Analysis of supervised word sense disambiguation systems

1. Introduction

Recent trends in word sense disambiguation (Ide and Veronis, 1998) show that the most effective paradigm for word sense disambiguation (WSD) is that of supervised learning. Supervised algorithms rely on tagged examples to construct a language model, which is used to disambiguate new examples. Nevertheless, current literature has not shown that supervised methods can scale up to disambiguate all words in a text into reference (perhaps fine-grained) word senses. Possible causes of this failure are:

1. Most tagging exercises use idiosyncratic word senses (e.g. ad-hoc built senses, translations, thesaurus, homographs, ...) instead of widely recognized semantic lexical resources (ontologies like Sensus, Cyc, EDR, WordNet, EuroWordNet, etc., or machine-readable dictionaries like OALDC, Webster's, LDOCE, etc.) which usually have fine-grained sense differences.
2. Unavailability of training data: existing hand-tagged corpora seems not to be enough for current state-of-the-art systems.
3. The Machine Learning (ML) algorithms applied are too limited to confront the problem.
4. The feature sets used to model the language do not extract all the knowledge required to perform WSD. Features obtained with complex analysis of the text (morphological, syntactic, semantic...) and the combination of different types of features are not taken into account.
5. Problem is wrongly defined: tagging with word senses is hopeless (Senseval, 2001). We will not tackle this issue here.

In this paper, we try to shed some light on these points. First, as word sense inventory, we chose to work with WordNet (Miller et al., 1990). This allows us to compare our results with other works, and to use available lexical resources.

Regarding the unavailability of hand-tagged data, we test how far can we go with existing hand-tagged corpora like Semcor (Miller et al., 1993) and the DSO corpus (Ng and Lee, 1996), which have been tagged with word senses from WordNet. Besides we test an algorithm that automatically acquires training examples from the Internet, based on (Mihalcea and Moldovan, 1999).

In order to analyze supervised WSD systems, we can concentrate either on the ML algorithm applied, or in the feature set used to build the language model. In this work, we chose to fix the ML algorithm and focus on new features. We think, as in (Pedersen, 2000), that the feature set variations contribute more to disambiguation performance than variations in Machine Learning algorithms. Our goal is to test different feature sets in WSD tasks (separately and in combination) and analyze their behavior. As ML method, we use one of the most successful learning algorithms to date (Yarowsky, 1994), which is based on decision lists. Different implementations of this method have won Senseval-1 and Senseval-2 (Senseval, 2001). This algorithm allows us to combine different types of information easily (lexical, morphological, syntactic, semantic...).

Finally, for the analysis of different features, first we study the basic lexical feature sets used in the literature (word bigrams and trigrams, bags of words...). We evaluate them on both Semcor and DSO corpora, and try to test how far could we go with such big corpora. The next step is to get more rich features using a syntactic parser and analyze their performance on the same corpora.

Additionally, after completing basic experiments with these features, we analyze the system under different conditions. There are few in-depth analysis of algorithms, and precision figures are usually the only features available. We think that if new ways out of the acquisition bottleneck are to be explored, previous questions about supervised algorithms should be answered: how much data is needed, how much noise can they accept, can they be ported from one corpus to another, can they deal with really fine sense distinctions, the compromise between precision and coverage, what performance can we expect… etc. We will design some series of experiments in order to shed light on these questions.

The paper is organized as follows. First, the previous work on the field, in Section 2. Section 3 is devoted to explain the decision list algorithm. The experimental setting applied is described in Section 4. The basic set of features and the syntactic features are defined in Section 5 and 6, respectively. In section 7, we show the experiments with the basic set and in section 8 the experiments using syntactic features are illustrated. Section 9 is dedicated to experiments regarding different issues about supervised systems. Finally, in section 10 we analyze a method to acquire training data from the Internet, in section 11 we present some conclusions of the work.

2. Previous work

In (Yarowsky, 1994), a basic set of features was defined which has been used widely (with some variations) by WSD systems. It consisted in words appearing in a window of ±k positions around the target and bigrams and trigrams constructed with the target word. He used words, lemmas, coarse part-of-speech tags and special classes of words, such as "Weekday". These features have been used by other approaches, with variations such as the size of the window, the distinction between open class/closed class words or the pre-selection of significative words to look up in the context of the target word.

In (Ng, 1996), local collocations and surrounding words are selected as features only if they are indicative of some sense; which is previously measured using conditional probability. Their basic set of features is similar to those defined by Yarowsky, but they also use syntactic information: verb-object and subject-verb relations. The results obtained by the syntactic features are poor, and no analysis of the features or any reason for the low performance is given.

(Leacock et al., 1998) also rely on basic features. They define a small window for the local context and a wider one for the topical context. Words and part-of-speech tags are used in the features. They differentiate between open class words (nouns, verbs, adjectives and adverbs) and closed class words and analyze the effect of each type of feature.

(Stetina et al., 1998) achieve good results with syntactic relations as features. They use a measure of semantic distance based on WordNet to find similar features. The features are extracted using Collins Parser (Collins, 1996), and consist on the head and modifiers of each phrase. They use the whole parse tree to disambiguate all the words in the context.

In a recent work, (Pedersen 2000) defines his feature set using bigrams of words that appear in a window of ±50 words around the target. He uses different statistics to select significant bigrams, which will be used to construct a decision tree. He obtains good results using only this source of information. An algorithm based on this idea was applied in (Senseval 2001). The Senseval 2001 workshop was held in Toulouse in July. Although the descriptions of all the systems are not yet available, most of the systems have provided a brief description in the web page of Senseval. There we can see that most techniques extract only basic features to train their ML algorithms. However, there are also many others that use semantic relations between words to construct their models (mostly based on WordNet). There are few methods that apply syntactic relations. The team from the University of Sussex extracts selectional preferences based on subject-verb and verb-object relations and employs them to disambiguate senses. They use the WordNet hierarchy to obtain the selectional preferences. Their implementation obtains low recall in the all-words task. It is not clear yet which algorithms or feature sets have worked best. A deep analysis of the systems should be performed in order to get some conclusions.

3. Decision lists

Decision lists as defined in (Yarowsky, 1994) are simple means to resolve ambiguity problems. He has applied it successfully to accent restoration (Yarowsky, 1994), homograph disambiguation (Yarowsky, 1996) and word sense disambiguation (Yarowsky, 1995). It was one of the most successful systems on the Senseval-1 word sense disambiguation competition (Kilgarriff and Evans, 2000) and also in (Senseval, 2001).

The training data is processed to extract the features, which are weighted with a log-likelihood measure. The list of all features ordered by the log-likelihood values constitutes the decision list. We adapted the original formula in order to accommodate ambiguities higher than two. In our case, the weight of $sense_i$ when $feature_k$ occurs in the context is computed as the logarithm of the probability of $sense_i$ given $feature_k$ divided by the summatory of the probabilities of the other senses given $feature_k$.

$$weight(sense_i, feature_k) = Log\left(\frac{\Pr(sense_i \mid feature_k)}{\sum_{j \neq i} \Pr(sense_j \mid feature_k)}\right)$$

It is not clear what to do when all weights of the senses for the given feature are below 0. We decided to delete such features from the decision lists.

When testing, the decision list is checked in order and the feature with highest weight that is present in the test sentence selects the winning word sense. The probabilities have been estimated using the maximum likelihood estimate, smoothed using a simple method: when the denominator in the formula is 0 we replace it with 0.1**.** The estimates can be improved using more sophisticated smoothing techniques (Chen, 1996).

4. Experimental setting

The experiments were targeted at three different corpora. Semcor (Miller et al., 1993) is a subset of the Brown corpus with a number of texts comprising about 200.000 words in which all content words have been manually tagged with senses from WordNet (Miller et al. 1990). It has been produced by the same team that created WordNet. The DSO corpus

(Ng and Lee, 1996) was differently designed. 191 polysemous words (nouns and verbs) and 1000 sentences per word were selected from the Wall Street Journal and Brown corpus. In the 191.000 sentences only the target word was hand-tagged with WordNet senses. Both corpora are publicly available[1]. Finally, following a technique described in Section 10, an Internet corpora was automatically acquired for 7 words comprising around 100 examples per word sense.

For the experiments, we decided to focus on 19 content words, selected using the following criteria:

- Frequency, number of training examples in Semcor (low, high)
- Ambiguity, number of senses (low, high)
- Skew of most frequent sense in Semcor (low, high)

The two first criteria are interrelated (frequent words tend to be highly ambiguous), but there are exceptions. The third criterion seems to be independent, but high skew is sometimes related to low ambiguity. We could not find all 8 combinations for all parts of speech and the following samples were selected (cf. Table 4 in Section 7): 2 adjectives, 2 adverbs, 8 nouns and 7 verbs. These 19 words form the test set A.

The DSO corpus does not contain adjectives or adverbs, and focuses in high frequency words. Only 5 nouns and 3 verbs from Set A were present in the DSO corpus, forming Set B of test words (cf. Table 4 in Section 7).

In addition, 4 files from Semcor previously used in the literature (Agirre and Rigau, 1996) were selected, and all the content words in the files were disambiguated (cf. 7.2).

Semcor has been cited as having scarce data to train supervised learning algorithms (Miller et al., 1994). *Church* for instance occurs 128 times, but *duty* only 25 times and *account* 27. In order to use all available data and have a fair evaluation of such limited amount of data, we performed 10-fold cross validation in all experiments, including the most frequent baseline.

The measures we use are precision, recall and coverage, all ranging from 0 to 1. Given N, number of test instances, A, number of instances which have been tagged, and C, number of instances which have been correctly tagged:

- precision = C / A
- recall = C / N
- coverage = A / N

Because of ties, we used a modified measure of precision, equivalent to choosing at random in ties. Instead of counting 1 when any of the winning senses is correct, we count only a fraction. That is, we substitute C with C' in the above formulae, where C' is computed as follows:

$$C' = \sum_{i \in test\_instances} c(i) \quad where\, c(i) = \begin{cases} \frac{1}{number\ of\ winning\ senses} & if\ i\ correctly\ done \\ 0 & otherwise \end{cases}$$

[1] Consult www.cogsci.princeton.edu and www.ldc.org for conditions.

5. Basic Features

We analyzed several features already mentioned in the literature (Yarowsky, 1994; Ng, 1997; Leacock et al., 1998). Different sets of features have been created to test the influence of each feature type in the results. Topical and local information has been used.

Topical features correspond to features that appear in windows of different sizes around the target word (words and lemmas). We used five different window-sizes: 4, 20 and 50 words around the target, all the words in the sentence, and all the words in the sentence plus the previous and next sentences. Local features include bigrams and trigrams that contain the target word, formed by part of speech, lemmas and word forms. We also used bigrams and trigrams with more general parts of speech: nouns, verbs, adjectives, adverbs and others.

This is the total set of topical features:

- win_lem_Nw: the lemmas appearing in a (-N,+N) word window around the target word.
- win_wf_Nw: the word forms appearing in a (-N,+N) word window around the target word.
- win_lem_0s: the lemmas in the sentence of the target word.
- win_wf_0s: the word forms in the sentence of the target word.
- win_lem_1s: the lemmas appearing in a (-1,+1) sentence window around the target word.

This is the total set of local features:

- big_wf_+1: the bigram formed by a word followed by the target word.
- big_wf_-1: the bigram formed by a word following the target word.
- big_lem_+1: the bigram formed by a lemma followed by the target word.
- big_lem_-1: the bigram formed by a lemma followed by the target word.
- big_subpos_+1: the bigram formed by a part of speech tag followed by the target word.
- big_subpos_-1: the bigram formed by a part of speech tag followed by the target word.
- big_pos_+1: the bigram formed by a "coarse" part of speech tag followed by the target word.
- big_pos_-1: the bigram formed by a "coarse" part of speech tag followed by the target word.
- trig_wf_+1: the trigram formed by the target word and the two previous words.
- trig_wf_0: the trigram formed by the target word and the previous and next word.
- trig_wf_-1: the trigram formed by the target word and the two following words.
- trig_lem_+1: the trigram formed by the target word and the two previous lemmas.
- trig_lem_0: the trigram formed by the target word and the previous and next lemmas.
- trig_lem_-1: the trigram formed by the target word and the two following lemmas.
- trig_subpos_+1: the trigram formed by the target word and the two previous part of speech tags.
- trig_subpos_0: the trigram formed by the target word and the previous and next part of speech tags.
- trig_subpos_-1: the trigram formed by the target word and the two following part of speech tags.
- trig_pos_+1: the trigram formed by the target word and the two previous "coarse" part of speech tags.
- trig_pos_0: the trigram formed by the target word and the previous and next "coarse" part of speech tags.
- trig_pos_-1: the trigram formed by the target word and the two following "coarse" part of speech tags.

In the following example from DSO, we train the decision lists with the basic set to disambiguate the verb "know" in a sentence. In Table 1, we show the features, the associated sense, and the log-likelihood value. We see that only features related to the first and fourth senses of “know” receive positive values, from the 8 senses the word has in WordNet 1.6 . Sense 4 is “correctly chosen”[2] because of the word “widely” appearing as bigram, and in the ±3 word window.

e.g.: “There is nothing in the whole range of human experience more widely **known** and universally felt than spirit .”

know has 8 senses, sense 1 and 4 are defined as follows:

Sense 1: know, cognize -- (be cognizant or aware of a fact or a specific piece of information; "I know that the President lied to the people"; "I want to know who is winning the game!"; "I know it's time")

Sense 4: know -- (be familiar or acquainted with a person or an object; "She doesn't know this composer"; "Do you know my sister?" "We know this movie")

Feature	Arguments	Sense	Log-likelihood
Win_wf_3w	Widely	4	2.99
Big_wf_+1	Known widely	4	2.99
Big_wf_-1	Known and	4	1.09
Win_wf_0s	Whole	1	0.91
Win_wf_0s	Widely	4	0.69
Win_wf_0s	Known	4	0.43
Trig_subpos_+1	RB RB	4	0.15

Table 1: Decision list for **know** in the example “There is nothing in the whole range of human experience more widely **known** and universally felt than spirit”.

6. Syntactic Features

In order to extract useful syntactic features from the tagged examples, we needed a parser that would met the following requirements:

A- is free for research
B- provides syntactic relations directly (in contrast with partial parsers that only provide constituent structures, or parse trees without relation markings)
C- has been positively evaluated on well-established corpora
D- is fast enough for big corpora

From the parsers we started looking at, only 2 fulfilled all the requirements at that moment: “Link Grammar”(Sleator and Temperley, 1993) and “Dekang Lin’s Minipar” (Lin, 1993). The different parsers are described in Appendix A. We performed a set of experiments using the mentioned two parsers in order to determine which was more adequate for our task. Even if we obtained similar precision with both, we chose “Minipar” for three

[2] This means that the system makes the same choice as the human taggers. We will not discuss the arbitrariness of the choice.

reasons: it was faster, it showed better coverage extracting similar relations, and the output was easier to process for us.
From the output of the parser, we extracted different sets of features. First, we distinguish between direct relations (those that can be extracted directly from the output of the parser) and indirect relations (those based on words that are two or more dependencies apart in the syntax tree). Indirect relations are constituted by the heads of the complements of the target. For example, from the following sentence: *"**Henry L. Bowden** was **listed** on the **petition** as the mayor 's attorney ."* the direct relation verb-object is extracted between *listed* and; *Henry L. Bowden* and the indirect relation "head of a modifier prepositional phrase" between *listed* and *petition*. For each relation we store also its inverse. The relations are coded in trigrams, as follows:

[Henry L. Bowden	**obj_word**	listed]
[listed	**objI_word**	Henry L. Bowden]
[petition	**mod_Prep_pcomp-n_N_word**	listed]
[listed	**mod_Prep_pcomp-n_NI_word**	petition]

There, for instance, the code of the indirect relation [listed *mod_Prep_pcomp-n_NI_word* petition] is constructed using these components: *mod_Prep* indicates that the word "listed" has some prepositional phrase attached, *pcomp-n_N* indicates that "petition" is the head of the prepositional phrase, *I* indicates that it is an inverse relation, and *word* that the relation is between words (as opposed to relations between lemmas or synsets).
The most relevant relations are shown in Table 2. For each relation this information is provided: the acronym of the relation, whether it is used as a direct relation or to construct indirect relations, a short description, some examples and additional comments. The complete list of relations is given in Appendix B.
Table illustrates the way the different dependencies are related. The arguments are given in bold. We see that some dependencies are defined by 2 or 3 relations in Minipar. For each relation, we show the part-of-speech tags of the components and some examples. The part-of-speech tags give information about the subcategorization of the words, and we will use that to build some features. The different tags are illustrated in Appendix C.

We classified the syntactic features in two groups:

A) Those related to the value of the dependency relations (relations for short): we collect [wordsense relation value] trigrams. As values for the relations, we will use words, lemmas and synsets. Synsets are available only for the content words in Semcor. There are some examples for the target noun "church". In the first case, the features are linked to the "building" sense of church; and in the second case to the "group of Christians" sense.

Example 1 (direct relation): "...Anglican **churches** have been **demolished**..."

[Church#2	obj_lem	demolish]
[Church#2	obj_synset	01137612]
[Church#2	obj_word	demolished]

Example 2 (indirect relation): "...to whip men into a **surrender** to a particular **churh**..."

[Church#1	mod_Prep_pcomp-n_N_lem	surrender]
[Church#1	mod_Prep_pcomp-n_N_synset	05414577]
[Church#1	mod_Prep_pcomp-n_N_word	surrender]

The first example links directly the verb with its object. In the second example, we see that the verb "surrender" is linked to "church" (which is the head of a prepositional phrase) via an indirect relation.

B) Features related to the relations themselves: in this case, we collect bigrams [wordsense relation] and also n-grams [wordsense relation1 relation2 relation3 ...]. The n-grams are linked to the subcategorization of nouns, adjectives and verbs. We have applied n-grams only with verbs. As previously mentioned, Minipar includes simple subcategorization information in the grammatical categories (V_N_N, N_C, A_C ...). The arguments listed can appear or not. We have defined 3 types of n-grams:
   a. Using the subcategorization information as it is. E.g.: V_N_N (the verb can have to nominal adjuncts).
   b. Filtering the arguments in the categories with the relations that actually occurred in the sentence (so adjuncts are surely discarded, but we can miss some arguments that were not in Minipar's lexical specification).
   c. Taking all dependencies in the parse tree (we do not miss arguments, but adjuncts will also be included)

There is an example of the n-grams linked to the verb "fall":

Example: "His mother was nudging him, but he was still **falling**"

[Fall#1	ngram1	V_N_N]
[Fall#1	ngram2	subj]
[Fall#1	ngram3	be+amodstill+subj]

The first feature indicates that the verb has two arguments (i.e. it is transitive). We can see that this is an error of Minipar. The second feature indicates simply that it has a subject and the third feature denotes also the presence of an auxiliar verb and the adverbial modifier "still".

