

Lexically Sensitive Disambiguation Techniques Research Proposal

Tanja Gaustad

Alfa-Informatica

PIONIER-Project: *Algorithms for Linguistic Processing*

University of Groningen

T.Gaustad@let.rug.nl

Contents

1	Introduction	2
2	Word Sense Disambiguation	3
2.1	Approaches to WSD	3
2.2	Attempt at Evaluation: SENSEVAL	4
2.3	Experiments	6
2.3.1	Pseudowords	6
2.3.2	Naive Bayes Classification	7
2.3.3	Varying Corpus Size	8
2.3.4	Varying Thresholds for Contextwords	9
2.3.5	Pseudowords vs. Real Ambiguous Words	10
2.4	Conclusions	15
3	PhD Project Proposal	16
3.1	Linguistic Information for WSD	17
3.1.1	Morphological Information	17
3.1.2	Syntactic Information	17
3.1.3	Semantic Information	18
3.1.4	Pragmatic Information	18
3.2	Follow-up Studies and Experiments	19
3.2.1	Unsupervised WSD	19
3.2.2	Combination of (Linguistic) Information Sources	20
3.3	Schedule	21
	References	22

1 Introduction

This paper gives an outline of the Ph.D. project “Lexically Sensitive Disambiguation Techniques”. This Ph.D. project is part of the PIONIER-Project “Algorithms for Linguistic Processing” which focuses on ambiguity and processing efficiency in NLP. The main goal of the Ph.D. project presented here is to investigate Word Sense Disambiguation (WSD) for Dutch using statistical methods in combination with different sources of linguistic information.

The proposed research questions are:

1. What kind of linguistic information is most useful for WSD?
2. How can one successfully combine statistical approaches to WSD with linguistic information?
3. How can the interplay between corpus, linguistic information sources and disambiguation proper be optimised?

In the project, the use of linguistic information for a large-scale, all (content) words WSD system for Dutch will be investigated. We will try to develop an unsupervised system which makes as much use as possible from linguistic information (such as e.g. part-of-speech tags or syntactic relations). This means that the resulting system will be unsupervised while, at the same time, exploiting as much supervision as possible. The advantage of using an unsupervised algorithm is that no annotated data is needed for the disambiguation process itself. Exploiting supervision means that the system makes use of as much (high quality) linguistic information as is available and useful.

In the first part of this proposal, an introduction to the subject of WSD is given, which includes an explanation of the possible approaches in WSD, a more detailed description of the chosen approach, as well as an elaboration of the difficulties of corpus-based WSD. Furthermore, the WSD-specific evaluation experiment SENSEVAL and related tasks are discussed. A detailed description of the experiments conducted so far concludes the first part.

The research questions posed in this Ph.D. Project are stated in the second section. There, we also list the kinds of linguistic information which can be used to aid the disambiguation process (in addition to statistical methods) and in what way they can be combined. In addition, future studies and experiments are discussed. Finally, the section contains a (preliminary) schedule of future work.

2 Word Sense Disambiguation

A major problem in natural language processing is that of lexical ambiguity, be it syntactic or semantic. A word’s *syntactic* ambiguity can be resolved by applying part-of-speech taggers which predict the syntactic category of a word in texts with high levels of accuracy (see for example (Brill, 1995) or (Brants, 2000)). The problem of resolving *semantic* ambiguity, which is generally known as word sense disambiguation (WSD), has proved to be more difficult than syntactic disambiguation.¹

Disambiguation involves two major steps: First, the possible senses for every ambiguous word have to be determined. This can either be achieved through an inventory of senses (e.g. Machine Readable Dictionaries (MRDs)), listing equivalents in a different language (bilingual dictionaries) or grouping features, categories and/or associated words. In a second step, the appropriate sense has to be assigned to ambiguous words using information about the context (linguistic and extra-linguistic) as well as external knowledge sources.

The only way to assign the meaning of a word in a particular usage is thus to examine its context. For instance, the English word *bank*—an extensively cited example of lexical ambiguity—can refer to the bank of a river or to the pecuniary institution. For this reason, a computer program analyzing the sentence “The boy leapt from the bank into the cold water” will need to decide which reading of ‘bank’ was intended, in order to be able to come up with the correct meaning for the sentence. The overall goal of word sense disambiguation systems is to attribute the correct sense(s) to words in a text.

