easily tuned to disambiguate verbs in texts from a specific domain. Our experiments on verb sense disambiguation highlight a beneficial effect of a class-based approach to construction of verb lexicon. A class-based verb lexicon, like Levin, provides a verb sense inventory where each class corresponds to one unique verb sense. Falling back on the information about the class to which an ambiguous verb belongs makes it possible to build a verb sense disambiguator without using any HTD.

Several avenues can be taken for future research. First, although this class-based approach to disambiguating Levin's verb yields promising results, it still remains to be shown that it generalizes across frames and alternations. Levin lists 79 alternations involving almost 200 verb classes. In our experiment, we only investigate 3 frame types. It is a matter of future work to apply this class-based approach to more verbs, classes and frames. Next, we choose simple lexical collocation features in our disambiguation task. Although these features prove to be very effective in WSD, we plan to explore the output of parsers and use argument fillers as context features. In particular, we want to examine how the choice of different context features affects the performance of our class-based approach to disambiguating Levin's verbs.

Acknowledgments

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

https://doi.org/10.1017/S1351324910000136 Published online by Cambridge University Press

References

Carroll, G., and Rooth, M. 1998. Valence induction with a head-lexicalized PCFG. In *Proceedings of the 1998 Conference on EMNLP*, pp. 58–63.

Charniak, E. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 2000 Conference of NAACL*, pp. 132–139.

Corley, S., Corley, M., Keller, F., Cocker, M., and Trewin, S. 2001. Finding syntactic structure in unparsed corpora. *Computers and the Humanities* **35**(2): 81–94.

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*, pp. 293–299.

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

Dunning, T. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics* **19**(1): 61–74.

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

Goldberg, A. 1995. *Constructions*. Chicago: University of Chicago Press.

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

Henderson, J. 2003. Inducing history representations for broad-coverage statistical parsing. In *Proceedings of the 2003 Joint Meeting of NAACL/HLT*, pp. 103–110.

Hoste, V., Kool, A., and Daelemans, W. 2001. Classifier optimization and combination in the English all words tasks. In *Proceedings of the SENSEVAL-2 Workshop*, pp. 84–86.

Hoste, V., Hendrickx, I., Daelemans, W., and van den Bosch, A. 2002. Parameter optimization for machine learning of word sense disambiguation. *Natural Language Engineering* **8**(4): 311–325.

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., and Palmer, M. 2000. Class-Based Construction of a Verb Lexicon. In *Proceedings of AAAI/IAAI*, pp. 691–696.

Korhonen, A. 2002. *Subcategorization Acquisition*. Cambridge University Press.

Korhonen, A., Krymolowski, Y., and Marx, Z. 2003. Clustering polysemic subcategorization frame distributions semantically. In *Proceedings of the 41st Annual Meeting of ACL*, pp. 64–71.

Korhonen, A., and Preiss, J. 2003. Improving subcategorization acquisition using word sense disambiguation. In *Proceedings of the 41st Annual Meeting of ACL*, pp. 48–55.

Lapata, M. 1999. Acquiring lexical generalizations from corpora: a case study for diathesis alternations. In *Proceedings of the 37th Annual Meeting of ACL*, pp. 397–404.

Lapata, M., and Brew, C. 2004. Verb class disambiguation using informative priors. *Computational Linguistics* **30**(1): 295–313.

Leacock, C., Chodorow, C., and Miller, C. 1998. Using corpus statistics and WordNet relations for sense identification. *Computational Linguistics* **24**(1): 147–165.

Lee, K., and Ng, H. 2002. An empirical evaluation of knowledge sources and learning algorithm for word sense disambiguation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pp. 41–48.

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

Li, J., and Brew, C. 2008. Which are the best features for automatic verb classification. In *Proceedings of the 46th Annual Meeting of ACL*, pp. 434–442.

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*, pp. 768–774.

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*, pp. 280–287.

Merlo, P., Joanis, E., and Henderson, J. 2005. Unsupervised verb class disambiguation based on diathesis alternations, Manuscript.

https://doi.org/10.1017/S1351324910000136 Published online by Cambridge University Press

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

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

Pado, S., and Lapata, M. 2003. Constructing semantic space from parsed corpora. In *Proceedings of the 41st Annual Meeting of ACL*, pp. 545–552.

Palmer, M. 2000. Consistent criteria for sense distinction. *Computers and the Humanities* **1**(2): 217–222.

Rohde, D., Gonnerman, L., and Plaut, D. 2004. An improved method for deriving word meaning from lexical co-occurrence. *Cognitive Psychology* **7**: 573–605.

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

Schutze, H. 1998. Automatic word sense disambiguation. *Computational Linguistics* **24**(1): 97–124.

Shen, D., and Lapata, M. 2007. Using semantic roles to improve question answering. In *Proceedings of the 2007 Conference on EMNLP-CoNLL*, pp. 12–21.

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*, pp. 100–111.

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

Yarowsky, D. 1992. Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proceedings of the 15th International Conference on COLING*, pp. 88–94.

https://doi.org/10.1017/S1351324910000136 Published online by Cambridge University Press