Summing up, in the following experiments we will use seven syntactic feature sets. We will analyze their performance separately and in combination with other features previously defined. The different sets will be coded as follows:

A-direct: trigrams of direct relations with its value (lemas, words and synsets).
A-indirect: trigrams of indirect relations with its value (lemas, words and synsets).
B-direct: bigrams indicating the presence of direct relations.
B-indirect: bigrams indicating the presence of indirect relations.
B-ngram1, B-ngram2 and B-ngram3: n-grams with subcategorization information.

Relation	Direct	Indirect	Description	Example	Comments
by-subj	X		Subj. with passives		
C		X	clausal complement	... that <-c- John loves Mary (?) I go there for + infinitive clause go <-mod- (inf) <-c- for <-i- mainverb	
Cn		X	nominalized clause	to issue is great be <-s inf <-cn inf <-i issue	OFTEN WRONG
comp1	X		complement (PP, inf/fin clause) of noun	... one of the boys one (N_P) <-comp1- of <-pcomp-n- boy .. grants to finance hospitals grants (N_C) <- c1- (inf) <-i- finance ... resolution which voted ... resolution (N_C) <-c1- (fin) <-i- voted	“boy in the garage” is MOD
Desc	X		description	... make a man a child make <-desc- child .... become eclectic	Occurs frequently
Fc	X		finite complement(?)	... said there is ... say <-fc- (fin) <-i- mainverb	
I		X	see c and fc, dep. between clause and main verb		
Mod	X		Modifier	strikes increase as workers demand increase <-mod as <-comp1 fin <-i dema raises to cope with situation raise <-mod inf <-i cope <-mod with <-pcomp-n situation lost <-mod- already satisfactory -mod-> condition	
Obj	X		Object		
pcomp-c	X		clause of pp	in voting itself in <-pcomp-c vpsc <-i- votig	
Pcomp-n	X		nominal head of pp	in the house in <-pcomp-n house	
Pnmod	X		postnominal mod.	person <-pnmod missing	
Pred	X		predicative (can be A or N)	John is beatuful (fin) <-i- is <-pred beautiful <-subj John	
Sc	X		sentential complement	force John to do force <-sc-do	
Subj	X				
Vrel	X		passive verb modifier of nouns	fund <-vrel- granted	When “pnmod”, is tagged as adj. (often wrongly), here is tagged as verb

Table 2: The most relevant syntactic relations.

Source PoS	Dependency	PoS	dep2	PoS	dep3	PoS	Examples
V_N	**Obj**	N	-	-	-	-	
V_N	**Subj**	N CN	- NO	-	-	-	It does not link like s
A (?)	**Subj**	C	i	V			*it is possible to <-subj- be*
V_N	**S**	N CN	- cn	- C	- i	- V	831 *to buy is funny*
V_N	**By-subj**	Pr	pcomp-n pcomp-c	N C	- i	- V	580 *made by J.* 37 *made by cutting*
N *(no subcat)*	Mod	P A	pcomp-...				*end of doing, position of* *accepted practice*
A *(no subcat)*	Mod	P A	pcomp-n/-c				*essential for, fastidious in* *heavily traveled*
VBE	Mod	C P N A	i...				*is to enter* - - *is absolutely*
V *(no subcat)*	Mod	C P A	i... pcom...				*combine to investigate* *join after completing* *was aproved earlier*
C *(no subcat)*	Mod	P C A N	pcomp...				*On other matters, sb. does ...*
N_A/_C/_P A_C/_P	**comp1**	A P C	pcomp-c/-n i	... V			(only N) *sth. close* *one of the day* *time to be*
V_N/V_A	**Desc**	A N					
N	Pnmod	A					*persons missing*
N	Vrel	V					*bonds issued by*
VBE	**Pred**	A N C P	i ... pco...				*there is a plan* *birs are to end* *is across ...*
V_C	**Fc**	C	i	V			subcat C: *have to face*
V_I	**Sc**	V					subcat I: *force sb to take*
V *no subcat*	Amod	A					*even know*

Table 3: Dependencies and their relations.

7. Experiments on Basic Features

In our first experiment, we defined an initial set of features and compared the results with two baselines: the random baseline and the more informed "most frequent sense" baseline (MFS).

We selected a basic combination of features: word-form bigrams and trigrams, and part of speech bigrams and trigrams, a bag with the word-forms in a window spanning 4 words left and right, and a bag with the word forms in the sentence.

The results for the Semcor and DSO corpus, are shown in Table 4, individually per word and averaged across parts of speech, alongside with the baselines. We want to point out the following:

- The number of examples per word sense is very low for Semcor (around 11 for the words in Set B), while DSO has substantially more training data (around 66 in set B). It has to be noted, that several word senses do not occur neither in Semcor nor in DSO.

- The random baseline attains 0.17 precision for Set A, and 0.10 precision for Set B.
- The MFS baseline is higher for the DSO corpus (0.59 for Set B) than for the Semcor corpus (0.50 for Set B). This rather high discrepancy can be due to tagging disagreement, as will be commented on section 9.6.
- Overall, decision lists significantly outperform the two baselines in both corpora: for set B 0.60 vs. 0.50 in Semcor, and 0.70 vs. 0.59 on DSO, and for Set A 0.70 vs. 0.61 on Semcor. For a few words the decision lists trained on Semcor are not able to beat MFS, but in DSO decision lists overcome in all words. **The scarce data in Semcor seems enough to get some basic results. The larger amount of data in DSO warrants a better performance, but limited to 0.70 precision.** Next subsection elaborates the results according to the kind of words.
- The coverage in Semcor does not reach 1.00, because some decisions are rejected when the log likelihood is below 0. On the contrary, the richer data in DSO enables 1.0 coverage.

Regarding the execution time, Table 5 shows training and testing times for each word in Semcor. Training the 19 words in set A takes around 2 hours and 30 minutes, and is linear to the number of training examples, around 2.85 seconds per example. Most of the training time is spent processing the text files and extracting all the features, which includes complex window processing. Once the features have been extracted, training time is negligible as is the test time (around 2 seconds for all instances of a word). Training time has been measured on CPU total time on a Sun Sparc 10 machine with 512 Megabytes of memory at 360 Mhz.

				Semcor				DSO			
Word	POS	Senses	Random	Examples	Ex. Per sense	MFS	Decision Lists	Examples	Ex. Per senses	MFS	Decision Lists
All	A	2	0.50	211	105.50	**0.99**	**0.99**/1.00				
Long	A	10	0.10	193	19.30	0.53	**0.63**/0.99				
Most	B	3	0.33	238	79.33	0.74	**0.78**/1.00				
Only	B	7	0.14	499	71.29	0.51	**0.69**/1.00				
Account	N	10	0.10	27	2.70	0.44	**0.57**/0.85				
Age	N	5	0.20	104	20.80	0.72	**0.76**/1.00	491	98.20	0.62	**0.73**/1.00
Church	N	3	0.33	128	42.67	0.41	**0.69**/1.00	370	123.33	0.62	**0.71**/1.00
Duty	N	3	0.33	25	8.33	0.32	**0.61**/0.92				
Head	N	30	0.03	179	5.97	0.78	**0.88**/1.00	866	28.87	0.40	**0.79**/1.00
Interest	N	7	0.14	140	20.00	0.41	**0.62**/0.97	1479	211.29	0.46	**0.62**/1.00
Member	N	5	0.20	74	14.80	**0.91**	**0.91**/1.00	1430	286.00	0.74	**0.79**/1.00
People	N	4	0.25	282	70.50	**0.90**	**0.90**/1.00				
Die	V	11	0.09	74	6.73	**0.97**	**0.97**/0.99				
Fall	V	32	0.03	52	1.63	0.13	**0.34**/0.71	1408	44.00	0.75	**0.80**/1.00
Give	V	45	0.02	372	8.27	0.22	**0.34**/0.78	1262	28.04	0.75	**0.77**/1.00
Include	V	4	0.25	144	36.00	**0.72**	0.70/0.99				
Know	V	11	0.09	514	46.73	0.59	**0.61**/1.00	1441	131.00	0.36	**0.46**/0.98
Seek	V	5	0.20	46	9.20	0.48	**0.62**/0.89				
Understand	V	5	0.20	84	16.80	**0.77**	**0.77**/1.00				
Set A Avg. A		5.82	0.31	202.00	34.71	0.77	**0.82**/1.00				
Avg. B		5.71	0.20	368.50	64.54	0.58	**0.72**/1.00				
Avg. N		9.49	0.19	119.88	12.63	0.69	**0.80**/0.99				
Avg. V		20.29	0.10	183.71	9.05	0.51	**0.58**/0.92				
Overall		12.33	0.17	178.21	14.45	0.61	**0.70**/0.97				
Set B Avg. N		10.00	0.16	125.00	12.50	0.63	**0.77**/0.99	927.20	92.72	0.56	**0.72**/1.00
Avg. V		29.33	0.06	312.67	10.66	0.42	**0.49**/0.90	1370.33	46.72	0.61	**0.67**/0.99
Overall		17.25	0.10	195.38	11.33	0.50	**0.60**/0.94	1093.38	63.38	0.59	**0.70**/1.00

Table 4: Data for each word and results for baselines and basic set of features.

Word	POS	Senses	Examples	Ex. Per sense	Testing time (secs)	Training time (secs)
All	A	2	211	105.50	2.00	711.20
Long	A	10	193	19.30	2.00	745.20
Most	B	3	238	79.33	2.40	851.80
Only	B	7	499	71.29	5.20	1143.50
Account	N	10	27	2.70	0.00	131.60
Age	N	5	104	20.80	1.00	302.90
Church	N	3	128	42.67	1.00	175.60
Duty	N	3	25	8.33	0.00	133.30
Head	N	30	179	5.97	1.20	500.40
Interest	N	7	140	20.00	1.30	397.20
Member	N	5	74	14.80	1.00	303.70
People	N	4	282	70.50	2.80	686.60
Die	V	11	74	6.73	0.20	276.50
Fall	V	32	52	1.63	0.20	303.10
Give	V	45	372	8.27	4.60	968.30
Include	V	4	144	36.00	1.30	526.70
Know	V	11	514	46.73	4.40	924.30
Seek	V	5	46	9.20	0.00	230.80
Understand	V	5	84	16.80	0.90	344.70
Set A Avg. A		5.82	202.00	34.71	2.00	728.20
Avg. B		5.71	368.50	64.54	3.80	997.65
Avg. N		9.49	119.88	12.63	1.04	328.91
Avg. V		20.29	183.71	9.05	1.66	510.63

Table 5: Execution time for the words in Semcor.

7.1. Local vs. Topical: local best precision, combined best coverage

We also analyzed the performance of topical features versus local features. We consider as local bigrams and trigrams (PoS tags and word-forms), as topical all the word-forms in the sentence plus a special 4 word-form window around the target. The results are shown in Table 6.

The part of speech of the target influences the results: in Semcor we can observe that while the topical context performed well for the nouns, the accuracy dropped for the other categories. These results are consistent with those obtained by (Gale et al. 1993) and (Leacock et al. 1998), which show that topical context works better for nouns. However, the results in the DSO are in clear contradiction with those from Semcor: local features seem to perform better for all parts of speech. It is hard to explain the reasons for this contradiction, but it can be related to the amount of data available in DSO.

The combination of both kinds of features attains lower precision in average than the local features alone, but this is compensated by a higher coverage, and overall the recall is very similar in both corpora (0.66 vs. 0.68 in Semcor, 0.70 vs. 0.70 in DSO).

The t-test column shows whether the precision difference was found significant or not. One of the t-test columns stands for the local/topical difference, and the other for the winner/combination difference. A star means it was found significant. In Semcor, the local method was found to significantly better than the topical method overall, but not for the combination. In DSO all performance differences were found to be significant.

		Semcor					DSO				
PoS		Local Context	Topical Context	t-test	Combination	t-test	Local	Topical	t-test	Combination	t-test
All	A	0.99/1.00	0.98/0.91		0.99/1.00						
Long	A	0.67/0.98	0.61/0.87		0.63/0.99						
most	B	0.79/1.00	0.71/0.95		0.78/1.00						
only	B	0.72/1.00	0.60/0.96		0.69/1.00						
account	N	0.55/0.78	0.47/0.56		0.57/0.85						
age	N	0.73/0.99	0.78/0.87		0.76/1.00		0.76/0.98	0.70/0.97		0.73/1.00	
church	N	0.60/0.98	0.74/0.89		0.69/1.00		0.68/1.00	0.72/0.96		0.71/1.00	
Duty	N	0.62/0.84	0.75/0.48		0.61/0.92						
Head	N	0.89/1.00	0.90/0.85		0.88/1.00		0.78/0.99	0.76/0.97		0.79/1.00	
Interest	N	0.55/0.86	0.57/0.86		0.62/0.97		0.68/0.91	0.60/0.98		0.62/1.00	
Member	N	0.90/0.99	0.91/0.89		0.91/1.00		0.81/1.00	0.78/1.00		0.79/1.00	
People	N	0.90/1.00	0.89/0.94		0.90/1.00						
Die	V	0.97/0.99	0.96/0.70		0.97/0.99						
Fall	V	0.35/0.60	0.35/0.25		0.34/0.71		0.81/0.99	0.80/0.96		0.80/1.00	
Give	V	0.41/0.54	0.32/0.52		0.34/0.78		0.77/1.00	0.78/0.98		0.77/1.00	
Include	V	0.69/0.98	0.73/0.85		0.70/0.99						
Know	V	0.59/0.99	0.57/0.90		0.61/1.00		0.52/0.89	0.37/0.81		0.46/0.98	
Seek	V	0.70/0.80	0.40/0.43		0.62/0.89						
Understand	V	0.77/1.00	0.75/0.81		0.77/1.00						
A		0.84/0.99	0.81/0.89		0.82/1.00						
B		0.74/1.00	0.64/0.96	*	0.72/1.00						
N		0.78/0.96	0.81/0.87		0.80/0.99		0.75/0.97	0.71/0.98	*	0.72/1.00	*
V		0.61/0.84	0.57/0.72		0.58/0.92		0.70/0.96	0.66/0.91	*	0.67/0.99	*
Overall		0.72/0.93	0.68/0.84	*	0.70/0.97		0.73/0.96	0.69/0.95	*	0.70/1.00	*

Table 6: Local context Vs Topical context.

7.2. Overall in Semcor: 0.68 precision for any text

In order to evaluate the expected performance of decision lists trained on Semcor, we selected four files previously used in the literature (Agirre and Rigau, 1996) and all the content words in the files were disambiguated.

For each file, the decision lists were trained using the rest of Semcor. Table 7 shows the results. Surprisingly, decision lists attain a very similar performance in all four files (random and most frequent baselines also show the same behavior). As Semcor is a balanced corpus, it seems reasonable to say that 68% precision can be expected if any running text is disambiguated using decision lists trained on Semcor.

The fact that the results are similar for texts from different sources (journalistic, humor, science) and similar results can be expected for words with varying degrees of ambiguity and frequency (as we will see in Section 9.1), seems to confirm that the training data in Semcor allows to **expect similar precision across all kinds of words and texts**, except for highly skewed words, where we can expect better performance than average.

File	PoS	Avg. Senses	Examples	Random	MFS	Dlist (prec./cov.)
br-a01	Overall	6.60	792	0.26	0.63	0.68/0.95
br-b20	Overall	6.86	756	0.24	0.64	0.66/0.95
br-j09	Overall	6.04	723	0.24	0.64	0.69/0.95
br-r05	Overall	7.26	839	0.24	0.63	0.68/0.92
average	A	5.49	122.00	0.28	0.71	0.71/0.92
	B	3.76	48.50	0.34	0.72	0.80/0.97
	N	4.87	366.75	0.28	0.66	0.69/0.94
	V	10.73	240.25	0.16	0.54	0.61/0.95
	Overall	6.71	777.50	0.25	0.63	0.68/0.94

Table 7: Overall results in Semcor.

7.3. Overall in DSO: state-of-the-art results

In order to compare decision lists with other state of the art algorithms we tagged all 191 words in the DSO corpus. The results in (Ng et al., 1996) only tag two subsets of all the data, but (Escudero et al., 2000a) implement both Ng's example-based approach and a Naive-Bayes system and test it on all 191 words. The same test set is also used in (Escudero et al., 2000b), which presents a boosting approach to word sense disambiguation. The features they use are similar to ours, but not exactly the same. The precision obtained, summarized on Table 8 shows that decision lists provide state-of-the-art performance. Decision list attained 0.99 coverage.

PoS	MFS	Example Based	Naive-Bayes	Boosting	Decision Lists
N	0.59/1.00	0.69	0.68	0.71	**0.72**/0.99
V	0.53/1.00	0.65	0.65	0.67	**0.68**/0.98
Overall	0.56/1.00	0.67	0.67	**0.70**	**0.70**/0.99

Table 8: Overall results in DSO.

8. Experiments on Syntactic Features

We carried out different experiments using the groups of syntactic features defined previously to disambiguate the set of 19 words (set A) in Semcor and the set of 8 words (set B) in DSO. The results obtained in Semcor for the MFS baseline, the basic set of features,

and the syntactic sets are shown in Table 9. The B-ngram features were applied only to verbs, in an attempt to learn subcategorization information useful for disambiguation.
The syntactic sets exhibited different performances. **B-direct** was the only feature set able to obtain acceptable coverage overall (85%), but its precision was lower than the basic feature set and the MFS baseline. **B-ngram1** and **B-ngram2** obtained good coverage for verbs, but they also got lower precision than the baselines. We have to notice that the MFS baseline for verbs is as good as the decision lists with the basic set of features, which makes it difficult to defeat.
The **A-direct** feature set was better in overall precision than the MFS baseline, but with a coverage of 53%. The **indirect** feature sets obtained high precision for some parts of speech, but could only be applied in a few cases.

	A		B		N		V		Overall	
	Prec.	Cov.	Prec.	Cov.	Prec.	Cov.	Prec.	Cov.	Prec.	Cov.
MFS	0.77	1.000	0.58	1.000	0.69	1.000	0.51	1.000	0.61	1.000
Base Features	0.825	1.000	0.699	1.000	0.793	1.000	**0.512**	1.000	**0.670**	1.000
A-direct	0.865	0.312	0.711	0.202	0.782	0.690	0.491	0.692	0.640	0.539
B-direct	0.793	0.892	0.565	0.708	0.705	0.928	0.438	0.873	0.587	0.855
A-indirect	**1.000**	0.010	**1.000**	0.007	**0.909**	0.080	0.479	0.193	0.592	0.099
B-indirect	0.819	0.054	0.817	0.081	0.623	0.453	0.445	0.512	0.537	0.347
B-ngram1							0.453	0.997		
B-ngram2							0.419	0.927		
B-ngram3							0.472	0.664		

Table 9: Basic and Syntactic feature sets in Semcor.