2.1 Approaches to WSD

There are three ways to approach the problem of assigning the correct sense(s) to ambiguous words in context: a *knowledge-based approach*, which uses an explicit lexicon (MRDs, Thesauri), *corpus-based disambiguation*, where the relevant information about word senses is gathered from training on a large corpus, or, third alternative, a *hybrid approach* combining aspects of the aforementioned methodologies (see (Ide and Véronis, 1998) for a more thorough discussion).

A corpus-based approach has the advantage that text material is easily accessible. The possible means used to attribute senses to ambiguous words are then *distributional information* and *contextwords*. Distributional information about an ambiguous word is the frequency distribution of its senses.

¹See (Wilks, 2000) for arguments in favor of keeping POS-tagging and WSD separate tasks.

Contextwords are the words found to the right and/or the left of a certain word, thus collocational information.

There are two possible approaches to corpus-based WSD systems: *Supervised* and *unsupervised WSD*. Supervised approaches use a training and a testing phase. During training on a disambiguated corpus probabilistic information about contextwords as well as distributional information about the different senses of an ambiguous word are collected. In the testing phase, the sense with the highest probability computed on the basis of the training data (contextwords) is chosen. Training and evaluating such an algorithm presupposes the existence of sense-tagged corpora. Unsupervised algorithms, on the other hand, are applied to raw text material and annotated data is only needed for evaluation.

The major difficulties of a corpus-based approach are the need for manual sense-tagging and data sparseness.

So far there has not been a lot of sense-tagged material made publicly available, and even for English the corpora are still very (if not too) small. One approach to solve the problem is to manually sense-tag corpora using e.g. (Euro)WordNet hierarchies. Another, less time consuming, possibility is the application of unsupervised machine learning techniques (Schütze, 1998) to WSD (although the ‘evaluation problem’ stays the same, see sections 2.2 and 3.2.1).

The difficulty of data sparseness for WSD lies in the fact that there is a disparity in frequency among different senses of an ambiguous word. *Smoothing* is used to ensure that infrequent data or unseen data is treated properly. Class-based (Yarowsky, 1992) and similarity-based (Karov and Edelman, 1996; Karov and Edelman, 1998) models try to overcome data sparseness by generalizing over classes of words.

2.2 Attempt at Evaluation: SENSEVAL

Evaluation is an important matter within the discipline of NLP in general, and in WSD in particular. To evaluate means to compare the results of a particular system with what is seen as correct solution to the problem. In WSD, sense-tagged corpora are needed for evaluation. So far, reliable evaluation data can only be produced through hand-annotation which is very time and expertise-intensive as well as dependent on the skills of the annotator(s).²

²A measure for the quality of hand-annotated text has been established, the *Inter-Tagger Agreement* (ITA). See (Kilgarriff, 1998a) for an extensive discussion over the production of Gold Standard datasets.

Another difficulty of evaluating WSD systems with regard to each other is that different lexicons with different sense inventories are used. This means that there is no basis on which to compare the systems. Also, different additional knowledge sources might be employed by different systems which does not facilitate comparison either.

A first attempt within WSD to setup a common task for several systems in order to allow for evaluation is SENSEVAL. SENSEVAL1, held in 1998, was “the first open, community-based evaluation exercise for WSD programs” in which 18 systems participated (Kilgarriff and Rosenzweig, 2000). The setup allowed for supervised and unsupervised systems to participate, and included a coarse and fine-grained level of sense distinctions.

Several choices regarding task design, corpus and dictionary used had to be made. The task was chosen to be a *lexical* task which means that only a (small) set of ambiguous words is disambiguated. An *all-words* approach, in contrast, would mean annotating all ambiguous (content) words in a given corpus. The HECTOR lexical database (Atkins, 1993) was chosen for corpus and dictionary since this database had not been widely used in WSD before and was readily available. The results of SENSEVAL1 show the state-of-the-art for supervised (fine-grained) WSD to be 74%-78% correct. Unfortunately, no precise results on unsupervised systems are reported. It is only stated that for unsupervised systems “scores were both lower and more variable” (although of the 18 participating systems 10 were supervised and 8 were unsupervised).

Since there exist quite substantial differences in linguistic resources between English and other languages, a specific competition for Romance languages called ROMANSEVAL was started in parallel to SENSEVAL1 (See (Segond, 2000) and (Calzolari and Corazzari, 2000) for the setup and results for French and Italian, respectively).

After the success of SENSEVAL1, SENSEVAL2 was started in 2000, broadening the task to different languages, to a choice between lexical or all-words disambiguation, as well as to a more flexible framework³. Its results will be evaluated at the ACL 2001, in Toulouse.⁴

³More information about task descriptions, participating languages, etc. is available from <http://www.sle.sharp.co.uk/senseval2/>.