In our next experiment, we used the syntactic features in combination with the basic set of features. The results obtained in Semcor are shown in Table 10. We see that there is no difference in performance adding the syntactic features to the basic set. At this point, we analyzed the behavior of the different features separately, in order to know the reasons for that.
In Table 11 and Table 12, we show the features with overall precision higher than 90% and 70% in Semcor, disambiguating nouns and verbs, respectively. The features are sorted by precision. We see that all the high-precision features are syntactic, but that they always attain very low coverage (below 8%). In Table 13 and Table 14, the features are sorted according to the recall. Only the features that have more than 20% recall are shown, and we can notice that there are only two syntactic features (in bold) for nouns, which correspond to the prepositional complement of the nouns. Most of syntactic features start appearing in the10%-20% recall range. For verbs, the **B-ngram** sets obtain good recall, even better than basic bigrams and trigrams in some cases. This indicates that some subcategorization information has been acquired. Other syntactic features that appear in this table are those related to the subject of the target verb, but attain low coverage. The tables for the whole set of features are given in Appendix D. There we can notice that many syntactic features do not appear in the corpus.

	A		B		N		V		Overall	
	Prec.	Cov.	Prec.	Cov.	Prec.	Cov.	Prec.	Cov.	Prec.	Cov.
MFS	0.77	1.000	0.58	1.000	0.69	1.000	0.51	1.000	0.61	1.000
Base Features	0.825	1.000	0.699	1.000	0.793	1.000	0.512	1.000	**0.670**	1.000
Base + A-direct	0.828	1.000	0.703	1.000	0.794	1.000	0.511	1.000	**0.671**	1.000
Base + B-direct	0.827	1.000	0.699	1.000	0.791	1.000	0.512	1.000	**0.669**	1.000
Base + A-indirect	0.825	1.000	0.699	1.000	0.792	1.000	0.514	1.000	**0.670**	1.000
Base + B-indirect	0.825	1.000	0.699	1.000	0.791	1.000	0.511	1.000	**0.669**	1.000
Base + B-ngram1							0.513	1.000		
Base + B-ngram2							0.512	1.000		
Base + B-ngram3							0.513	1.000		

Table 10: Basic and Syntactic feature sets combined in Semcor.

Feature	Type	Prec.	Cov.	Recall	Feature	Type	Prec.	Cov.	Recall
mod_Prep_pcomp-n_N_word	A-indirect	1,000	0,033	0,033	possI_lem	A-direct	1,000	0,002	0,002
mod_Prep_pcomp-n_N_synset	A-indirect	1,000	0,031	0,031	possI_word	A-direct	1,000	0,002	0,002
mod_lem	A-direct	1,000	0,010	0,010	vrelI_lem	A-direct	1,000	0,002	0,002
mod_synset	A-direct	1,000	0,010	0,010	vrelI_synset	A-direct	1,000	0,002	0,002
mod_word	A-direct	1,000	0,010	0,010	vrelI_word	A-direct	1,000	0,002	0,002
postI_lem	A-direct	1,000	0,007	0,007	has_relat_appo	B-direct	1,000	0,002	0,002
postI_word	A-direct	1,000	0,007	0,007	has_relat_gen	B-direct	1,000	0,002	0,002
has_relat_mod_C_i_VI	B-indirect	1,000	0,007	0,007	has_relat_mod_asI	B-direct	1,000	0,002	0,002
sI_lem	A-direct	1,000	0,006	0,006	has_relat_mod_outI	B-direct	1,000	0,002	0,002
sI_synset	A-direct	1,000	0,006	0,006	has_relat_possI	B-direct	1,000	0,002	0,002
sI_word	A-direct	1,000	0,006	0,006	comp1_C_i_V_lem	A-indirect	1,000	0,002	0,002
subjI_lem	A-direct	1,000	0,006	0,006	comp1_C_i_V_synset	A-indirect	1,000	0,002	0,002
subjI_synset	A-direct	1,000	0,006	0,006	comp1_C_i_V_word	A-indirect	1,000	0,002	0,002
subjI_word	A-direct	1,000	0,006	0,006	comp1_Prep_pcomp-n_NI_lem	A-indirect	1,000	0,002	0,002
has_relat_mod_perI	B-direct	1,000	0,006	0,006	comp1_Prep_pcomp-n_NI_synset	A-indirect	1,000	0,002	0,002
postI_synset	A-direct	1,000	0,005	0,005	comp1_Prep_pcomp-n_NI_word	A-indirect	1,000	0,002	0,002
has_relat_guestI	B-direct	1,000	0,005	0,005	has_relat_s_CN_cn_C_i_VI	B-indirect	1,000	0,002	0,002
has_relat_mod_fromI	B-direct	1,000	0,005	0,005	mod_Prep_pcomp-n_NI_synset	A-indirect	0,963	0,028	0,027
genI_synset	A-direct	1,000	0,004	0,004	obj_word	A-direct	0,959	0,051	0,049
objI_lem	A-direct	1,000	0,004	0,004	mod_Prep_pcomp-n_N_lem	A-indirect	0,947	0,040	0,038
objI_word	A-direct	1,000	0,004	0,004	modI_synset	A-direct	0,941	0,071	0,067
has_relat_vrelI	B-direct	1,000	0,004	0,004	obj_lem	A-direct	0,933	0,061	0,057
has_relat_mod_forI	B-direct	1,000	0,003	0,003	modI_lem	A-direct	0,926	0,085	0,079
conjI_lem	A-direct	1,000	0,002	0,002	modI_word	A-direct	0,926	0,085	0,079
conjI_synset	A-direct	1,000	0,002	0,002	mod_Prep_pcomp-n_NI_word	A-indirect	0,914	0,024	0,022
guestI_lem	A-direct	1,000	0,002	0,002	mod_Prep_pcomp-n_NI_lem	A-indirect	0,900	0,031	0,028
guestI_word	A-direct	1,000	0,002	0,002					
nn_lem	A-direct	1,000	0,002	0,002					
nn_synset	A-direct	1,000	0,002	0,002					
nn_word	A-direct	1,000	0,002	0,002					

Table 11: Performance of features when disambiguating nouns in Semcor, sorted by precision (only features with precision higher than 90%).

Feature	Type	Prec.	Cov.	Recall
has_relat_descl	B-direct	1,000	0,003	0,003
conj_synset	A-direct	1,000	0,002	0,002
conjl_lem	A-direct	1,000	0,002	0,002
conjl_synset	A-direct	1,000	0,002	0,002
guestl_synset	A-direct	1,000	0,002	0,002
Has_relat_mod_atl	B-direct	1,000	0,002	0,002
Has_relat_mod_lnl	B-direct	1,000	0,002	0,002
mod_C_i_V_synset	A-indirect	0,846	0,010	0,008
sc_lem	A-direct	0,833	0,005	0,004
sc_word	A-direct	0,833	0,005	0,004
sc_synset	A-direct	0,800	0,004	0,003
mod_C_i_V_lem	A-indirect	0,752	0,016	0,012
modl_synset	A-direct	0,734	0,012	0,009

Table 12: Performance of features when disambiguating verbs in Semcor, sorted by precision (only features with precision higher than 70%).

Feature	Type	Prec.	Cov.	Recall
win_lem_50w	Basic	0,788	1,000	0,788
win_lem_0s	Basic	0,783	1,000	0,783
win_lem_1s	Basic	0,781	1,000	0,781
win_lem_20w	Basic	0,778	1,000	0,778
win_lem_4w	Basic	0,774	0,998	0,772
win_wf_0s	Basic	0,766	0,974	0,746
big_subpos_+1	Basic	0,708	0,970	0,687
trig_pos_0	Basic	0,663	0,973	0,645
trig_pos_+1	Basic	0,666	0,967	0,644
big_pos_+1	Basic	0,642	0,996	0,639
big_subpos_-1	Basic	0,630	0,975	0,614
trig_subpos_0	Basic	0,735	0,816	0,600
trig_subpos_+1	Basic	0,713	0,829	0,591
win_wf_4w	Basic	0,821	0,712	0,585
big_pos_-1	Basic	0,584	0,993	0,580
big_lem_+1	Basic	0,846	0,652	0,552
trig_pos_-1	Basic	0,648	0,828	0,537
big_wf_+1	Basic	0,849	0,623	0,529
big_lem_-1	Basic	0,740	0,704	0,521
win_wf_3w	Basic	0,833	0,594	0,495
trig_subpos_-1	Basic	0,732	0,656	0,480
big_wf_-1	Basic	0,740	0,633	0,468
pcomp-n_lem	**A-direct**	**0,751**	**0,300**	**0,225**
pcomp-n_word	**A-direct**	**0,751**	**0,300**	**0,225**
trig_lem_+1	Basic	0,878	0,248	0,218
trig_lem_0	Basic	0,824	0,256	0,211
trig_wf_0	Basic	0,831	0,253	0,210
trig_wf_+1	Basic	0,881	0,237	0,209

Table 13: Performance of features when disambiguating nouns in Semcor, sorted by recall (only features with recall higher than 20%).

Feature	Type	Prec.	Cov.	Recall
win_lem_20w	Basic	0,499	1,000	0,499
win_lem_50w	Basic	0,499	1,000	0,499
win_syn_50w	Basic	0,495	1,000	0,495
win_lem_0s	Basic	0,486	1,000	0,486
win_lem_4w	Basic	0,485	1,000	0,485
win_lem_1s	Basic	0,477	1,000	0,477
win_wf_0s	Basic	0,481	0,948	0,456
B-ngram1	**B-ngram**	**0,453**	**0,997**	**0,452**
big_subpos_-1	Basic	0,442	0,985	0,435
big_subpos_+1	Basic	0,443	0,980	0,434
big_pos_+1	Basic	0,433	0,996	0,431
big_pos_-1	Basic	0,426	0,995	0,424
trig_pos_0	Basic	0,431	0,980	0,422
trig_pos_+1	Basic	0,418	0,976	0,408
trig_pos_-1	Basic	0,422	0,922	0,389
B-ngram2	**B-ngram**	**0,419**	**0,927**	**0,388**
big_lem_-1	Basic	0,502	0,759	0,381
win_wf_4w	Basic	0,485	0,753	0,365
trig_subpos_0	Basic	0,447	0,808	0,361
trig_subpos_-1	Basic	0,427	0,808	0,345
big_lem_+1	Basic	0,464	0,741	0,344
trig_subpos_+1	Basic	0,412	0,828	0,341
B-ngram3	**B-ngram**	**0,472**	**0,664**	**0,313**
big_wf_-1	Basic	0,488	0,628	0,306
win_wf_3w	Basic	0,489	0,617	0,302
big_wf_+1	Basic	0,450	0,648	0,292
has_relat_subjl	**B-direct**	**0,448**	**0,620**	**0,278**
has_relat_sl	**B-direct**	**0,443**	**0,603**	**0,267**
subjl_lem	**A-direct**	**0,507**	**0,408**	**0,207**

Table 14: Performance of features when disambiguating verbs in Semcor, sorted by recall (only features with recall higher than 20%).

The low coverage of the syntactic features seems responsible for the lack of improvement of the combined feature sets. We focused on some words and analyzed the acquired decision lists. We observed that in most of the cases the syntactic features were below the basic features, although some of them were strong (e.g.: to have the “vrel” relation as strong indicator of one of the senses of “know”). But in the case of words with dominant senses, some syntactic features could introduce a lot of noise. Relations as “has_related_mod_to”, would point strongly to the most dominant sense. This happens also with non-syntactic features, but in a less harming scale because they comprise a more reduced and controlled set. This suggests that instead of using all the relations provided by Minipar, some selection should be made in order to discard the noisiest ones (this could be done using held-out data). Another conclusion of the analysis was that the parser fails to detect many dependencies and commits some errors, and this affects the coverage.

For our next experiments, we used the DSO corpus. We expected that this would help to improve the coverage of the syntactic features. The results are illustrated in Table 15. For these experiments, we grouped all the A-type features, all the B-type features and the whole set of syntactic features. We can see that the coverage is still poor, but the precision is higher than in Semcor. The MFS baseline is easily beaten, and the A-type features even improve the results of the basic set. The A-type features exhibit a better behavior with nouns, and the B-type features with verbs. But when we combined all the features, as shown in Table 16, there was no improvement over the basic set. The reason for this seems again the poor coverage of the syntactic features. Even with more examples, the features do not get enough information from the analysis of the parser.

	N	V	Overall
	Prec./Cov.	Prec./Cov.	Prec./Cov.
MFS	0.56/1.000	0.61/1.000	0.59/1.000
Base Features	0.731/1.000	0.691/0.994	0.712/0.997
A-direct + A-indirect	**0.762**/0.243	0.714/0.282	**0.737**/0.261
B-direct + B-indirect	0.684/0.350	**0.717**/0.268	0.698/0.310
B-ngram1		0.463/0.406	
B-ngram2		0.529/0.376	
B-ngram3		0.545/0.282	
All syntactic	0.713/0.359	0.693/0.348	0.703/0.354

Table 15: Basic and Syntactic feature sets in DSO.

	N	V	Overall
	Prec./Cov.	Prec./Cov.	Prec./Cov.
MFS	0.56/1.000	0.61/1.000	0.59/1.000
Base Features	0.731/1.000	0.691/0.994	0.712/0.997
Base + A-direct + A-indirect	0.732/1.000	0.695/0.994	0.714/0.997
Base + B-direct + B-indirect	0.732/1.000	0.692/0.994	0.713/0.997
Base + B-ngram1		0.679/1.000	
Base + B-ngram2		0.679/1.000	
Base + B-ngram3		0.680/1.000	
Base + All syntactic	0.733/1.000	0.696/0.995	0.715/0.998

Table 16: Basic and Syntactic feature sets combined in DSO.

From the analysis of the syntactic features, we can conclude that different ways have to be explored in order to take advantage of this source of information. We have mentioned that the selection of the best features could improve the system. Furthermore, we should take into account that some features seem to work better for some kinds of words. Another way to improve the system would be to introduce semantic knowledge, in order to model the selectional preferences of the different senses. This approach will be discussed in the conclusion chapter. Finally, we think that training the system with a more reliable corpus of syntactic relations should improve the performance.

9. Analysis of performance under different conditions

9.1. Results according to the kind of words: skew of MFS

We plotted the precision of decision lists as well as the difference between decision lists and MFS (DL-MFS) and decision lists and random baseline (DL-rand) according to several parameters, and observed the following:

- frequency: Figure 1 shows that no clear idea can be made whether better precision is attained for frequent or infrequent words.
- ambiguity: the data of Figure 2 does not indicate whether ambiguous words are easier or not.
- skew: this is the parameter affecting most the performance of decision lists. Words with high skew obtain better results, but the decision lists outperform MFS mostly on words with low skew (cf. Figure 3).

It needs to be noted the interrelation between ambiguity and frequency. Low ambiguity words may seem easier to disambiguate, but they tend to occur less, and therefore Semcor provides less data. On the contrary, highly ambiguous words occur more frequently, and therefore have more training data.

Overall decision lists perform very well (related to MFS) even with words with very few examples (“duty”, 25 or “account”, 27) or highly ambiguous words.

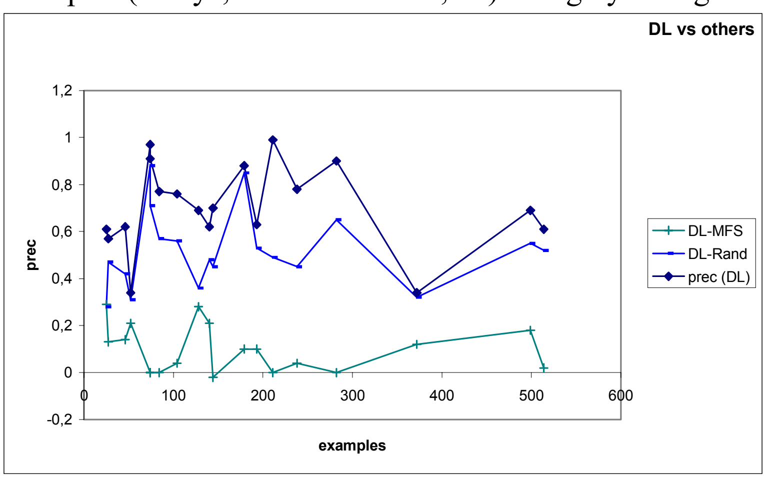

Figure 1: Results according to frequency.

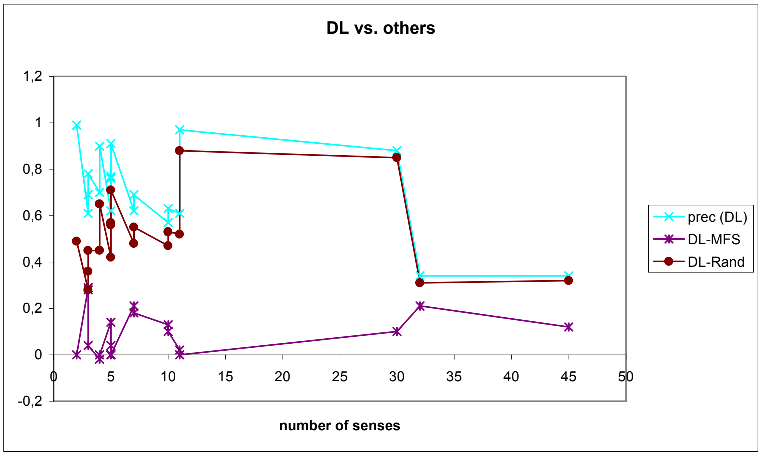

Figure 2: Results according to ambiguity.

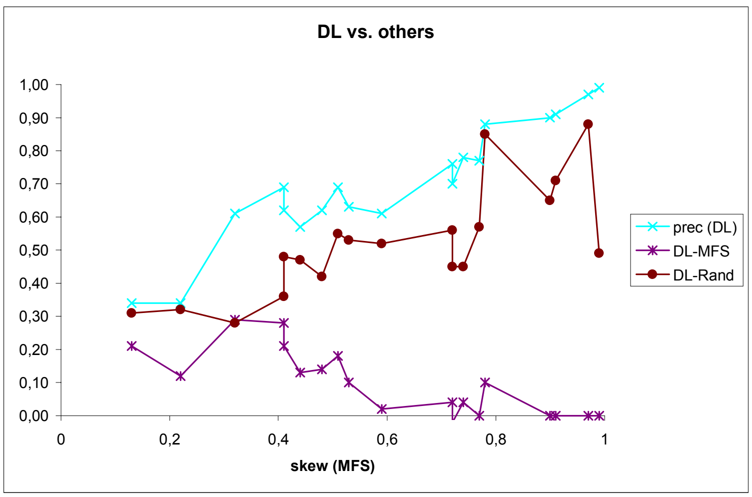

Figure 3: Results according to skew.