⁴See http://www.irit.fr/ACTIVITES/EQ_ILPL/ac1Web/ac12001.html: Workshops.

2.3 Experiments

In this section, we will report on three experiments that have been carried out during the last year. They all used a supervised WSD algorithm (see section 2.3.2) which was trained on either the European Corpus Initiative (ECI) corpus of Dutch⁵ or on the SENSEVAL1-corpus⁶. Since there is no disambiguated material available for the Dutch ECI corpus (which means that evaluation of results is not possible), we artificially created such data using pseudowords.

2.3.1 Pseudowords

The technique of pseudowords consists of introducing a form of artificial ambiguity in (untagged) corpora. First of all, two or more words, *sensewords*, are chosen. Training then takes place on the disambiguated corpus, collecting probabilities for the chosen sensewords. For testing, all occurrences of the sensewords are replaced by a non-existing word, a *pseudoword*. The goal is then to recover the correct senseword for every pseudoword introduced in the corpus.

Suppose we chose the sensewords ‘aantal’ and ‘tijd’ and combined them to form the pseudoword ‘aantijd’. The original sentences (1) and (2)—which are used in training as well as in evaluation—will then become test sentences (3) and (4).

- (1) Hun aantal groeit en volgens justitie lijkt aan die groei geen einde te komen.
- (2) Tot die tijd blijven de stellingen betrokken.
- (3) Hun *aantijd* groeit en volgens justitie lijkt aan die groei geen einde te komen.
- (4) Tot die *aantijd* blijven de stellingen betrokken.

Gale, Church, and Yarowsky (1992b) used pseudowords to overcome the “testing material bottleneck”, as well as Schütze (1992) and Schütze (1998), who tried to escape the need for hand-labeling using artificial ambiguous words for evaluation purposes.

⁵The ECI is a digitally available multilingual corpus distributed by ELSNET which contains material on a number of European languages, among others Dutch. See <http://www.elsnet.org/eci.html> for a complete listing of available languages and ordering information.

⁶Publicly available at <http://www.itri.brighton.ac.uk/events/senseval11/ARCHIVE/resources.html>.

2.3.2 Naive Bayes Classification

In the case of the preliminary experiments reported here, we chose to work with a *naive Bayes classifier* (Duda and Hart, 1973) because it is easy to implement, performs relatively well, is rather fast and is used fairly often.

In addition to that, a Bayes classifier uses only distributional information and contextwords to compute probabilities which corresponds to only using information which is available from the corpus itself without the need of any additional material, such as a dictionary or the like. The contextwords are assumed to be independent of position and of each other—they constitute a *bag of words*—which corresponds to the Bayes independence assumption.

First, the disambiguation algorithm is trained on part of the unambiguous corpus, attributing probabilities to the contextwords found to the right and the left of the senseword(s) for various context window sizes. This is done using Bayes rule

$$P(s_k|c) = \frac{P(c|s_k)}{P(c)}P(s_k)$$

where s_k is sense k of ambiguous word w in context $c = \{c_1, \dots, c_n\}$, the contextwords within the specified context window. “Training” as used here amounts to counting which senses are used most often in a given context.

Testing takes place on the ambiguous text where the algorithm selects the most probable senseword for each pseudoword according to Bayes decision rule

$$\text{Decide } s' \text{ if } P(s'|c) > P(s_k|c) \text{ for } s_k \neq s'$$

Finally, the computed sensewords are compared to the original sensewords in the disambiguated corpus and the percentage of correctly disambiguated instances of pseudowords is calculated.

Despite its relatively ‘naive’ approach, the Naive Bayes classifier performs relatively well, especially in comparison with other, more sophisticated approaches (Mooney, 1996; Escudero, Màrquez, and Rigau, 2000).

Smoothing As has been pointed out in section 2.1, sparse data is a problem in corpus-based WSD. If a contextword has not been seen with a particular sense of an ambiguous word in the training data, the probability $P(v_j)$ of contextword v_j in the context of all senses s_k of ambiguous word w will be 0. This means that no choice can be made using the naive Bayes Classification algorithm explained above.

In such a case, smoothing techniques are applied. In the experiments described in sections 2.3.5, 2.3.3 and 2.3.4, a fixed penalty of $-\log 0.01$ (corresponding to a probability of $p = 0.01$) has been used. Possible extensions would be to use more sophisticated smoothing techniques, e.g. Good-Turing.

Pseudow.	Senseword 1		Senseword 2		Senseword 3		Baseline
	word	frequ.	word	frequ.	word	frequ.	
aantijd	aantal	1995	tijd	1747			53.31%
lagem	land	2991	gemeente	1018			74.60%
nedmin	nederland	2675	minister	3155			54.11%
prespol	president	2356	politie	1568			60.04%
neduir	nederland	2675	duitsland	719	irak	2818	45.36%
plonbe	plan	1059	onderwijs	960	beleid	908	36.18%

Table 1: Overview Pseudowords

2.3.3 Varying Corpus Size

In a first experiment, we looked at the changes of performance in the classification algorithm used depending on corpus size. When working with statistical methods, changes in corpus size/training instances are expected to be reflected in changes of performance (Langley, Iba, and Thompson, 1992). The usual assumption is that the bigger the corpus the better the performance.

Settings: Corpus and Pseudowords The corpus used in this experiment was the ECI Corpus of Dutch which contains approximately 3 Million words of raw text. The corpus includes transcripts of radio programs, newspaper articles, magazine issues and some technical texts.

Choosing high frequency nouns, six pseudowords were created, four of which consist of two sensewords and two of which consist of 3 sensewords. Table 1 gives an overview over the sensewords chosen as well as their frequency and the frequency baseline⁷.

Underlying Assumptions In the experiment described, we departed from two underlying assumptions: Topic coherence and ‘All information’. The idea of topic (or discourse) coherence states that words usually keep the same sense within a paragraph or document (Gale, Church, and Yarowsky, 1992a; Yarowsky, 1993).⁸ The size of the context window used was thus restrained to paragraphs, which means that if the window size on either side

⁷There are two possibilities to calculate the baseline for WSD Systems: the *random baseline*, which chooses a possible sense at random, or the *frequency baseline*, where the most frequent sense is always chosen. Usually the frequency baseline lies higher than the random baseline, which is why it is a more representative lower bound.

⁸Krovetz (1998) has shown that this is only (partially) true for homonymous senses, but is not the case for polysemous words.

Pseudoword	Baseline	0.5M Words	1.5M Words	3M Words
aantijd	53.31	80.32	84.08	84.97 (+31.66)
lagem	74.60	78.54	80.08	82.65 (+08.05)
nedmin	54.11	84.45	83.18	85.02 (+30.91)
prespol	60.04	73.99	79.09	83.21 (+23.17)
neduir	45.36	65.83	66.38	70.83 (+25.47)
plonbe	36.18	58.32	67.25	67.70 (+31.52)

Table 2: Results with varying corpus size (in %)

was bigger than the paragraph boundary, everything beyond that boundary was not taken into account.⁹

Furthermore, no stoplist was used in the reported experiments. One of the working hypotheses was to test whether taking into consideration all available context information including function words could produce good results. In the case of nouns with different articles, for instance, working with a stoplist would definitely be counter-productive. Also, for words with a very different syntactic distribution, like ‘nederland’ and ‘minister’, prepositions and articles are good indicators for a certain ‘sense’.

Results and Evaluation In the reported experiment, results were ten times crossvalidated. The context window was restricted to 3 words to the left and the right of the pseudoword. We take a similar approach to (Chodorow, Leacock, and Miller, 2000) choosing a fixed context window size of ± 3 . Similar results can be observed when different context sizes are used.

The results obtained (see table 2) clearly show that more training instances do help improve the performance of the naive Bayes classification algorithm used. The overall performance of the algorithm is quite good, especially considering the fact that the results are solely based on statistical information.

2.3.4 Varying Thresholds for Contextwords

In a different experiment, we looked at the use of contextwords. The main idea was to only use contextwords of a certain informative value (expressed through placing a threshold on the probability of each contextword) and to find the cutoff at which the amount of data still used in the disambiguation process and the informative value converge. The thresholds repre-

⁹There was a big variation in paragraph lengths (1-15 sentences). It is not quite clear yet what sort of noise is introduced through this fact.

Pseudoword	Baseline	all	0.6	0.7	0.8	0.9	1
aantijd	53.31	84.97	85.03	84.97	79.83	76.64	72.11
lagem	74.60	82.65	82.65	82.62	82.88	81.83	81.60
nedmin	54.11	85.02	85.09	84.25	82.85	71.66	69.49
prespol	60.04	83.21	83.43	81.97	81.15	79.23	78.63

Table 3: Results with varying thresholds (in %), optimal performance per row in bold

sent how well a particular contextword helps to disambiguate an ambiguous word/pseudoword. A threshold of 1.0 means that a contextword is only used for disambiguation if the probability of contextword v_j given sense k of ambiguous word w is 1 ($p(v_j|s_k) = 1$).