9.2. Learning curve: examples in DSO enough

We tested the performance of decision lists with different amounts of training data. We retained increasing amounts of the examples available for each word: %10 of all examples in the corpus, 20%, 40%, 60%, 80% and 100%. We performed 10 rounds for each percentage of training data, choosing different slices of data for training and testing. The number of training examples, precision/coverage and recall[3] given for each percentage of training data is shown in Appendix E (Table 23, Table 24, Table 25 and Table 26). The same data is plotted in Figure 4 and Figure 5, with the number of examples available as a reference.

The improvement for nouns in Semcor seems to stabilize, but the higher amount of examples in DSO shows that the performance can still grow up to a standstill. The verbs show a steady increase in Semcor, confirmed by the DSO data, which seems to stop at 80% of the data.

[3] recall was chosen, in order to compensate for differences in both precision and coverage. That is, recall reflects both decreases in coverage and precision at the same time.

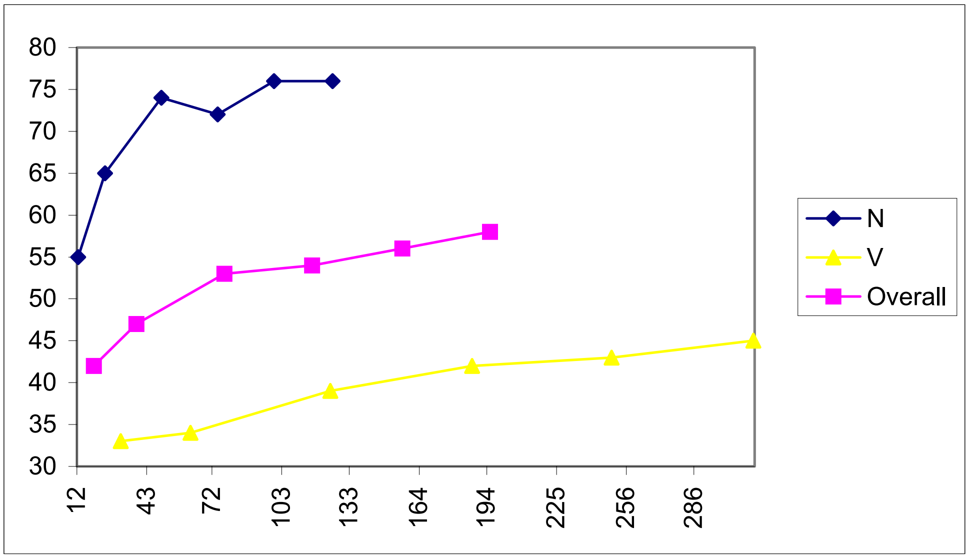

Figure 4: Learning curve in the Semcor corpus.

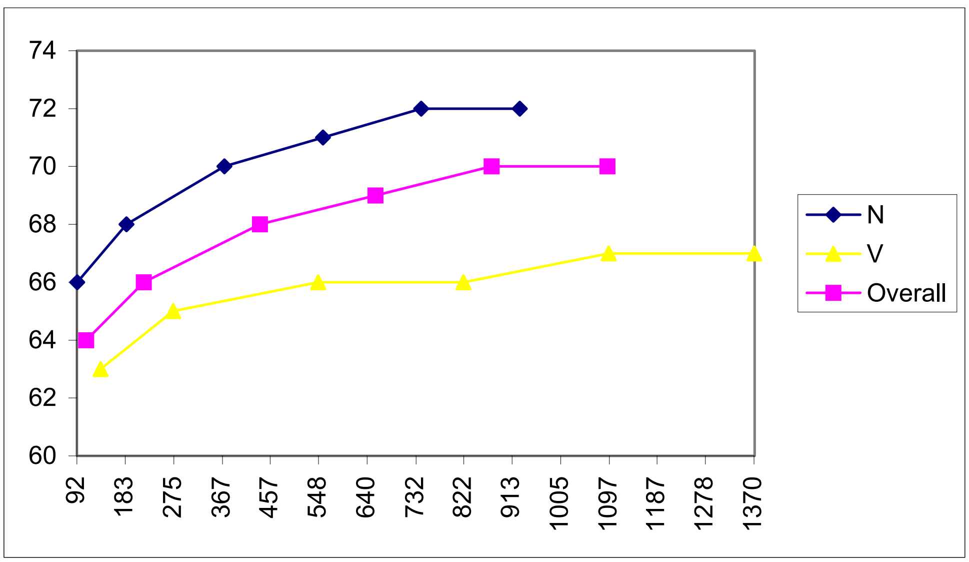

Figure 5: learning curve in the DSO corpus.

9.3. Noise: larger data is more resistant to noise

In order to analyze the effect of noise in the training data, we introduced some random tags in part of the examples. We created 4 new samples for training, with varying degrees of noise: 10% of the examples with random tags, 20%, 30% and 40%.

The results in precision/coverage and recall are illustrated in Table 17 and Table 18. Figures 6 and 7 show the recall data for Semcor and DSO. The decrease in recall is steady for both nouns and verbs in Semcor, but it is rather brusque in DSO. **This could mean that when more data is available, the system is more robust to noise**: the performance is hardly affected by 10%, 20% and 30% of noise.

	PoS	Prec. (40%)	Rec. (40%)	Prec. (30%)	Rec. (30%)	Prec. (20%)	Rec. (20%)	Prec. (10%)	Rec. (10%)	Prec. (0%)	Rec. (0%)
all	A	0.54/1.00	0.54	0.66/1.00	0.66	0.79/1.00	0.79	0.90/1.00	0.90	0.99/1.00	0.99
long	A	0.40/0.78	0.31	0.46/0.90	0.41	0.47/0.98	0.46	0.58/0.99	0.57	0.63/0.99	0.62
most	B	0.51/1.00	0.51	0.50/1.00	0.50	0.62/1.00	0.62	0.70/1.00	0.70	0.78/1.00	0.78
only	B	0.38/0.93	0.35	0.47/0.98	0.46	0.55/1.00	0.55	0.62/1.00	0.62	0.69/1.00	0.69
account	N	0.10/0.37	0.04	0.44/0.67	0.29	0.22/0.67	0.15	0.52/0.85	0.44	0.57/0.85	0.48
age	N	0.48/0.83	0.40	0.56/0.98	0.55	0.63/0.95	0.60	0.73/1.00	0.73	0.76/1.00	0.76
church	N	0.43/0.96	0.41	0.49/0.98	0.48	0.42/0.99	0.42	0.66/0.99	0.65	0.69/1.00	0.69
duty	N	0.37/0.76	0.28	0.71/0.96	0.68	0.55/0.84	0.46	0.58/0.96	0.56	0.61/0.92	0.56
head	N	0.52/0.93	0.48	0.60/0.98	0.59	0.73/1.00	0.73	0.81/0.99	0.80	0.88/1.00	0.88
interest	N	0.29/0.75	0.22	0.38/0.87	0.33	0.42/0.92	0.39	0.55/0.96	0.53	0.62/0.97	0.60
member	N	0.53/0.97	0.51	0.61/1.00	0.61	0.72/1.00	0.72	0.80/1.00	0.80	0.91/1.00	0.91
people	N	0.52/1.00	0.52	0.62/1.00	0.62	0.74/1.00	0.74	0.83/1.00	0.83	0.90/1.00	0.90
die	V	0.57/0.91	0.52	0.70/0.99	0.69	0.78/0.99	0.77	0.88/0.99	0.87	0.97/0.99	0.96
fall	V	0.33/0.46	0.15	0.28/0.48	0.13	0.24/0.48	0.12	0.41/0.62	0.25	0.34/0.71	0.24
give	V	0.16/0.40	0.06	0.24/0.54	0.13	0.26/0.59	0.15	0.29/0.67	0.19	0.34/0.78	0.27
include	V	0.44/0.97	0.43	0.52/0.97	0.50	0.52/0.98	0.51	0.60/0.97	0.58	0.70/0.99	0.69
know	V	0.35/0.81	0.28	0.44/0.94	0.41	0.50/0.98	0.49	0.57/1.00	0.57	0.61/1.00	0.61
seek	V	0.39/0.70	0.27	0.50/0.74	0.37	0.57/0.74	0.42	0.57/0.93	0.53	0.62/0.89	0.55
Understand	V	0.43/0.88	0.38	0.60/0.98	0.59	0.63/0.99	0.62	0.64/0.94	0.60	0.77/1.00	0.77
A		0.48/0.89	0.43	0.57/0.95	0.54	0.64/0.99	0.63	0.75/1.00	0.74	0.82/1.00	0.82
B		0.42/0.95	0.40	0.48/0.99	0.47	0.57/1.00	0.57	0.65/1.00	0.65	0.72/1.00	0.72
N		0.47/0.90	0.42	0.56/0.96	0.54	0.62/0.97	0.60	0.74/0.99	0.73	0.80/0.99	0.79
V		0.36/0.70	0.25	0.44/0.81	0.36	0.48/0.84	0.40	0.53/0.88	0.47	0.58/0.92	0.53
Overall		0.42/0.84	0.35	0.50/0.91	0.46	0.56/0.93	0.52	0.65/0.95	0.61	0.70/0.97	0.68

Table 17: Results with noise in Semcor.

	PoS	Prec. (40%)	Rec. (40%)	Prec. (30%)	Rec. (30%)	Prec. (20%)	Rec. (20%)	Prec. (10%)	Rec. (10%)	Prec. (0%)	Rec. (0%)
Age	N	0.39/0.96	0.37	0.65/0.99	0.64	0.67/0.99	0.66	0.68/0.99	0.67	0.73/1.00	0.73
Church	N	0.47/0.98	0.46	0.59/0.99	0.58	0.64/1.00	0.64	0.66/1.00	0.66	0.70/1.00	0.70
Head	N	0.44/0.95	0.42	0.72/1.00	0.72	0.74/1.00	0.74	0.77/1.00	0.77	0.79/1.00	0.79
Interest	N	0.41/0.92	0.38	0.62/1.00	0.62	0.62/0.99	0.61	0.61/0.99	0.60	0.62/1.00	0.62
Member	N	0.44/0.99	0.44	0.72/1.00	0.72	0.74/1.00	0.74	0.78/1.00	0.78	0.79/1.00	0.79
Fall	V	0.48/0.97	0.47	0.77/1.00	0.77	0.78/1.00	0.78	0.80/1.00	0.80	0.80/1.00	0.80
Give	V	0.46/0.96	0.44	0.74/1.00	0.74	0.76/1.00	0.76	0.76/1.00	0.76	0.77/1.00	0.77
Know	V	0.28/0.67	0.19	0.42/0.95	0.40	0.47/0.97	0.46	0.48/0.98	0.47	0.46/0.98	0.45
N		0.43/0.96	0.41	0.67/1.00	0.67	0.69/1.00	0.68	0.70/1.00	0.70	0.72/1.00	0.72
V		0.42/0.86	0.36	0.64/0.98	0.63	0.67/0.99	0.66	0.68/0.99	0.67	0.67/0.99	0.67
Overall		0.42/0.91	0.39	0.66/0.99	0.65	0.68/0.99	0.67	0.69/0.99	0.69	0.70/1.00	0.70

Table 18: Results with noise in DSO.

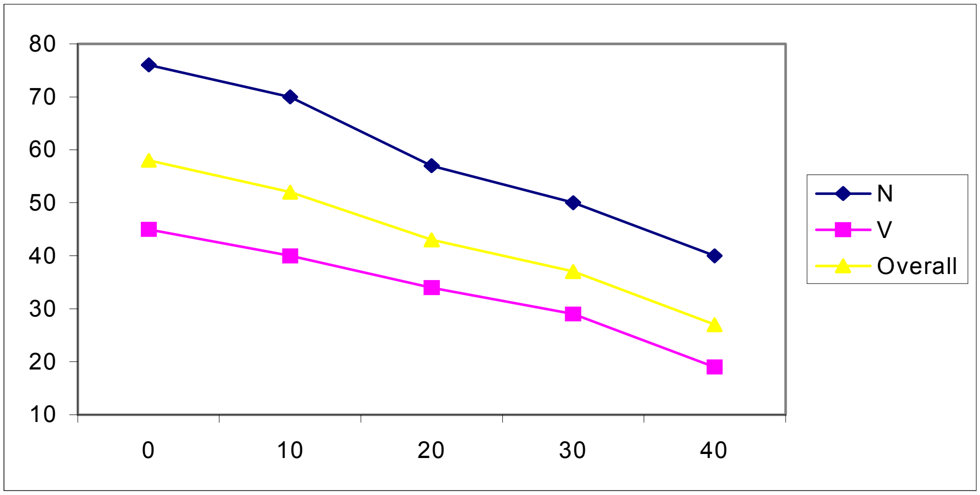

Figure 6: Results with noise in Semcor.

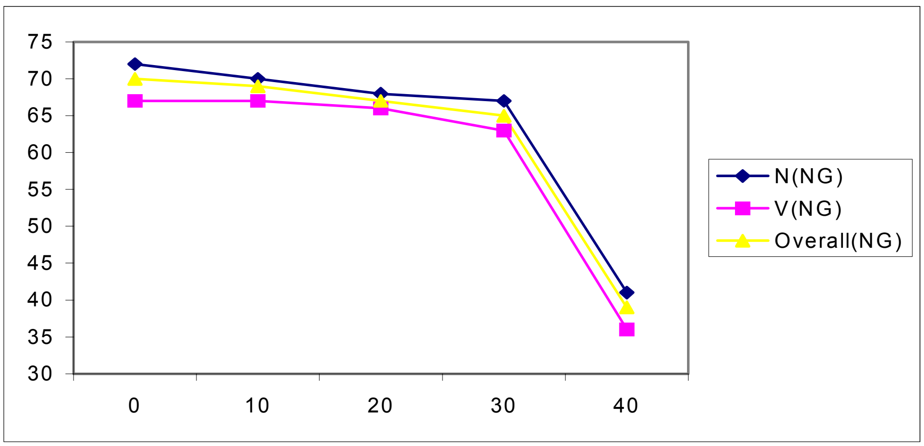

Figure 7: Results with noise in DSO.

9.4. Precision Vs Coverage

In the way explored by (Dagan and Itai, 1994) and (Leacock et al., 1998) we tried to improve the precision at the cost of coverage, not making decisions when the difference of the maximum likelihood among the senses was not big enough. For this purpose, a one-tailed confidence interval was created so we could state with confidence 1 - α that the true value of the difference measure was bigger than a given threshold (named θ). As in (Dagan and Itai, 1994), we adjusted the measure to the amount of evidence.

For each feature and sense, the lower bound $\beta_\alpha(sense_i)$ was calculated, using this formula:

$$\beta_\alpha(sense_i) = Log\left(\frac{N_i}{\sum_{j \neq i} Nj}\right) - Z_{1-\alpha}\sqrt{\frac{1}{Ni} + \frac{1}{\sum_{j \neq i} Nj}}$$

Where N_i denotes the frequency for the feature in the sense i, and $Z_{1-\alpha}$ is the confidence coefficient from the normal distribution.

The feature was excluded for the sense i when:

$$\beta_\alpha(\text{sense}_i) < \theta$$

Different values of θ were tested empirically. In Figure 8 and Figure 9 we see the results obtained in Semcor and DSO using a 60% confidence interval, as in (Dagan and Itai, 94). The values of θ range from 0 to 4.
We see that the use of this technique is profitable in DSO, where we can obtain results over 90% for a handful of examples. But the performance is worse in Semcor; even though the system works well with nouns, fails with verbs because of the scarcity of data.

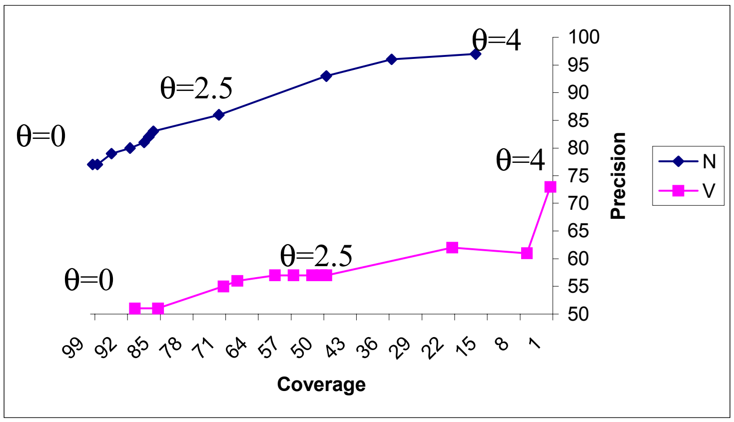

Figure 8: Precision Vs Coverage in Semcor.

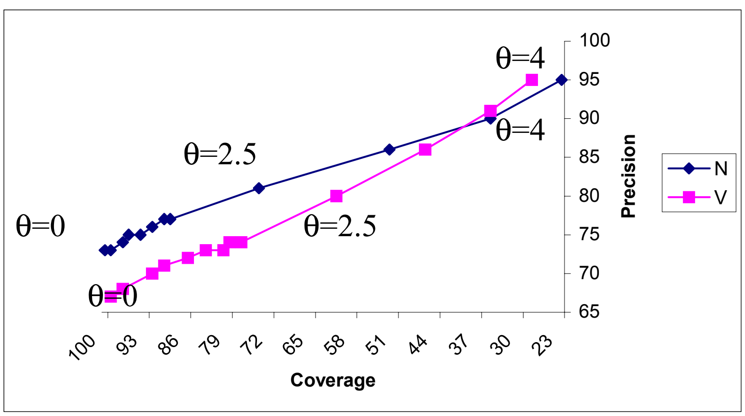

Figure 9: Precision Vs Coverage in DSO.

9.5. Coarse Senses: results reach 0.83 precision

It has been argued that the fine-grainedness of the sense distinctions in Semcor makes the task more difficult than necessary. WordNet allows making sense distinctions at the semantic file level, that is, two word senses that belong to the same semantic file can be taken as a single sense. We call the level of fine-grained original senses the synset level, and the coarser senses form the semantic file level.

In case any application finds these coarser senses useful, we trained the decision lists with these senses both in Semcor and DSO. The results are shown in Table 19 for the words in Set B. At this level the results on both corpora reach 83% of precision.

	PoS	Synsets	Semantic Fields	Synsets in Semcor	SF in Semcor	Synsets in DSO	SF in DSO
age	N	5	2	0.76/1.00	0.75/1.00	0.73/1.00	0.74/1.00
church	N	3	3	0.69/1.00	0.69/1.00	0.71/1.00	0.71/1.00
head	N	30	15	0.88/1.00	0.88/1.00	0.79/1.00	0.80/1.00
interest	N	7	5	0.62/0.97	0.67/0.99	0.62/1.00	0.72/1.00
member	N	5	4	0.91/1.00	0.91/1.00	0.79/1.00	0.79/1.00
fall	V	32	7	0.34/0.71	0.57/0.71	0.80/1.00	0.85/1.00
give	V	45	10	0.34/0.78	0.72/0.95	0.77/1.00	0.87/1.00
know	V	11	2	0.61/1.00	1.00/1.00	0.46/0.98	1.00/1.00
N		50	29	0.77/0.99	0.78/1.00	0.72/1.00	0.76/1.00
V		88	19	0.51/0.90	0.87/0.96	0.67/0.99	0.91/1.00
Overall		138	48	0.62/0.94	0.83/0.98	0.70/1.00	0.83/1.00

Table 19: Results disambiguating coarse senses.

9.6. Cross-tagging: hand taggers need to be coordinated

We wanted to check the performance of the decision lists training on one corpus and tagging the other. The DSO and Semcor corpora do not use exactly the same word sense system, as the former uses WordNet version 1.5 and the later WordNet version 1.6. We were able to easily map the senses form one to the other for all the words in Set B. We did not try to map the word senses that did not occur in any one of the corpora.

A previous study (Ng et al., 1999) used the fact that some sentences of the DSO corpus are also included in Semcor in order to study the agreement between the tags in both corpora. They showed that the hand-taggers of the DSO and Semcor teams only agree 57% of the time. This is a rather low figure, which explains why the results for one corpus or the other differ, e.g. the differences on the MFS results (see Table 4).

Considering this low agreement, we were not expecting good results on this cross-tagging experiment. The results shown in Table 20 confirmed our expectations, as the precision is greatly reduced (approximately one third in both corpora, but more than a half in the case of verbs). **Teams of hand-taggers need to be coordinated in order to produce results that are interchangeable**.

Word	PoS	Training Examples (in Semcor)	XMFS (in DSO)	XPrec./Cov. (in DSO)	Original Prec/Cov	Training Examples (in DSO)	XMFS (in Semcor)	XPrecCov (Semcor)	Original Prec/Cov
Age	N	104	0.62	0.67/0.97	0.76/1.00	491	0.72	0.63/1.00	0.73/1.00
Church	N	128	0.62	0.68/0.99	0.69/1.00	370	0.47	0.78/1.00	0.71/1.00
Head	N	179	0.40	0.40/0.97	0.88/1.00	866	0.03	0.77/1.00	0.79/1.00
Interest	N	140	0.18	0.37/0.90	0.62/0.97	1479	0.10	0.35/0.99	0.62/1.00
Member	N	74	0.74	0.74/0.97	0.91/1.00	1430	0.91	0.84/1.00	0.79/1.00
Fall	V	52	0.01	0.06/0.54	0.34/0.71	1408	0.04	0.32/0.96	0.80/1.00
Give	V	372	0.01	0.16/0.72	0.34/0.78	1262	0.09	0.15/1.00	0.77/1.00
Know	V	514	0.27	0.32/1.00	0.61/1.00	1441	0.14	0.44/0.98	0.46/0.98
N		125.00	0.48	0.55/0.95	0.77/0.99	927.20	0.35	0.66/1.00	0.72/1.00
V		312.67	0.10	0.21/0.76	0.51/0.90	1370.33	0.11	0.32/0.99	0.67/0.99
Overall		195.38	0.30	0.41/0.86	0.62/0.94	1093.38	0.21	0.46/0.99	0.70/1.00

Table 20: Cross tagging the corpora.

We examined the lowest performing words by hand. In order to explain the low performance of the cross-tagging, we analyzed the results by sense. In Table 21 we show three words for which the system performs badly: “head”, “fall” and “give”. For the word “fall”, for instance, there is a sense that dominates in DSO, represented by the concept *{fall, diminish, decrease, lessen}*. It appears in 1052 of the 1408 examples. However, it appears only twice in the 52 appearances of “fall” in Semcor. This causes this sense to be badly trained in semcor and diminishes its chances to be chosen when tagging DSO. Therefore none of the 1052 examples is hit by the decision lists, dropping dramatically the overall accuracy.

A similar behavior can be observed for the other mentioned words. In the case of “head”, one of the two dominating senses in DSO is again mistrained in Semcor, and his accuracy drops to %4. Finally, in the case of “give”, the two dominating senses in Semcor are mistrained in DSO (they only appear in 54 of 1262 examples) and as a result their performance is low.

Word	PoS	Total Senses	Sense	Examples in Semcor	Prec./Cov. in Semcor	Examples in DSO	Prec./Cov. in DSO
Fall	V	32	00103366	2	1.00/1.00	1052	0.00/0.49
			Overall	52	0.32/0.96	1408	0.06/0.54
Head	N	30	07311393	6	1.00/1.00	350	0.04/0.94
			04290247	140	0.92/1.00	350	0.93/1.00
			Overall	179	0.77/1.00	866	0.40/0.97
Give	V	45	01583087	80	0.05/1.00	16	0.69/0.81
			01597666	75	0.04/1.00	38	0.34/0.76
			Overall	372	0.15/1.00	1262	0.16/0.72

Table 21: Analysis of the results according to the senses.

10. Deriving training data from the Internet

In order to automatically derive training data from the Internet, we implemented (Mihalcea and Moldovan, 1999). The method uses information in WordNet (e.g. monosemous synonyms and glosses) to construct queries, which are later fed into a web search engine like Altavista. Four procedures can be used consecutively, in decreasing order of precision, but with increasing levels of examples retrieved. Mihalcea and Moldovan evaluated by hand 1080 retrieved instances of 120 word senses, and attested that 91% were correct. The method was not used to train a word sense disambiguation system.

In order to train our decision lists, we retrieved around 150 documents per word sense. The html documents were converted into ASCII texts, and segmented into paragraphs and sentences. We only used the sentence around the target to train the decision lists. As the gloss or synonyms were sometimes used to retrieve the text, we had to replace those with the target word.

The example below shows two senses of *church*, and two samples for each. For the first sense, part of the gloss, "*group of Christians*" was used to retrieve the example shown. For the second sense, the monosemous synonyms "*church building*".

Example: *church*

```
'church1' => {
'glos' => 'a group of Christians; any group professing Christian doctrine or belief; ',
'syns' => ['Christian church','Christianity'],
'proc1' => []
'proc2' => ['"group of Christians"','"group professing Christian doctrine"','"group professing Christian belief"'],
'proc3' => ['group NEAR Christians AND ("church" OR "Christian church" OR Christianity)',
            'group NEAR professing NEAR Christian NEAR doctrine AND ("church" OR "Christian church" OR Christianity)',
            'group NEAR professing NEAR Christian NEAR belief AND ("church" OR "Christian church" OR Christianity)'],
'proc4' => ['group AND Christians AND ("church" OR "Christian church" OR Christianity)',
            'group AND professing AND Christian AND doctrine AND ("church" OR "Christian church" OR Christianity)',
            'group AND professing AND Christian AND belief AND ("church" OR "Christian church" OR Christianity)'],

'church2' => {
'syns' => ['church building',
"glos' => 'for public (especially Christian) worship; ',
'proc1' => ['"church building"'],
'proc2' => ['"public worship"'],
'proc3' => ['public NEAR worship AND ("church" OR "church building")'],
'proc4' => ['public AND worship AND ("church" OR "church building")'],

'church3' => {
'glos' => 'a service conducted in a church; ',
'exam' => '"don\'t be late for church"',
'syns' => ['church service']
                    'proc1' => ['church service']
                    'proc2' => ['service conducted in a church'],
                    'proc3' => ['service NEAR conduct NEAR church AND ("church service")'],
                    'proc4' => ['service AND conduct AND church AND ("church service")'],


sisx05[154] nice searchword.pl church n altavista -type mih
parameters type:mih
church1 proc1
  > total: 0 url, 0 different, 150 left
  > total: 0 retrieved, 150 left
church1 proc2
  > data/urllist.church1.n.2.%22group+of+Christians%22.txt
    - 0 4492 %22group+of+Christians%22
  > data/urllist.church1.n.2.%22group+professing+Christian+doctrine%22.txt
    - 0 5 %22group+professing+Christian+doctrine%22
  > data/urllist.church1.n.2.%22group+professing+Christian+belief%22.txt
    - 0 0 %22group+professing+Christian+belief%22
  > total: 232 url, 232 different, 150 left
  > total: 156 retrieved, -6 left
church2 proc1
  > data/urllist.church2.n.1.%22church+building%22.txt
    - 0 50573 %22church+building%22
  > total: 766 url, 766 different, 150 left
  > total: 153 retrieved, -3 left
church3 proc1
  > data/urllist.church3.n.1.%22church+service%22.txt
```

- 0 37702 %22church+service%22
> total: 780 url, 780 different, 150 left
> total: 154 retrieved, -4 left

sense 1: 119
1 bart.northnet.com.au/worldview_society.html 2 "group_of_Christians" : In the nineteenth century a small >> church << in London - the Clapham Sect gave their support to Christian politician William Wilberforce in his long but fruitful anti-slavery campaign and to many other social and religious causes . :
1 www.oregonlive.com/st102910.html 2 "group_of_Christians" : Why is one >> church << satisfied and the other oppressed ? :

sense 2: 131
2 www.caverns.com/history8.html 1 "church_building" : Celestine formulated plans for a new >> church << . :
2 www.geocities.com/Genhist.htm 1 "church_building" : The result was a congregation formed at that place , and a >> church << erected . :
2 www.cityofthelord.org/secondspring.htm 1 "church_building" : The >> church << is a message to the world of that calling . :

sense 3 136
3 inst.augie.edu/Rudi9712.html 1 "church_service" : Since then she comes with her two children to the >> church << and now on December 1 she began in the diakonia station . :
3 www.damascus.com/video.html 1 "church_service" : Currently , it produces video clips for >> church << . :
3 www.rcc.ryerson.ca/CKCOTV_Kitchener_History.html 1 "church_service" : For over 40 years there has been a weekly >> church << from a local church on Sunday mornings . :

Several improvements can be made to the process, like using part-of-speech tagging and morphological processing to ensure that the replacement is correctly made, discarding suspicious[4] documents or sentences, etc. Besides (Leacock et al., 1998) and (Agirre et al., 2001) propose alternative strategies to construct the queries.

We used the Internet data to train the decision lists on the basic feature set and tag the Semcor data. The precision and coverage are shown in Table 22 compared to the precision in Semcor and the cross-tagging precision obtained training on DSO and tagging Semcor. For words with a high most frequent skew (e.g. *age, member, head*) the performance is significantly lower, but for the rest (*church*, *interest*) the performance is similar and much better than the cross-tagging exercise. The low performance apparently contradicts the good quality figures attained by (Mihalcea and Moldovan, 1999), reaching 90% of good examples. One possible explanation could be that the acquired examples, being correct themselves, provide systematically misleading features.

The work in (Leacock et al., 1998) also produces automatically training data, attaining results similar to hand-tagged data, but they are not always able to produce examples for some word senses.

Further work is needed to improve the quality of the acquired data, but considering that we use the raw data as it was, we think that the results are promising.

[4] Too long or too short, having many tables, indexes, etc.

Word	PoS	Data Source	Examples	Precision/Coverage
Church	N	Internet	380	0.65/0.98
		Semcor	128	0.69/1.00
		Xsemcor	370	0.78/1.00
Age	N	Internet	630	0.63/0.97
		Semcor	104	0.76/1.00
		Xsemcor	491	0.63/1.00
Interest	N	Internet	1039	0.56/0.93
		Semcor	140	0.62/0.97
		Xsemcor	1479	0.35/0.99
Member	N	Internet	694	0.39/0.91
		Semcor	74	0.91/1.00
		Xsemcor	1430	0.84/1.00
Head	N	Internet	3614	0.45/0.63
		Semcor	179	0.88/1.00
		Xsemcor	866	0.77/1.00
Know	V	Internet	1422	0.37/0.82
		Semcor	514	0.61/1.00
		Xsemcor	1441	0.44/0.98

Table 22: Results on Internet data.

11. Conclusions

This paper tries to tackle several questions regarding decision lists and supervised algorithms in general, in the context of word senses based on a widely used lexical resource like WordNet. In the introduction we have mentioned different problems affecting the scalability of these algorithms to real texts. We can summarize our conclusions on these issues along these lines:

1. **Word sense inventory:** we chose to work with WordNet 1.6. This gives us the possibility to compare our results with other works, and to use available lexical resources as Semcor or DSO.

2. **Unavailability of training data:** we tested how far we could go with existing hand-tagged corpora like Semcor and the DSO corpus. Besides we used a corpus automatically acquired from the Internet:

    - **Semcor:** it provides enough data to perform some basic general disambiguation, at 0.68 precision on any general running text. The performance on different words is surprisingly balanced, as ambiguity and number of examples are balanced in this corpus. The learning curve indicates that the data available for nouns could be close to being sufficient, but verbs have little available data.

    - **DSO:** provides large amounts of data for specific words, allowing for improved precision. It is nevertheless unable to overcome the 70% barrier. The learning curve also suggests that an upper bound has been reached for systems trained on Wordnet word senses and hand-tagged data.

    - **Internet**: the preliminary results shown in this paper are promising, but not conclusive.

3. **ML algorithm:** this paper shows that decision lists provide state-of-the-art results with simple and very fast means (it has been compared to Naive-Bayes, Exemplar based and Boosting algorithms). It is easy to include new features, and the method is robust enough when faced with spurious features. Besides decision lists are able to learn with low amounts of data.
4. **Features:** the basic set of features seems enough: contexts larger than windows do not provide much information, and introduce noise. Including lemmas, synsets or semantic files does not significantly alter the results. Using a simplified set of PoS tags (only 5 tags) does not degrade performance. Local features, i.e. collocations, are the strongest kind of features, but topical features enable to extend the coverage. The results obtained with syntactic features have been disappointing so far. We think that other ways should be explored, as the selection of features and the analysis of the kinds of words. Furthermore, the use of semantic knowledge in combination with syntactic features for WSD has been studied in (Agirre and Martinez, 2001).

We also performed experiments regarding the performance of the system under different conditions, and these conclusions were drawn:

- **Kinds of words:** the highest results can be expected for words with a dominating word sense. Nouns attain better performance with local features when enough data is provided. Individual words exhibit distinct behavior regarding the feature sets.
- **Cross-tagging**: the results are disappointing. Teams involved in hand-tagging need to coordinate with each other, at the risk of generating incompatible data.
- **Amount of data and noise**: Semcor is more affected by random noise than DSO. It could mean that higher amounts of data provide more robustness from noise.
- **Coarser word senses**: If decision lists are trained on coarser word senses inferred from WordNet itself, both Semcor and DSO attain more than 80% precision.
- **Precision Vs Coverage**: we observed that it is possible to obtain high precision at the cost of coverage. This technique could be useful for some applications or for bootstraping.

In related works, we have tested other kinds of features. We have applied semantic tags (Agirre and Martinez, 2000) and also selectional preferences to WSD (Agirre and Martinez, 2001). The results have not been conclusive, and our aim is to integrate these features with other sets. Also, Cross-tagging of corpora and the role of genre and topic have been further explored in (Martinez and Agirre, 2000).

12. Bibliography

Agirre E. and Rigau G. *Word Sense Disambiguation using Conceptual Density.* Proceedings of COLING'96, 16-22. Copenhagen (Denmark). 1996.

Agirre E. and Martínez D. *Exploring automatic word sense disambiguation with decision lists and the Web*. Proceedings of the Semantic Annotation And Intelligent Annotation workshop organized by COLING. Luxembourg, 2000.

Agirre E., Ansa O., Martínez D. and Hovy E. *Enriching WordNet concepts with topic signatures.* Proceedings of the NAACL workshop on WordNet and Other lexical Resources: Applications, Extensions and Customizations. Pittsburg, 2001.

Agirre E. and Martínez D. *Learning class-to-class selectional preferences.* Proceedings of the Workshop "Computational Natural Language Learning" (CoNLL-2001). In conjunction with ACL'2001/EACL'2001. Toulouse, France. 6-7th July 2001.

Chen S. F. *Building Probabilistic Models for Natural Language*. Ph.D. thesis. Technical Report TR-02-96, Center for Research in Computing Technology, Harvard University, 1996.

Collins M. *A new statistical parser based on bigram lexical dependencies.* In Proceedings of the 34th Annual Meeting of the ACL, pages 184-191. 1996.

Dagan I., and Itai A. *Word Sense Disambiguation Using a Second Language Monolingual Corpus.* Computational Linguistics 20:4, pp. 563--596. 1994.

Escudero G., Màrquez L. and Rigau G. *Naive Bayes and Exemplar-Based approaches to Word Sense Disambiguation Revisited*. Proceedings of the 14th European Conference on Artificial Intelligence, ECAI 2000. 2000.

Escudero G., Màrquez L. and Rigau G. *Boosting Applied to Word Sense Disambiguation*. Proceedings of the 12th European Conference on Machine Learning, ECML 2000. Barcelona, Spain. 2000.

Gale W., Church K. W., and Yarowsky D. *A Method for Disambiguating Word Senses in a Large Corpus*, Computers and the Humanities, 26, 415--439, 1993.

Ide, N. and Veronis J. *Introduction to the Special Issue on Word Sense Disambiguation: The State of the Art*. Computational Linguistics, 24(1), 1--40, 1998.

Kilgarriff, A. and Palmer M. (eds). *Special issue on SENSEVAL*. Computer and the Humanities, 34 (1-2). 2000.

Leacock, C., Chodorow M., and Miller G. A. *Using Corpus Statistics and WordNet Relations for Sense Identification.* Computational Linguistics, 24(1), 147--166, 1998.

Lin, D. *Principle Based parsing without Overgeneration*. In 31st Annual Meeting of the Association for Computational Linguistics. Columbus, Ohio. pp 112-120. 1993.

Martínez D. and Agirre E. *One Sense per Collocation and Genre/Topic Variations.* Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora. Hong Kong, 2000.

Mihalcea, R. and Moldovan I. *An Automatic Method for Generating Sense Tagged Corpora.* Proceedings of the 16th National Conference on Artificial Intelligence. AAAI Press, 1999.

Miller, G. A., Beckwith R., Fellbaum C., Gross D. and Miller K. *Five Papers on WordNet*. Special Issue of International Journal of Lexicography, 3(4), 1990.

Miller, G. A., Leacock C., Tengi R., and Bunker R. T. *A Semantic Concordance*. Proceedings of the ARPA Workshop on Human Language Technology, 1993.