The corpus and overall settings were the same as in the experiment reported in section 2.3.3. Only the four pseudowords consisting of two senses were used.

Results and Evaluation As the results in table 3 show there is no clear cutoff value at which the performance of the algorithm improves for all pseudowords. A tendency can be observed that using a threshold of 0.6 (which means that all contextwords are used *except* those which are (almost) equally likely to occur with both senses of a given ambiguous word) works best.

2.3.5 Pseudowords vs. Real Ambiguous Words

In the last experiments reported, we investigated whether disambiguating pseudowords is comparable to the task of disambiguating real ambiguous words and we reached the conclusion that these two tasks are *not* identical (Gaustad, 2001).

Outline of the Problem The idea to compare the task of disambiguating real ambiguous words to disambiguating artificially ambiguous words arose from our work on supervised WSD for Dutch. Since there are no sense-tagged corpora available for Dutch, another means of testing algorithms has to be used. An obvious solution is the use of pseudowords: they are easily created, only raw text material is needed and any supervised algorithm can be tested. The one question that remained unanswered was whether using pseudowords would yield results comparable to real WSD and whether the seemingly ‘easy way out’ could really be seen as equivalent to the disambiguation of real ambiguous words.

Unfortunately, there has not been a lot of work on pseudowords and, to the best of our knowledge, no work at all on their usefulness in testing word sense disambiguation systems. The major problem involved in this comparison is to find a valid setting for a comparison: the elements to be compared—pseudowords and real ambiguous words—are too different from each other to be compared directly. Schütze (1998) explains it in the following way: “[The better performance on pseudowords] can be explained by the fact that pseudowords have two focused senses—the two word pairs they are composed of.” Real ambiguous words, on the other hand, consist of subsenses that are difficult to identify for humans as well as for computers.

Way of Proceeding A direct comparison of the task of WSD and the task of disambiguating pseudowords is not possible. The only way to compare these two tasks is to *indirectly* compare their results on the same corpus, using the same algorithm and general settings.

The comparison does have its limitations: Although we use the same settings for both tasks, the difference between them lies in the actual words (or pseudowords) to be disambiguated. There is no measure to express their differences or similarities. This is precisely why there is no possibility of a direct comparison.

We decided to proceed in two steps. First, real ambiguous words were chosen from the SENSEVAL1 corpus making use of the dictionary entries as well as the training and testing material provided. Only nouns which were not ambiguous regarding part of speech and for which there was training data were taken into account.

In a second step, we chose the sensewords of a pseudoword according to the frequency distribution of the senses of the real ambiguous words that were tested. Among the possible sensewords that exhibited the same frequency distributions as the real ambiguous words and which fulfilled the constraint of having approximately the same baseline, an arbitrary selection was made.

If the results of this second task are significantly different from the results of the first task on the same corpus, this will show that the results involving pseudowords depend entirely on the choice of sensewords. This means that the disambiguation of pseudowords is *not* identical to the real WSD task. Note that if one does not have access to sense-tagged corpora, no information about the distribution of the senses of real ambiguous words is available, which means that it is not really a comparable setup!

Settings: Corpus and Ambiguous Words/Pseudowords The corpus used in this experiment are the English SENSEVAL1 resources. The advantage of using this material is that it is (lexically) sense-tagged for a number of real ambiguous words which means that the evaluation data for real ambiguous words is at hand. Furthermore, there have been numerous publications on the construction of the material, on choices made regarding annotation, on inter-annotator agreement, etc. ((Kilgarriff, 1998b; Kilgarriff and Rosenzweig, 2000), and see also section 2.2), which allow for a thorough understanding of the real world disambiguation task. This is an important precondition to being able to extensively compare this task to nearly the same task using pseudowords.

The perhaps most important factor in this comparison is the choice of elements of comparison, in this case the ambiguous words and the sensewords chosen to constitute the different pseudowords.

The choice of ambiguous words depended, on the one hand, on the available SENSEVAL1 material (evaluation data). On the other hand, we only selected nouns which were not ambiguous in part-of-speech.¹⁰ No stemming was used. The ambiguous words and their senses¹¹ chosen for the experiments can be seen in table 4¹².