Miller, G. A., Chodorow M., Landes S., Leacock C. and Thomas R. G. *Using a Semantic Concordance for Sense Identification.* Proceedings of the ARPA. 1994.

Ng, H. T. and Lee H. B. *Integrating Multiple Knowledge Sources to Disambiguate Word Sense: An Exemplar-based Approach.* Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics. 1996.

Ng, H. T. *Exemplar-Based Word Sense Disambiguation: Some Recent Improvements.* Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing, 1997.

Ng, H. T., Lim C. Y. and Foo S. K. *A Case Study on Inter-Annotator Agreement for Word Sense Disambiguation.* Proceedings of the Siglex-ACL Workshop on Standarizing Lexical Resources. 1999.

Pedersen, T. *A Decision Tree of Bigrams is an Accurate Predictor of Word Sense.* Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics (NAACL). 2001.

Senseval-2: Second International Workshop on Evaluating Word Sense Disambiguation Systems. 2001. http://www.sle.sharp.co.uk/senseval2

Sleator, D. and Temperley, D. *Parsing English with a Link Grammar*, *Third International Workshop on Parsing Technologies*. August 1993.

Stetina, J., Kurohashi S., Nagao M. *General Word Sense Disambiguation Method Based on a Full Sentential Context.* In Usage of WordNet in Natural Language Processing , Proceedings of COLING-ACL Workshop. Montreal (Canada). 1998.

Yarowsky, D. *Decision Lists for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French*. Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, pp. 88--95. 1994.

Yarowsky, D. *Unsupervised Word Sense Disambiguation Rivaling Supervised Methods.* Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics. Cambridge, MA, pp. 189-196, 1995.

Yarowsky, D. *Homograph Disambiguation in Text-to-speech Synthesis.* J Hirschburg, R. Sproat and J. Van Santen (eds.) Progress in Speech Synthesis, Springer-Vorlag, pp. 159-175. 1996.

Appendix A: Comparison between syntactic parsers

In order to extract useful syntactic features from the tagged examples, we needed a parser that would met the following requirements:

A- is free for research
B- provides syntactic relations directly (in contrast with partial parsers that only provide constituent structures, or parse trees without relation markings)
C- has been positively evaluated on well-established corpora
D- is fast enough for big corpora

We posted a question to the distribution list "corpora", and from the answers we elaborated a table to compare the systems. We did not have time to perform an exhaustive analysis of the different parsers, therefore we focused in the two that looked most promising from this table: Dekang Li's Minipar and Link Grammar.

	System	A	B	C	D	Comments
1.	Link Grammar	x	x	x	x	
2.	Collins	x	NO	x	x	
3.	Tilburg	NO	NO	x	?	
4.	Stevenson	x	x	NO	?	
5.	Schulte	x	x	x	?	Not portable (only for 7)
6.	Xerox	NO	x	x	?	
7.	Carroll	?	x	?	?	Probably not easy to install
8.	Dekang Lin's Minipar	x	x	x	?	

Appendix B: Complete list of syntactic relations provided by Minipar

In the following table, the complete list of relations provided by the Minipar parser is illustrated. For each relation we indicate his type, give a short description, and some examples and comments. In our analysis, we distinguish between 4 kinds of relations:

- Rel: main relation. We put here the relations that seem more useful for disambiguation, as object, subject...
- Aux: auxiliar relations. Auxiliar verbs, clauses...
- Fun: relations that seem irrelevant, but could help on disambiguation, as “age”.
- No: relations that seem useless for disambiguation, as “poss”.

Relation	rel	aux	fun	no	Description	Example	Comments
Abbrev			x		Abbreviation	NMR --> Nucleare ...	
Age			x		Age	John, 7, ...	
Amod	x				adverbial modif.	well --> thought, merely provide	
Appo			x		Apposition	John, director general, ...	
appo-mod			x		apposition moddif.		OFTEN WRONG
as-arg				x			
as1				x			
as2				x			
Aux		x			Aux. Verb	John should be promoted Joihn -s-> resign <-aux- should <-be- be	
Be		x			“be” as aux. Verb	is <-be- sleeping	
Being		x			“being” as aux verb		
by-subj	x				Subj. with passives		
C		x			clausal complement	... that <-c- John loves Mary (?) I go there for + infinitive clause go <-mod- (inf) <-c- for <-i- mainverb	
Cn		x?			nominalized clause	to issue is great be <-s inf <-cn inf <-i issue	OFTEN WRONG
comp1	x				complement (PP, inf/fin clause) of noun	... one of the boys one (N_P) <-comp1- of <-pcomp-n- boy .. grants to finance hospitals grants (N_C) <- c1- (inf) <-i- finance ... resolution which voted ... resolution (N_C) <-c1- (fin) <-i- voted	“boy in the garage” is MOD
comp2	x				?????		Few occurrences
Conj		x			Conjunction		Indirectly, to find obj
Desc	x				description	... make a man a child make <-desc- child .... become eclectic	Occurs frequently
Dest			x		Destination		OFTEN WRONG
Det	x				Determinant		

expletive	x				It, ...	it was disclosed: it -exp-> disclose it means, it seems	
fc	x				finite complement(?)	... said there is ... say <-fc- (fin) <-i- mainverb	
gen	x				Genitive	court's -gen-> ward	
guest	x				adjunts(?)	make house <-g- at church	
have		x			“have” as aux. Verb		
head		x			Dep. between query and main verb	should i go.... Q <-inv-aux- should <.head- go	
i		x			see c and fc, dep. between clause and main verb		
inside			x				
inv-aux		x			see head		
inv-be		x					
inv-have		x					
lex-dep	?	?			??	rep., mayor, Mr. ...	It has errors
lex-mod	?	?			?? multiword terms ?	oil-filed: field <-lex-mod- oil to edge up : edge<- up grand jury: jury <-lex-mod grand child welfare service "The Constitution" now and then	makes a single lexical entry: oilfiled, "edge up", "grand jury"
location			x				
mod	x				modifier	strikes increase as workers demand increase <-mod as <-comp1 fin <-i dema raises to cope with situation raise <-mod inf <-i cope <- mod with <-pcomp-n situation lost <-mod- already satisfactory -mod-> condition	
neg				x			
nn	x				noun-noun modifier, see also lex-mod	field services sector secotr <-nn field <-nn service	
obj	x				object		
obj2	x				indirect object		Sometimes wrong
p-spec		x			pp specifier	back -p-spec-> to	
pcomp-c	x				clause of pp	in voting itself in <-pcomp-c vpsc <-i- votig	
pcomp-n	x				nominal head of pp	in the house in <-pcomp-n house	
pnmod	x				postnominal mod.	person <-pnmod missing	
poss				x	only for 's		use gen
post	x				the thing after det	few ideas, the first man	
pre	x				the thig before det	all the men, such men	
pred	x				predicative (can be A or N)	John is beatuful (fin) <-i- is <-pred beautiful <-subj John	
rel		x			relative clause	earnings which grow earning <-rel fin <-whn which <-i grow	

s	x				surface subject, better to use subj		
sc	x				sentential complement	force John to do force <-sc-do	
self	x				himself...		
spellout			x				
subj	x						
vrel	x				passive verb modifier of nouns	fund <-vrel- granted	When "pnmod", is tagged as adj. (often wrongly), here is tagged as verb
wha				x			
whn				x			
whp				x			

Appendix C: Part of speech tags in Minipar

The meaning of the part of speech tags as is given in the documentation of Minipar:

Det: Determiners
PreDet: Pre-determiners (search for PreDet in data/wndict.lsp for instances)
PostDet: Post-determiners (search for PostDet in data/wndict.lsp for instances)
NUM: numbers
C: Clauses
I: Inflectional Phrases
V: Verb and Verb Phrases
N: Noun and Noun Phrases
NN: noun-noun modifiers
P: Preposition and Preposition Phrases
PpSpec: Specifiers of Preposition Phrases (search for PpSpe**c** in data/wndict.lsp for instances)
A: Adjective/Adverbs
Have: have
Aux: Auxilary verbs, e.g. should, will, does, ...
Be: Different forms of be: is, am, were, be, ...
COMP: Complementizer
VBE: be used as a linking verb. E.g., I am hungry
V_N verbs with one argument (the subject), i.e., intransitive verbs
V_N_N verbs with two arguments, i.e., transitive verbs
V_N_I verbs taking small clause as complement

Appendix D: Results for single features using Semcor

In the following 4 tables, the results obtained using single features are shown. The first two tables illustrate the results for nouns sorted by precision and recall, respectively. The last two tables are devoted to nouns. Many syntactic features do not appear in the training corpus, and are not included in the tables. Some semantic features that have been tested in other works have not been removed from the tables and appear as "basic" features (win_syn_4w, win_anc_0s, win_anc3_0s, win_hyper_0s, win_level4_0s...).

Results in Semcor for the whole set of features disambiguating nouns (sorted by precision)

Feature	Type	Prec.	Cov.	Recall
Mod_Prep_pcomp-n_N_word	indir	1,000	0,033	0,033
Mod_Prep_pcomp-n_N_synset	indir	1,000	0,031	0,031
Mod_lem	A-dir	1,000	0,010	0,010
Mod_synset	A-dir	1,000	0,010	0,010
Mod_word	A-dir	1,000	0,010	0,010
postl_lem	A-dir	1,000	0,007	0,007
postl_word	A-dir	1,000	0,007	0,007
Has_relat_mod_C_i_Vl	indir	1,000	0,007	0,007
sl_lem	A-dir	1,000	0,006	0,006
sl_synset	A-dir	1,000	0,006	0,006
sl_word	A-dir	1,000	0,006	0,006
subjl_lem	A-dir	1,000	0,006	0,006
subjl_synset	A-dir	1,000	0,006	0,006
subjl_word	A-dir	1,000	0,006	0,006
has_relat_mod_perl	B-dir	1,000	0,006	0,006
postl_synset	A-dir	1,000	0,005	0,005
has_relat_guestl	B-dir	1,000	0,005	0,005
has_relat_mod_froml	B-dir	1,000	0,005	0,005
genl_synset	A-dir	1,000	0,004	0,004
objl_lem	A-dir	1,000	0,004	0,004
objl_word	A-dir	1,000	0,004	0,004
has_relat_vrell	B-dir	1,000	0,004	0,004
has_relat_mod_forl	B-dir	1,000	0,003	0,003
conjl_lem	A-dir	1,000	0,002	0,002
conjl_synset	A-dir	1,000	0,002	0,002
guestl_lem	A-dir	1,000	0,002	0,002
guestl_word	A-dir	1,000	0,002	0,002
nn_lem	A-dir	1,000	0,002	0,002
nn_synset	A-dir	1,000	0,002	0,002
nn_word	A-dir	1,000	0,002	0,002
possl_lem	A-dir	1,000	0,002	0,002
possl_word	A-dir	1,000	0,002	0,002
vrell_lem	A-dir	1,000	0,002	0,002
vrell_synset	A-dir	1,000	0,002	0,002
vrell_word	A-dir	1,000	0,002	0,002
has_relat_appo	B-dir	1,000	0,002	0,002
has_relat_gen	B-dir	1,000	0,002	0,002
has_relat_mod_asl	B-dir	1,000	0,002	0,002
has_relat_mod_outl	B-dir	1,000	0,002	0,002
has_relat_possl	B-dir	1,000	0,002	0,002
comp1_C_i_V_lem	indir	1,000	0,002	0,002
comp1_C_i_V_synset	indir	1,000	0,002	0,002
comp1_C_i_V_word	indir	1,000	0,002	0,002
comp1_Prep_pcomp-n_Nl_lem	indir	1,000	0,002	0,002
comp1_Prep_pcomp-n_Nl_synset	indir	1,000	0,002	0,002
comp1_Prep_pcomp-n_Nl_word	indir	1,000	0,002	0,002
has_relat_s_CN_cn_C_i_Vl	indir	1,000	0,002	0,002
mod_Prep_pcomp-n_Nl_synset	indir	0,963	0,028	0,027
obj_word	A-dir	0,959	0,051	0,049
mod_Prep_pcomp-n_N_lem	indir	0,947	0,040	0,038
modl_synset	A-dir	0,941	0,071	0,067
obj_lem	A-dir	0,933	0,061	0,057
modl_lem	A-dir	0,926	0,085	0,079
modl_word	A-dir	0,926	0,085	0,079
mod_Prep_pcomp-n_Nl_word	indir	0,914	0,024	0,022
mod_Prep_pcomp-n_Nl_lem	indir	0,900	0,031	0,028
obj_synset	A-dir	0,887	0,055	0,049
trig_wf_+1	basic	0,881	0,237	0,209
trig_lem_+1	basic	0,878	0,248	0,218
nnl_synset	A-dir	0,868	0,031	0,027
nnl_lem	A-dir	0,857	0,044	0,038
detl_synset	A-dir	0,856	0,014	0,012
nnl_word	A-dir	0,854	0,043	0,037
big_wf_+1	basic	0,849	0,623	0,529
win_syn_4w	basic	0,848	0,413	0,350
big_lem_+1	basic	0,846	0,652	0,552
genl_lem	A-dir	0,840	0,120	0,101
genl_word	A-dir	0,836	0,118	0,099
win_wf_3w	basic	0,833	0,594	0,495

comp1_Prep_pcomp-n_N_lem	indir	0,833	0,006	0,005
comp1_Prep_pcomp-n_N_word	indir	0,833	0,006	0,005
trig_wf_0	basic	0,831	0,253	0,210
trig_lem_0	basic	0,824	0,256	0,211
win_wf_4w	basic	0,821	0,712	0,585
has_relat_postI	B-dir	0,815	0,012	0,010
has_relat_mod_ofI	B-dir	0,800	0,109	0,087
win_syn_50w	basic	0,799	0,999	0,798
win_syn_1s	basic	0,799	0,985	0,787
win_syn_20w	basic	0,798	0,974	0,777
trig_wf_-1	basic	0,791	0,167	0,132
has_relat_obj	B-dir	0,789	0,213	0,168
win_lem_50w	basic	0,788	1,000	0,788
trig_lem_-1	basic	0,787	0,173	0,136
has_relat_mod_inI	B-dir	0,787	0,027	0,021
has_relat_comp1_Prep_pcomp-n_N	indir	0,784	0,015	0,012
win_lem_0s	basic	0,783	1,000	0,783
win_syn_0s	basic	0,782	0,840	0,657
win_lem_1s	basic	0,781	1,000	0,781
win_lem_20w	basic	0,778	1,000	0,778
win_lem_4w	basic	0,774	0,998	0,772
win_anc_0s	basic	0,771	0,972	0,749
has_relat_genI	B-dir	0,771	0,157	0,121
has_relat_comp1_ofI	B-dir	0,769	0,014	0,011
win_anc3_0s	basic	0,768	0,971	0,746
Pred_synset	A-dir	0,767	0,018	0,014
Win_wf_0s	basic	0,766	0,974	0,746
Win_hyper_0s	basic	0,765	0,931	0,712
Win_level4_0s	basic	0,759	0,910	0,691
has_relat_subj	B-dir	0,754	0,162	0,122
pcomp-n_lem	A-dir	0,751	0,300	0,225
pcomp-n_word	A-dir	0,751	0,300	0,225
subj_lem	A-dir	0,750	0,050	0,038
has_relat_pred	B-dir	0,750	0,029	0,022
has_relat_comp1_Prep_pcomp-n_NI	indir	0,750	0,013	0,010
win_sf_20w	basic	0,745	1,000	0,745
win_sf_4w	basic	0,742	0,999	0,741
subj_word	A-dir	0,742	0,037	0,027
has_relat_objI	B-dir	0,742	0,028	0,021
s_word	A-dir	0,741	0,037	0,027
big_lem_-1	basic	0,740	0,704	0,521
big_wf_-1	basic	0,740	0,633	0,468
subj_synset	A-dir	0,738	0,046	0,034
trig_subpos_0	basic	0,735	0,816	0,600
trig_subpos_-1	basic	0,732	0,656	0,480
win_sf_0s	basic	0,730	1,000	0,730
win_sf_50w	basic	0,727	1,000	0,727
pred_lem	A-dir	0,723	0,026	0,019
has_relat_sI	B-dir	0,721	0,035	0,025
win_sf_1s	basic	0,717	1,000	0,717
trig_subpos_+1	basic	0,713	0,829	0,591
big_subpos_+1	basic	0,708	0,970	0,687
has_relat_s	B-dir	0,707	0,157	0,111
has_relat_nnI	B-dir	0,704	0,102	0,072
pred_Prep_pcomp-n_N_word	indir	0,700	0,005	0,004
has_relat_nn	B-dir	0,696	0,024	0,017
pred_word	A-dir	0,692	0,013	0,009
detI_word	A-dir	0,691	0,279	0,193
detI_lem	A-dir	0,689	0,285	0,196
s_lem	A-dir	0,688	0,047	0,032
has_relat_subjI	B-dir	0,688	0,041	0,028
has_relat_mod	B-dir	0,667	0,016	0,011
has_relat_mod_withI	B-dir	0,667	0,006	0,004
has_relat_comp1_C_i_V	indir	0,667	0,006	0,004
trig_pos_+1	basic	0,666	0,967	0,644
trig_pos_0	basic	0,663	0,973	0,645
trig_pos_-1	basic	0,648	0,828	0,537
big_pos_+1	basic	0,642	0,996	0,639
has_relat_mod_Prep_pcomp-n_N	indir	0,635	0,264	0,168
s_synset	A-dir	0,632	0,036	0,023
big_subpos_-1	basic	0,630	0,975	0,614
has_relat_detI	B-dir	0,615	0,304	0,187
has_relat_pnmodI	B-dir	0,608	0,005	0,003
has_relat_by-subj_Prep_pcomp-n_N	indir	0,608	0,005	0,003
has_relat_modI	B-dir	0,591	0,223	0,132
has_relat_conj	B-dir	0,586	0,034	0,020
big_pos_-1	basic	0,584	0,993	0,580
has_relat_mod_Prep_pcomp-n_NI	indir	0,560	0,209	0,117
has_relat_conjI	B-dir	0,558	0,037	0,021
has_relat_pcomp-n	B-dir	0,521	0,344	0,179
has_relat_mod_onI	B-dir	0,502	0,004	0,002
has_relat_lex-mod	B-dir	0,496	0,008	0,004
has_relat_appoI	B-dir	0,432	0,007	0,003
has_relat_mod_toI	B-dir	0,429	0,007	0,003
prel_lem	A-dir	0,333	0,003	0,001
prel_word	A-dir	0,333	0,003	0,001
has_relat_preI	B-dir	0,333	0,003	0,001
comp1_Prep_pcomp-n_N_synset	indir	0,333	0,003	0,001
has_relat_mod_atI	B-dir	0,294	0,007	0,002
has_relat_mod_Prep_pcomp-c_C_i_VI	indir	0,203	0,005	0,001
has_relat_pred_Prep_pcomp-n_N	indir	0,167	0,006	0,001
pred_Prep_pcomp-n_N_lem	indir	0,167	0,006	0,001
pred_Prep_pcomp-n_N_synset	indir	0,167	0,006	0,001