The main criteria for choosing the sensewords constituting the pseudowords were their frequency in the corpus as well as their part of speech. For the comparison with each ambiguous word, five pseudowords were made up. The distribution of these pseudowords' sensewords was chosen to be as similar as possible to the distribution of the different senses of the ambiguous words. An overview (including frequencies) of the pseudowords and the corresponding ambiguous word is given in table 4.

Results and Evaluation The results in table 5 clearly show that the performance of the naive Bayes classification algorithm used is significantly better on pseudowords than on real ambiguous words. A possible reason for this is the relatedness of sense distinctions in real ambiguous words whereas the sensewords that constitute pseudowords have two very clearly distinct senses.

¹⁰A number of ambiguous words in the SENSEVAL1 material had to be simultaneously part-of-speech and lexically disambiguated, e.g. *bet*, *giant*, *promise*. There were also cases with no training material provided (*disability*, *hurdle*, *rabbit*, *steering*) which were not taken into account given that we worked with a supervised algorithm.

¹¹The senses were taken from the SENSEVAL1 dictionary entries. Only the coarse-grained distinctions were taken into account.

¹²Since the sense *hairsh* does not occur in the testing data, we decided to only consider two senses for *shirt* and, consequently, for the pseudowords.

	Amb./Ps.word	Senses/S.words	Freq. train	Freq. test	Baseline
Ambig. word	accident	crash	1058	248	92.88%
		chance	178	19	
Pseudowords	timwe	time	722	306	91.90%
		weekend	73	27	
	yeatra	year	708	307	92.47%
		traffic	86	25	
	peolang	people	673	268	92.10%
		language	54	23	
	woan	world	422	187	92.12%
		animal	39	16	
	goveq	government	396	184	92.35%
		equipment	31	15	
Ambig. word	behavior	social	969	267	95.70%
		of thing	29	12	
Pseudowords	peostan	people	673	268	93.40%
		standards	41	19	
	tima	time	722	306	95.33%
		machine	49	15	
	yeagro	year	708	307	95.34%
		growth	58	15	
	wodat	world	422	187	94.92%
		data	36	10	
	gopay	government	396	181	95.26%
		payment	30	9	
Ambig. word	excess	aglut	103	108	58.06%
		of or after poss	65	67	
		surplus	10	9	
		too much	73	2	
Pseudowords	womuconba	world	422	187	58.62%
		music	231	97	
		concert	43	16	
		battle	42	19	
	gopoemch	government	396	184	57.64%
		police	218	98	
		empire	37	16	
		champion	45	19	
	dacipapro	day	373	161	57.71%
		city	211	83	
		palace	37	16	
		protection	45	19	
	pemanora	people	673	268	58.64%
		man	377	154	
		noise	33	16	
		railway	33	19	
	heterite	head	349	150	58.37%
		team	162	72	
		river	42	16	
		technology	34	19	
Ambig. word	shirt	t-shirt	132	73	57.06%
		garment	336	105	
Pseudowords	schoclu	school	178	87	59.02%
		club	140	72	
	mastre	market	190	89	58.55%
		street	158	63	
	cimon	city	211	83	58.04%
		month	130	60	
	coufam	country	201	91	57.96%
		family	117	66	
	wogia	women	189	91	58.33%
		giants	140	65	

Table 4: Overview ambiguous words and corresponding pseudowords

	Basel.	Results	Difference
accident	92.88	84.45	- 8.43
timwe	91.90	91.56	- 0.34
yeatra	92.47	91.77	- 0.70
peolang	93.10	91.88	- 0.59
woan	92.12	93.44	+ 0.97
goveq	92.35	91.33	- 1.14
<i>mean</i>			- 0.40 [\pm 0.89]
behaviour	95.70	84.95	- 10.75
peostan	93.40	92.99	- 0.41
tima	95.33	95.64	+ 0.31
yeagro	95.34	94.04	- 1.30
wodat	94.92	93.79	- 1.13
gopay	95.26	96.36	+ 1.10
<i>mean</i>			- 0.29 [\pm 1.24]
excess	58.06	50.35	- 7.71
womuconba	58.62	71.86	+13.24
gopoemch	57.64	72.92	+15.28
dacipapro	57.71	73.98	+16.27
pemanora	58.64	73.00	+14.36
heterite	58.37	74.39	+16.02
<i>mean</i>			+15.03 [\pm 1.55]
shirt	58.98	57.50	- 1.48
schoclu	59.02	72.79	+13.77
mastre	58.55	74.83	+16.28
cimon	58.04	78.69	+20.65
coufam	57.96	63.91	+ 5.95
wogia	58.33	72.22	+13.89
<i>mean</i>			+14.1 [\pm 6.6]

Table 5: Pseudowords vs. Real Ambiguous Words: Results (in %)

A possible explanation for the fact that the performance on real ambiguous words is considerably bad—it constantly fails to reach the baseline—is that there is not enough training data. Note that the baseline of most ambiguous nouns in the SENSEVAL corpus is relatively high which means that one sense accounts for most occurrences of the ambiguous word. This makes the disambiguation task comparatively harder and might be a possible explanation for the bad performance on real ambiguous words.