Results in Semcor for the whole set of features disambiguating verbs (sorted by precision)

Feature	Type	Prec.	Cov.	Recall
has_relat_descI	B-dir	1,000	0,003	0,003
conj_synset	A-dir	1,000	0,002	0,002
conjI_lem	A-dir	1,000	0,002	0,002
conjI_synset	A-dir	1,000	0,002	0,002
guestI_synset	A-dir	1,000	0,002	0,002
has_relat_mod_atI	B-dir	1,000	0,002	0,002
has_relat_mod_InI	B-dir	1,000	0,002	0,002
mod_C_i_V_synset	indir	0,846	0,010	0,008
sc_lem	A-dir	0,833	0,005	0,004
sc_word	A-dir	0,833	0,005	0,004
sc_synset	A-dir	0,800	0,004	0,003
mod_C_i_V_lem	indir	0,752	0,016	0,012
modI_synset	A-dir	0,734	0,012	0,009
has_relat_mod_C_i_V	indir	0,687	0,067	0,046
has_relat_mod_aboutI	B-dir	0,684	0,015	0,010
has_relat_by-subj_byI	B-dir	0,667	0,005	0,003
has_relat_by-subj_Prep_pcomp-n_NI	indir	0,667	0,005	0,003
has_relat_vrel	B-dir	0,634	0,009	0,006
fc_C_i_V_synset	indir	0,575	0,015	0,009
has_relat_amodI	B-dir	0,567	0,113	0,064
fc_C_i_VI_word	indir	0,567	0,071	0,040
has_relat_fc_C_i_VI	indir	0,566	0,164	0,093
mod_C_i_V_word	indir	0,566	0,011	0,006
modI_word	A-dir	0,558	0,019	0,011
fc_C_i_V_word	indir	0,554	0,014	0,008
amodI_synset	A-dir	0,548	0,078	0,043
trig_wf_0	basic	0,543	0,287	0,156
amodI_lem	A-dir	0,540	0,083	0,045
amodI_word	A-dir	0,540	0,083	0,045
sI_synset	A-dir	0,537	0,055	0,030
trig_lem_0	basic	0,533	0,315	0,168
mod_Prep_pcomp-n_NI_word	indir	0,528	0,018	0,010
trig_lem_-1	basic	0,527	0,252	0,133
has_relat_sc	B-dir	0,522	0,024	0,013
conj_lem	A-dir	0,518	0,003	0,002
modI_lem	A-dir	0,517	0,021	0,011
fc_C_i_VI_lem	indir	0,516	0,088	0,045
has_relat_comp1_C_i_V	indir	0,508	0,069	0,035
subjI_lem	A-dir	0,507	0,408	0,207
subjI_word	A-dir	0,507	0,369	0,187
big_lem_-1	basic	0,502	0,759	0,381
trig_wf_+1	basic	0,500	0,273	0,137
pred_C_i_V_word	indir	0,500	0,002	0,001
win_lem_20w	basic	0,499	1,000	0,499
win_lem_50w	basic	0,499	1,000	0,499
trig_wf_-1	basic	0,499	0,226	0,113
win_hyper_0s	basic	0,498	0,896	0,446
win_syn_50w	basic	0,495	1,000	0,495
sI_lem	A-dir	0,492	0,390	0,192
win_anc_0s	basic	0,491	0,932	0,458
win_anc3_0s	basic	0,490	0,929	0,455
win_wf_3w	basic	0,489	0,617	0,302
win_syn_1s	basic	0,488	0,988	0,482
big_wf_-1	basic	0,488	0,628	0,306
trig_lem_+1	basic	0,487	0,307	0,150
win_lem_0s	basic	0,486	1,000	0,486
win_syn_20w	basic	0,486	0,978	0,475
win_lem_4w	basic	0,485	1,000	0,485
win_wf_4w	basic	0,485	0,753	0,365
sI_word	A-dir	0,485	0,352	0,171
objI_synset	A-dir	0,485	0,072	0,035
subjI_synset	A-dir	0,483	0,063	0,030
win_wf_0s	basic	0,481	0,948	0,456
win_syn_0s	basic	0,480	0,811	0,389
win_lem_1s	basic	0,477	1,000	0,477
win_syn_4w	basic	0,476	0,479	0,228
auxI_lem	A-dir	0,474	0,208	0,099
auxI_word	A-dir	0,474	0,208	0,099
B-ngram3	B-ngram	0,472	0,664	0,313
big_lem_+1	basic	0,464	0,741	0,344
obj2I_synset	A-dir	0,464	0,022	0,010
has_relat_auxI	B-dir	0,462	0,230	0,106
win_sf_4w	basic	0,461	1,000	0,461
obj2I_lem	A-dir	0,459	0,029	0,013
win_level4_0s	basic	0,456	0,859	0,392
win_sf_50w	basic	0,454	1,000	0,454
win_sf_20w	basic	0,453	1,000	0,453
B-ngram1	B-ngram	0,453	0,997	0,452
big_wf_+1	basic	0,450	0,648	0,292
win_sf_1s	basic	0,449	1,000	0,449
win_sf_0s	basic	0,448	0,988	0,443
has_relat_subjI	B-dir	0,448	0,620	0,278
trig_subpos_0	basic	0,447	0,808	0,361
big_subpos_+1	basic	0,443	0,980	0,434
has_relat_sI	B-dir	0,443	0,603	0,267
big_subpos_-1	basic	0,442	0,985	0,435
has_relat_mod_Prep_pcomp-c_C_i_V	indir	0,439	0,012	0,005
has_relat_mod_asI	B-dir	0,436	0,005	0,002

big_pos_+1	basic	0,433	0,996	0,431
trig_pos_0	basic	0,431	0,980	0,422
trig_subpos_-1	basic	0,427	0,808	0,345
big_pos_-1	basic	0,426	0,995	0,424
trig_pos_-1	basic	0,422	0,922	0,389
has_relat_mod_byl	B-dir	0,422	0,006	0,003
B-ngram2	B-ngram	0,419	0,927	0,388
trig_pos_+1	basic	0,418	0,976	0,408
has_relat_mod_ofl	B-dir	0,416	0,046	0,019
bel_lem	A-dir	0,414	0,058	0,024
bel_word	A-dir	0,414	0,058	0,024
trig_subpos_+1	basic	0,412	0,828	0,341
objl_lem	A-dir	0,405	0,234	0,095
comp1_C_i_V_synset	indir	0,401	0,008	0,003
fc_C_i_V_lem	indir	0,396	0,019	0,008
obj2l_word	A-dir	0,394	0,026	0,010
has_relat_fc_C_i_V	indir	0,392	0,062	0,024
fc_C_i_Vl_synset	indir	0,386	0,058	0,022
mod_Prep_pcomp-n_Nl_lem	indir	0,384	0,025	0,010
objl_word	A-dir	0,383	0,203	0,078
havel_lem	A-dir	0,364	0,048	0,017
havel_word	A-dir	0,364	0,048	0,017
has_relat_s_CN_cn_C_i_V	indir	0,342	0,018	0,006
has_relat_s_CN_cn_C_i_Vl	indir	0,338	0,006	0,002
has_relat_mod_onl	B-dir	0,333	0,005	0,002
has_relat_mod_intol	B-dir	0,333	0,002	0,001
comp1_C_i_V_lem	indir	0,329	0,033	0,011

has_relat_bel	B-dir	0,301	0,062	0,019
comp1_C_i_V_word	indir	0,301	0,028	0,008
has_relat_modl	B-dir	0,286	0,072	0,021
has_relat_mod_Prep_pcomp-n_Nl	indir	0,278	0,173	0,048
has_relat_havel	B-dir	0,274	0,051	0,014
has_relat_conj	B-dir	0,243	0,020	0,005
has_relat_mod_C_i_Vl	indir	0,213	0,034	0,007
has_relat_objl	B-dir	0,200	0,521	0,104
has_relat_mod_inl	B-dir	0,192	0,031	0,006
mod_C_i_Vl_synset	indir	0,167	0,005	0,001
has_relat_conjl	B-dir	0,153	0,022	0,003
guestl_lem	A-dir	0,147	0,026	0,004
guestl_word	A-dir	0,147	0,026	0,004
mod_C_i_Vl_word	indir	0,143	0,006	0,001
mod_Prep_pcomp-n_Nl_synset	indir	0,133	0,012	0,002
pred_C_i_V_lem	indir	0,120	0,006	0,001
mod_C_i_Vl_lem	indir	0,111	0,007	0,001
has_relat_pred_C_i_V	indir	0,095	0,007	0,001
has_relat_guestl	B-dir	0,062	0,062	0,004
has_relat_mod_forl	B-dir	0,056	0,010	0,001
has_relat_mod_tol	B-dir	0,028	0,023	0,001
has_relat_obj2l	B-dir	0,014	0,096	0,001

Results in Semcor for the whole set of features disambiguating nouns (sorted by recall)

Feature	Type	Prec.	Cov.	Recall
Win_syn_50w	basic	0,799	0,999	0,798
Win_lem_50w	basic	0,788	1,000	0,788
win_syn_1s	basic	0,799	0,985	0,787
win_lem_0s	basic	0,783	1,000	0,783
win_lem_1s	basic	0,781	1,000	0,781
win_lem_20w	basic	0,778	1,000	0,778
win_syn_20w	basic	0,798	0,974	0,777
win_lem_4w	basic	0,774	0,998	0,772
win_anc_0s	basic	0,771	0,972	0,749
win_anc3_0s	basic	0,768	0,971	0,746
win_wf_0s	basic	0,766	0,974	0,746
win_sf_20w	basic	0,745	1,000	0,745
win_sf_4w	basic	0,742	0,999	0,741
win_sf_0s	basic	0,730	1,000	0,730
win_sf_50w	basic	0,727	1,000	0,727
win_sf_1s	basic	0,717	1,000	0,717
win_hyper_0s	basic	0,765	0,931	0,712
win_level4_0s	basic	0,759	0,910	0,691
big_subpos_+1	basic	0,708	0,970	0,687
win_syn_0s	basic	0,782	0,840	0,657
trig_pos_0	basic	0,663	0,973	0,645
trig_pos_+1	basic	0,666	0,967	0,644
big_pos_+1	basic	0,642	0,996	0,639
big_subpos_-1	basic	0,630	0,975	0,614
trig_subpos_0	basic	0,735	0,816	0,600
trig_subpos_+1	basic	0,713	0,829	0,591
win_wf_4w	basic	0,821	0,712	0,585
big_pos_-1	basic	0,584	0,993	0,580
big_lem_+1	basic	0,846	0,652	0,552
trig_pos_-1	basic	0,648	0,828	0,537
big_wf_+1	basic	0,849	0,623	0,529
big_lem_-1	basic	0,740	0,704	0,521
win_wf_3w	basic	0,833	0,594	0,495
trig_subpos_-1	basic	0,732	0,656	0,480
big_wf_-1	basic	0,740	0,633	0,468
win_syn_4w	basic	0,848	0,413	0,350
pcomp-n_lem	A-dir	0,751	0,300	0,225
pcomp-n_word	A-dir	0,751	0,300	0,225
trig_lem_+1	basic	0,878	0,248	0,218
trig_lem_0	basic	0,824	0,256	0,211
trig_wf_0	basic	0,831	0,253	0,210
trig_wf_+1	basic	0,881	0,237	0,209
detl_lem	A-dir	0,689	0,285	0,196
detl_word	A-dir	0,691	0,279	0,193
has_relat_detl	B-dir	0,615	0,304	0,187
Has_relat_pcomp-n	B-dir	0,521	0,344	0,179
has_relat_obj	B-dir	0,789	0,213	0,168
has_relat_mod_Prep_pcomp-n_N	indir	0,635	0,264	0,168
trig_lem_-1	basic	0,787	0,173	0,136
trig_wf_-1	basic	0,791	0,167	0,132
has_relat_modl	B-dir	0,591	0,223	0,132
has_relat_subj	B-dir	0,754	0,162	0,122
has_relat_genl	B-dir	0,771	0,157	0,121
has_relat_mod_Prep_pcomp-n_Nl	indir	0,560	0,209	0,117
has_relat_s	B-dir	0,707	0,157	0,111
genl_lem	A-dir	0,840	0,120	0,101
genl_word	A-dir	0,836	0,118	0,099
has_relat_mod_ofl	B-dir	0,800	0,109	0,087
modl_lem	A-dir	0,926	0,085	0,079
modl_word	A-dir	0,926	0,085	0,079
has_relat_nnl	B-dir	0,704	0,102	0,072
modl_synset	A-dir	0,941	0,071	0,067
obj_lem	A-dir	0,933	0,061	0,057
obj_word	A-dir	0,959	0,051	0,049
obj_synset	A-dir	0,887	0,055	0,049
mod_Prep_pcomp-n_N_lem	indir	0,947	0,040	0,038
nnl_lem	A-dir	0,857	0,044	0,038
subj_lem	A-dir	0,750	0,050	0,038
nnl_word	A-dir	0,854	0,043	0,037
subj_synset	A-dir	0,738	0,046	0,034
Mod_Prep_pcomp-n_N_word	indir	1,000	0,033	0,033
s_lem	A-dir	0,688	0,047	0,032
Mod_Prep_pcomp-n_N_synset	indir	1,000	0,031	0,031
mod_Prep_pcomp-n_Nl_lem	indir	0,900	0,031	0,028
has_relat_subjl	B-dir	0,688	0,041	0,028
mod_Prep_pcomp-n_Nl_synset	indir	0,963	0,028	0,027
nnl_synset	A-dir	0,868	0,031	0,027
subj_word	A-dir	0,742	0,037	0,027
s_word	A-dir	0,741	0,037	0,027
has_relat_sl	B-dir	0,721	0,035	0,025
s_synset	A-dir	0,632	0,036	0,023
mod_Prep_pcomp-n_Nl_word	indir	0,914	0,024	0,022
has_relat_pred	B-dir	0,750	0,029	0,022
has_relat_mod_inl	B-dir	0,787	0,027	0,021
has_relat_objl	B-dir	0,742	0,028	0,021
has_relat_conjl	B-dir	0,558	0,037	0,021
has_relat_conj	B-dir	0,586	0,034	0,020
pred_lem	A-dir	0,723	0,026	0,019

has_relat_nn	B-dir	0,696	0,024	0,017
pred_synset	A-dir	0,767	0,018	0,014
detl_synset	A-dir	0,856	0,014	0,012
has_relat_comp1_Prep_pcomp-n_N	indir	0,784	0,015	0,012
has_relat_comp1_ofl	B-dir	0,769	0,014	0,011
has_relat_mod	B-dir	0,667	0,016	0,011
Mod_lem	A-dir	1,000	0,010	0,010
Mod_synset	A-dir	1,000	0,010	0,010
mod_word	A-dir	1,000	0,010	0,010
has_relat_postl	B-dir	0,815	0,012	0,010
has_relat_comp1_Prep_pcomp-n_Nl	indir	0,750	0,013	0,010
pred_word	A-dir	0,692	0,013	0,009
postl_lem	A-dir	1,000	0,007	0,007
postl_word	A-dir	1,000	0,007	0,007
has_relat_mod_C_i_Vl	indir	1,000	0,007	0,007
sl_lem	A-dir	1,000	0,006	0,006
sl_synset	A-dir	1,000	0,006	0,006
sl_word	A-dir	1,000	0,006	0,006
subjl_lem	A-dir	1,000	0,006	0,006
subjl_synset	A-dir	1,000	0,006	0,006
subjl_word	A-dir	1,000	0,006	0,006
has_relat_mod_perl	B-dir	1,000	0,006	0,006
postl_synset	A-dir	1,000	0,005	0,005
has_relat_guestl	B-dir	1,000	0,005	0,005
has_relat_mod_froml	B-dir	1,000	0,005	0,005
comp1_Prep_pcomp-n_N_lem	indir	0,833	0,006	0,005
comp1_Prep_pcomp-n_N_word	indir	0,833	0,006	0,005
genl_synset	A-dir	1,000	0,004	0,004
objl_lem	A-dir	1,000	0,004	0,004
objl_word	A-dir	1,000	0,004	0,004
has_relat_vrell	B-dir	1,000	0,004	0,004
pred_Prep_pcomp-n_N_word	indir	0,700	0,005	0,004
has_relat_mod_withl	B-dir	0,667	0,006	0,004
has_relat_comp1_C_i_V	indir	0,667	0,006	0,004
has_relat_lex-mod	B-dir	0,496	0,008	0,004
has_relat_mod_forl	B-dir	1,000	0,003	0,003
has_relat_pnmodl	B-dir	0,608	0,005	0,003
has_relat_by-subj_Prep_pcomp-n_N	indir	0,608	0,005	0,003
has_relat_appol	B-dir	0,432	0,007	0,003
has_relat_mod_tol	B-dir	0,429	0,007	0,003
conjl_lem	A-dir	1,000	0,002	0,002
conjl_synset	A-dir	1,000	0,002	0,002
guestl_lem	A-dir	1,000	0,002	0,002
guestl_word	A-dir	1,000	0,002	0,002
nn_lem	A-dir	1,000	0,002	0,002
nn_synset	A-dir	1,000	0,002	0,002

nn_word	A-dir	1,000	0,002	0,002
possl_lem	A-dir	1,000	0,002	0,002
possl_word	A-dir	1,000	0,002	0,002
vrell_lem	A-dir	1,000	0,002	0,002
vrell_synset	A-dir	1,000	0,002	0,002
vrell_word	A-dir	1,000	0,002	0,002
has_relat_appo	B-dir	1,000	0,002	0,002
has_relat_gen	B-dir	1,000	0,002	0,002
has_relat_mod_asl	B-dir	1,000	0,002	0,002
has_relat_mod_outl	B-dir	1,000	0,002	0,002
has_relat_possl	B-dir	1,000	0,002	0,002
comp1_C_i_V_lem	indir	1,000	0,002	0,002
comp1_C_i_V_synset	indir	1,000	0,002	0,002
comp1_C_i_V_word	indir	1,000	0,002	0,002
comp1_Prep_pcomp-n_Nl_lem	indir	1,000	0,002	0,002
comp1_Prep_pcomp-n_Nl_synset	indir	1,000	0,002	0,002
comp1_Prep_pcomp-n_Nl_word	indir	1,000	0,002	0,002
has_relat_s_CN_cn_C_i_Vl	indir	1,000	0,002	0,002
has_relat_mod_onl	B-dir	0,502	0,004	0,002
has_relat_mod_atl	B-dir	0,294	0,007	0,002
prel_lem	A-dir	0,333	0,003	0,001
prel_word	A-dir	0,333	0,003	0,001
has_relat_prel	B-dir	0,333	0,003	0,001
comp1_Prep_pcomp-n_N_synset	indir	0,333	0,003	0,001
has_relat_mod_Prep_pcomp-c_C_i_Vl	indir	0,203	0,005	0,001
has_relat_pred_Prep_pcomp-n_N	indir	0,167	0,006	0,001
pred_Prep_pcomp-n_N_lem	indir	0,167	0,006	0,001
pred_Prep_pcomp-n_N_synset	indir	0,167	0,006	0,001