We conclude from the results that the task of disambiguating pseudowords is comparable only in a limited way to the task of disambiguating real ambiguous words. The results on pseudowords will usually be better which might lead to false assumptions about the performance of a given algorithm on the real problem.

The results obtained from disambiguating artificial ambiguous words differ greatly from the results of real ambiguous words. This indicates that pseudowords cannot be taken as a substitute for testing with real ambiguous words.

2.4 Conclusions

Testing of WSD algorithms is very difficult without evaluation data. The assumption that artificially created ambiguous words are a good substitute for real ambiguous words is *not* valid, as has been shown by the experiment reported in section 2.3.5. Thus the initial problem—wanting to test algorithms for languages without sense tagged corpora—remains. The most promising option is the use of unsupervised techniques.

3 PhD Project Proposal

Based on the research review and the conducted experiments, the following questions concerning WSD are proposed.

1. What kind of linguistic information is most useful for WSD?
2. How can one successfully combine statistical approaches to WSD with linguistic information?
3. How can the interplay between corpus, linguistic information sources and disambiguation proper be optimised?

The future research plans are to implement a WSD algorithm which makes use of different types of linguistic information (e.g. part-of-speech, dependency relations, selectional restrictions) in combination with statistical methods.

Statistical approaches have proved to be successful and rather efficient, but intuitively one would think that the addition of linguistic information should lead to increases in performance¹³. In WSD, it has not yet been systematically investigated whether this is the case.

A major question that will be investigated is what kind of linguistic information is most useful for word sense disambiguation (see section 3.1). Also, different linguistic information might be useful depending on the syntactic category and other characteristics of the target word, as different disambiguation strategies might be needed for different groups or classes of target words.

The word sense disambiguator we are aiming to develop should be an unsupervised system which exploits as much supervision as possible. Unsupervised means that no annotated data is needed for the disambiguation process itself (i.e. there is no training phase); exploiting supervision means that the system makes use of as much (high quality) linguistic information as is available and useful. Especially in the context of the PIONIER-project of which this Ph.D. project is part, it seems reasonable to include information made available by the Alpino grammar and parser. This ‘collaboration’ also facilitates the integration of the developed WSD system into the final NLP tool for Dutch.

¹³For example, Wilks and Stevenson (1997) state that: “Our intuition is that word sense disambiguation can be most effectively carried out combining [different, orthogonal] knowledge sources.”

3.1 Linguistic Information for WSD

There are different sources of knowledge which can be used in WSD. As has been said above, the use of linguistic knowledge for a WSD system has not been thoroughly investigated yet, although it might lead to significant increases in performance.¹⁴ Types of linguistic knowledge to be possibly used in WSD are detailed in the following sections. The linguistic information will be used in addition to statistical information (e.g. bigrams, see (Pedersen, 2001)) and a statistics-based algorithm (see section 2.3.2).

3.1.1 Morphological Information

Lemmatising Lemmatising is the process of stripping words from morphological information and only keeping stems or root forms. Generalizing over different morphological realisations of words might be an advantage when working with verbs. For nouns (and probably also adjectives and adverbs) it might be disadvantageous not to include morphological information. Important facts about e.g. senses which are only applicable in case of plural might be lost. There is also the danger of overgeneralisation which might lead to not being able to distinguish between different words anymore because they are shortened to the same stem.

3.1.2 Syntactic Information

Part-of-speech (POS) POS tags contain syntactic information at word level. They can be used in the following way. Senses that do not match the POS of the ambiguous word in context can be eliminated. This reduces the number of senses that have to be considered in the disambiguation process.

Subcategorization Frames Subcategorization frames provide valency information for different categories of words (nouns, verb, adjectives). They are typically associated with *selectional* restrictions, i.e. restrictions on the meaning of grammatical complements, and thus they could be useful e.g. in determining which senses of nouns are appropriate as complements of a certain sense of a verb given a particular syntactic construction, and vice versa.