Results in Semcor for the whole set of features disambiguating verbs (sorted by recall)

Feature	Type	Prec.	Cov.	Recall
win_lem_20w	basic	0,499	1,000	0,499
win_lem_50w	basic	0,499	1,000	0,499
win_syn_50w	basic	0,495	1,000	0,495
win_lem_0s	basic	0,486	1,000	0,486
win_lem_4w	basic	0,485	1,000	0,485
win_syn_1s	basic	0,488	0,988	0,482
win_lem_1s	basic	0,477	1,000	0,477
win_syn_20w	basic	0,486	0,978	0,475
win_sf_4w	basic	0,461	1,000	0,461
win_anc_0s	basic	0,491	0,932	0,458
win_wf_0s	basic	0,481	0,948	0,456
win_anc3_0s	basic	0,490	0,929	0,455
win_sf_50w	basic	0,454	1,000	0,454
win_sf_20w	basic	0,453	1,000	0,453
B-ngram1	B-ngram	0,453	0,997	0,452
win_sf_1s	basic	0,449	1,000	0,449
win_hyper_0s	basic	0,498	0,896	0,446
win_sf_0s	basic	0,448	0,988	0,443
big_subpos_-1	basic	0,442	0,985	0,435
big_subpos_+1	basic	0,443	0,980	0,434
big_pos_+1	basic	0,433	0,996	0,431
big_pos_-1	basic	0,426	0,995	0,424
trig_pos_0	basic	0,431	0,980	0,422
trig_pos_+1	basic	0,418	0,976	0,408
win_level4_0s	basic	0,456	0,859	0,392
win_syn_0s	basic	0,480	0,811	0,389
trig_pos_-1	basic	0,422	0,922	0,389
B-ngram2	B-ngram	0,419	0,927	0,388
big_lem_-1	basic	0,502	0,759	0,381
win_wf_4w	basic	0,485	0,753	0,365
trig_subpos_0	basic	0,447	0,808	0,361
trig_subpos_-1	basic	0,427	0,808	0,345
big_lem_+1	basic	0,464	0,741	0,344
trig_subpos_+1	basic	0,412	0,828	0,341
B-ngram3	B-ngram	0,472	0,664	0,313
big_wf_-1	basic	0,488	0,628	0,306
win_wf_3w	basic	0,489	0,617	0,302
big_wf_+1	basic	0,450	0,648	0,292
has_relat_subjI	B-dir	0,448	0,620	0,278
has_relat_sI	B-dir	0,443	0,603	0,267
win_syn_4w	basic	0,476	0,479	0,228
subjI_lem	A-dir	0,507	0,408	0,207
sI_lem	A-dir	0,492	0,390	0,192
subjI_word	A-dir	0,507	0,369	0,187
sI_word	A-dir	0,485	0,352	0,171
trig_lem_0	basic	0,533	0,315	0,168
trig_wf_0	basic	0,543	0,287	0,156
trig_lem_+1	basic	0,487	0,307	0,150
trig_wf_+1	basic	0,500	0,273	0,137
trig_lem_-1	basic	0,527	0,252	0,133
trig_wf_-1	basic	0,499	0,226	0,113
has_relat_auxI	B-dir	0,462	0,230	0,106
has_relat_objI	B-dir	0,200	0,521	0,104
auxI_lem	A-dir	0,474	0,208	0,099
auxI_word	A-dir	0,474	0,208	0,099
objI_lem	A-dir	0,405	0,234	0,095
has_relat_fc_C_i_VI	indir	0,566	0,164	0,093
objI_word	A-dir	0,383	0,203	0,078
has_relat_amodI	B-dir	0,567	0,113	0,064
has_relat_mod_Prep_pcomp-n_NI	indir	0,278	0,173	0,048
has_relat_mod_C_i_V	indir	0,687	0,067	0,046
amodI_lem	A-dir	0,540	0,083	0,045
amodI_word	A-dir	0,540	0,083	0,045
fc_C_i_VI_lem	indir	0,516	0,088	0,045
amodI_synset	A-dir	0,548	0,078	0,043
fc_C_i_VI_word	indir	0,567	0,071	0,040
has_relat_comp1_C_i_V	indir	0,508	0,069	0,035
objI_synset	A-dir	0,485	0,072	0,035
sI_synset	A-dir	0,537	0,055	0,030
subjI_synset	A-dir	0,483	0,063	0,030
beI_lem	A-dir	0,414	0,058	0,024
beI_word	A-dir	0,414	0,058	0,024
has_relat_fc_C_i_V	indir	0,392	0,062	0,024
fc_C_i_VI_synset	indir	0,386	0,058	0,022
has_relat_modI	B-dir	0,286	0,072	0,021
has_relat_mod_ofI	B-dir	0,416	0,046	0,019
has_relat_beI	B-dir	0,301	0,062	0,019
haveI_lem	A-dir	0,364	0,048	0,017
haveI_word	A-dir	0,364	0,048	0,017
has_relat_haveI	B-dir	0,274	0,051	0,014
has_relat_sc	B-dir	0,522	0,024	0,013
obj2I_lem	A-dir	0,459	0,029	0,013
mod_C_i_V_lem	indir	0,752	0,016	0,012
modI_word	A-dir	0,558	0,019	0,011
modI_lem	A-dir	0,517	0,021	0,011
comp1_C_i_V_lem	indir	0,329	0,033	0,011
has_relat_mod_aboutI	B-dir	0,684	0,015	0,010
mod_Prep_pcomp-n_NI_word	indir	0,528	0,018	0,010

obj2I_synset	A-dir	0,464	0,022	0,010
obj2I_word	A-dir	0,394	0,026	0,010
mod_Prep_pcomp-n_NI_lem	indir	0,384	0,025	0,010
modI_synset	A-dir	0,734	0,012	0,009
fc_C_i_V_synset	indir	0,575	0,015	0,009
mod_C_i_V_synset	indir	0,846	0,010	0,008
fc_C_i_V_word	indir	0,554	0,014	0,008
fc_C_i_V_lem	indir	0,396	0,019	0,008
comp1_C_i_V_word	indir	0,301	0,028	0,008
has_relat_mod_C_i_VI	indir	0,213	0,034	0,007
has_relat_vrel	B-dir	0,634	0,009	0,006
mod_C_i_V_word	indir	0,566	0,011	0,006
has_relat_s_CN_cn_C_i_V	indir	0,342	0,018	0,006
has_relat_mod_inI	B-dir	0,192	0,031	0,006
has_relat_mod_Prep_pcomp-c_C_i_V	indir	0,439	0,012	0,005
has_relat_conj	B-dir	0,243	0,020	0,005
sc_lem	A-dir	0,833	0,005	0,004
sc_word	A-dir	0,833	0,005	0,004
guestI_lem	A-dir	0,147	0,026	0,004
guestI_word	A-dir	0,147	0,026	0,004
has_relat_guestI	B-dir	0,062	0,062	0,004
has_relat_descI	B-dir	1,000	0,003	0,003
sc_synset	A-dir	0,800	0,004	0,003
has_relat_by-subj_byI	B-dir	0,667	0,005	0,003
has_relat_by-subj_Prep_pcomp-n_NI	indir	0,667	0,005	0,003
has_relat_mod_byI	B-dir	0,422	0,006	0,003

comp1_C_i_V_synset	indir	0,401	0,008	0,003
has_relat_conjI	B-dir	0,153	0,022	0,003
conj_synset	A-dir	1,000	0,002	0,002
conjI_lem	A-dir	1,000	0,002	0,002
conjI_synset	A-dir	1,000	0,002	0,002
guestI_synset	A-dir	1,000	0,002	0,002
has_relat_mod_atI	B-dir	1,000	0,002	0,002
has_relat_mod_InI	B-dir	1,000	0,002	0,002
conj_lem	A-dir	0,518	0,003	0,002
has_relat_mod_asI	B-dir	0,436	0,005	0,002
has_relat_s_CN_cn_C_i_VI	indir	0,338	0,006	0,002
has_relat_mod_onI	B-dir	0,333	0,005	0,002
mod_Prep_pcomp-n_NI_synset	indir	0,133	0,012	0,002
pred_C_i_V_word	indir	0,500	0,002	0,001
has_relat_mod_intoI	B-dir	0,333	0,002	0,001
mod_C_i_VI_synset	indir	0,167	0,005	0,001
mod_C_i_VI_word	indir	0,143	0,006	0,001
pred_C_i_V_lem	indir	0,120	0,006	0,001
mod_C_i_VI_lem	indir	0,111	0,007	0,001
has_relat_pred_C_i_V	indir	0,095	0,007	0,001
has_relat_mod_forI	B-dir	0,056	0,010	0,001
has_relat_mod_toI	B-dir	0,028	0,023	0,001
has_relat_obj2I	B-dir	0,014	0,096	0,001

Appendix E: Learning Curve Tables

	PoS	Occ. (10%)	Prec. (10%)	Rec. (10%)	Occ. (20%)	Prec. (20%)	Rec. (20%)	Occ. (40%)	Prec. (40%)	Rec. (40%)
all	A	21	0.99/0.99	0.98	42	0.99/0.99	0.98	84	0.99/1.00	0.99
long	A	19	0.50/0.81	0.41	38	0.57/0.98	0.56	77	0.59/1.00	0.59
most	B	23	0.78/0.97	0.76	47	0.77/1.00	0.77	95	0.75/1.00	0.75
only	B	49	0.65/0.98	0.64	99	0.66/0.99	0.65	199	0.67/1.00	0.67
account	N	2	0.00/0.00	0.00	5	0.67/0.33	0.22	10	0.56/0.59	0.33
age	N	10	0.69/0.80	0.55	20	0.70/0.96	0.67	41	0.74/0.97	0.72
church	N	12	0.55/0.80	0.44	25	0.53/0.92	0.49	51	0.65/0.98	0.64
duty	N	2	0.00/0.08	0.00	5	1.00/0.12	0.12	10	0.61/0.72	0.44
head	N	17	0.85/0.91	0.77	35	0.87/0.97	0.84	71	0.89/1.00	0.89
interest	N	14	0.41/0.57	0.23	28	0.50/0.82	0.41	56	0.59/0.96	0.57
member	N	7	0.88/0.86	0.76	14	0.89/0.99	0.88	29	0.90/1.00	0.90
people	N	28	0.89/0.96	0.85	56	0.90/1.00	0.90	112	0.89/1.00	0.89
die	V	7	0.95/0.54	0.51	14	0.97/0.80	0.78	29	0.97/0.95	0.92
fall	V	5	0.23/0.25	0.06	10	0.44/0.31	0.14	20	0.28/0.48	0.13
give	V	37	0.27/0.45	0.12	74	0.23/0.41	0.09	148	0.30/0.62	0.19
include	V	14	0.76/0.79	0.60	28	0.71/0.91	0.65	57	0.74/0.94	0.70
know	V	51	0.54/0.95	0.51	102	0.56/0.97	0.54	205	0.57/0.99	0.56
seek	V	4	0.50/0.22	0.11	9	0.37/0.59	0.22	18	0.47/0.70	0.33
understand	V	8	0.73/0.52	0.38	16	0.76/0.81	0.62	33	0.72/0.94	0.68
A		20.00	0.78/0.90	0.71	40.00	0.79/0.99	0.78	80.50	0.80/1.00	0.80
B		36.00	0.69/0.98	0.68	73.00	0.70/0.99	0.69	147.0	0.70/1.00	0.70
N		11.50	0.76/0.80	0.61	23.50	0.77/0.91	0.70	47.50	0.79/0.97	0.76
V		18.00	0.54/0.68	0.37	36.14	0.56/0.74	0.41	72.86	0.56/0.84	0.47
Overall		17.37	0.67/0.81	0.54	35.11	0.69/0.87	0.60	70.79	0.69/0.93	0.64

Table 23: Learning Curve in the Semcor corpus. (10%-40%).

	PoS	Occ. (60%)	Prec. (60%)	Rec. (60%)	Occ. (80%)	Prec. (80%)	Rec. (80%)	Occ. (100%)	Prec. (100%)	Rec. (100%)
All	A	126	0.99/1.00	0.99	168	0.99/1.00	0.99	211	0.99/1.00	0.99
Long	A	115	0.61/1.00	0.61	154	0.66/1.00	0.66	193	0.63/0.99	0.62
Most	B	142	0.76/1.00	0.76	190	0.81/1.00	0.81	238	0.78/1.00	0.78
Only	B	299	0.72/1.00	0.72	399	0.70/1.00	0.70	499	0.69/1.00	0.69
Account	N	16	0.35/0.85	0.30	21	0.52/0.81	0.42	27	0.57/0.85	0.48
Age	N	62	0.75/0.99	0.74	83	0.77/1.00	0.77	104	0.76/1.00	0.76
Church	N	76	0.57/0.99	0.56	102	0.65/1.00	0.65	128	0.69/1.00	0.69
Duty	N	15	0.39/0.88	0.34	20	0.59/0.92	0.54	25	0.61/0.92	0.56
Head	N	107	0.88/0.99	0.87	143	0.89/1.00	0.89	179	0.88/1.00	0.88
Interest	N	84	0.58/0.95	0.55	112	0.62/0.97	0.60	140	0.62/0.97	0.60
Member	N	44	0.91/1.00	0.91	59	0.91/1.00	0.91	74	0.91/1.00	0.91
People	N	169	0.90/1.00	0.90	225	0.90/1.00	0.90	282	0.90/1.00	0.90
Die	V	44	0.97/0.97	0.94	59	0.97/0.99	0.96	74	0.97/0.99	0.96
Fall	V	31	0.49/0.65	0.32	41	0.35/0.69	0.24	52	0.34/0.71	0.24
Give	V	223	0.31/0.72	0.22	297	0.30/0.80	0.24	372	0.34/0.78	0.27
Include	V	86	0.72/0.97	0.70	115	0.71/0.99	0.70	144	0.70/0.99	0.69
Know	V	308	0.58/1.00	0.58	411	0.58/1.00	0.58	514	0.61/1.00	0.61
Seek	V	27	0.62/0.83	0.51	36	0.63/0.89	0.56	46	0.62/0.89	0.55
Understand	V	50	0.78/0.99	0.77	67	0.77/1.00	0.77	84	0.77/1.00	0.77
A		120.50	0.81/1.00	0.81	161.00	0.83/1.00	0.83	202.00	0.82/1.00	0.82
B		220.50	0.73/1.00	0.73	294.50	0.74/1.00	0.74	368.50	0.72/1.00	0.72
N		71.62	0.77/0.98	0.75	95.62	0.79/0.99	0.79	119.88	0.80/0.99	0.79
V		109.86	0.57/0.89	0.51	146.57	0.56/0.92	0.52	183.71	0.58/0.92	0.53
Overall		106.53	0.69/0.95	0.66	142.21	0.70/0.97	0.68	178.21	0.70/0.97	0.68

Table 24: Learning Curve in the Semcor corpus. (60%-100%).

	PoS	Occ. (10%)	Prec. (10%)	Rec. (10%)	Occ. (20%)	Prec. (20%)	Rec. (20%)	Occ. (40%)	Prec. (40%)	Rec. (40%)
age	N	49	0.65/0.98	0.64	98	0.68/0.99	0.67	196	0.69/1.00	0.69
church	N	37	0.62/0.97	0.60	74	0.66/0.99	0.65	148	0.66/1.00	0.66
head	N	86	0.71/0.98	0.70	173	0.73/1.00	0.73	346	0.76/1.00	0.76
interest	N	147	0.57/0.96	0.55	295	0.58/0.97	0.56	591	0.60/0.99	0.59
Member	N	143	0.77/1.00	0.77	286	0.78/1.00	0.78	572	0.79/1.00	0.79
Fall	V	140	0.77/0.99	0.76	281	0.77/1.00	0.77	563	0.78/1.00	0.78
Give	V	126	0.76/1.00	0.76	252	0.76/1.00	0.76	504	0.77/1.00	0.77
Know	V	144	0.41/0.94	0.39	288	0.44/0.97	0.43	576	0.45/0.97	0.44
N		92.40	0.67/0.98	0.66	185.20	0.69/0.99	0.68	370.60	0.70/1.00	0.70
V		136.67	0.65/0.98	0.63	273.67	0.65/0.99	0.65	547.67	0.66/0.99	0.66
Overall		109.00	0.66/0.98	0.64	218.38	0.67/0.99	0.66	437.00	0.68/0.99	0.68

Table 25: Learning curve in the DSO corpus (10%-40%).

	PoS	Occ. (60%)	Prec. (60%)	Rec. (60%)	Occ. (80%)	Prec. (80%)	Rec. (80%)	Occ. (100%)	Prec. (100%)	Rec. (100%)
age	N	294	0.71/1.00	0.71	392	0.73/1.00	0.73	491	0.73/1.00	0.73
church	N	222	0.69/1.00	0.69	296	0.69/1.00	0.69	370	0.71/1.00	0.71
head	N	519	0.75/1.00	0.75	692	0.77/1.00	0.77	866	0.79/1.00	0.79
interest	N	887	0.61/0.99	0.60	1183	0.62/1.00	0.62	1479	0.62/1.00	0.62
Member	N	858	0.80/1.00	0.80	1144	0.80/1.00	0.80	1430	0.79/1.00	0.79
Fall	V	844	0.79/1.00	0.79	1126	0.80/1.00	0.80	1408	0.80/1.00	0.80
Give	V	757	0.77/1.00	0.77	1009	0.77/1.00	0.77	1262	0.77/1.00	0.77
Know	V	864	0.44/0.98	0.43	1152	0.46/0.97	0.45	1441	0.46/0.98	0.45
N		556.00	0.71/1.00	0.71	741.40	0.72/1.00	0.72	927.20	0.72/1.00	0.72
V		821.67	0.66/0.99	0.66	1095.67	0.67/0.99	0.67	1370.33	0.67/0.99	0.67
Overall		655.62	0.69/1.00	0.69	874.25	0.70/1.00	0.70	1093.38	0.70/1.00	0.70

Table 26: Learning curve in the DSO corpus (60%-100%).