¹⁴There are attempts to integrate linguistic knowledge into WSD systems (see e.g. Ng and Lee's (1996) exemplar-based approach which uses POS, stemming and verb-object syntactic relations), but, to the best of our knowledge, no systematic research has been conducted yet.

Parsing Parsing assigns syntactic structure to word strings (at sentence level). The main contribution of parsing is to give information about grammatical relations. These can then be used as input for e.g. selectional restrictions.

Dependency Relations Dependency relations hold between constituents in a sentence which are dependent on each other. Since dependency structures do not necessarily reflect surface (syntactic) structure, they are a valuable source of information when full parses are unavailable. Being able to identify dependencies between constituents could be used to determine which words might contain more important disambiguation clues than others.

3.1.3 Semantic Information

WordNet WordNet is a lexicon where the similarity between words is represented hierarchically. Lexical information is organised by semantic properties. In addition to semantic information, relations, such as for instance hyponymy or hyperonymy, between words and concepts are also defined. Different approaches to WSD have used information from WordNet (Leacock, Chodorow, and Miller, 1998; Hawkins, 1999). An asset is that WordNet provides semantic information attached to lexical items. One of the difficulties with the architecture of this lexicon, on the other hand, is the fact that different POS are contained in separate hierarchies. Also, related words cannot be clustered easily since they might be classified far away from each other (referred to as ‘tennis problem’, see section 3.5 in (Hawkins, 1999)).

Selectional Restrictions Selectional restrictions are restrictions on the meaning of grammatical complements. Resnik (1993) used selectional restrictions to resolve syntactic ambiguity. His hypothesis is that many lexical relationships reflect underlying conceptual relationships and that statistical disambiguation strategies should take those into account. Prior to using selectional restrictions the grammatical links between words must already have been identified. A drawback of this approach is that selectional restrictions are only useful for the resolution of broad sense ambiguity since fine-grained senses often belong, or are used with, the same semantic class.

3.1.4 Pragmatic Information

Topical information A thesaurus usually contains information about subject areas, often called pragmatic codes. Using these codes can help to optimise the choice of senses: the goal is to achieve the greatest overlap of

pragmatic codes within a text with paragraphs as basic units (Wilks and Stevenson, 1997; Wilks and Stevenson, 1998). Yarowsky (1992), for instance, uses Roget’s thesaurus to identify salient contextwords that appear with an ambiguous word belonging to a certain category. These can then be used to resolve ambiguity since they provide evidence for a particular category.

3.2 Follow-up Studies and Experiments

The experiments described in the first part of this research proposal have shown that the use of pseudowords to investigate WSD is not a viable option. The production of sense-tagged training material for supervised algorithms is very time- and expertise-intensive.

An interesting option is to work with an unsupervised algorithm. In addition to implementing such an algorithm, we will test the use of linguistic information for a WSD system in general, and for Dutch in particular. The advantage of using an unsupervised algorithm is that no annotated data is needed for the disambiguation process itself. Linguistic information should ideally help improve the performance of such a WSD system.

3.2.1 Unsupervised WSD

Regarding the choice between a supervised and an unsupervised system, *evaluation* has to be considered. In both cases, evaluation is difficult: sense-tagged material has to be available which (so far) can only reliably be produced through hand-annotation. In the case of an unsupervised algorithm the advantage is that no (additional) annotated material is needed for the actual disambiguation.

“While it is worthwhile to utilize annotated data when it is available, one might argue that the future success of learning for natural language systems cannot depend on a paradigm requiring that large, annotated data sets be created for each new problem or genre. Thus, long-term progress in NLP is likely to be dependent on the use of unsupervised or weakly supervised learning techniques, which do not require large annotated data sets.” (Kehler and Stolcke, 1999)

Schütze (1998) uses a vector space model for unsupervised WSD. He applies clustering to WSD using semantic similarity. Senses are interpreted as clusters of similar contexts of the ambiguous words. Words, contexts, and senses are represented in a high-dimensional space in which closeness

corresponds to semantic similarity. The algorithm is fully unsupervised in training and application: senses are induced from an unlabeled corpus and no other external knowledge sources are used. A possible extension to this approach would be to use a given sense inventory as a starting point for the algorithm.

Another approach which we would like to experiment with is bootstrapping. Yarowsky (1995) developed a method which uses a decision list learner for disambiguation in connection with a human seed collection. Decision list learners generate a set of conjunctive rules. These rules are ordered according to some heuristic measure and stored as a list, which is subsequently used in classifying new examples. A new example is classified by the first rule that matches it in the list. Yarowsky is bootstrapping sense-tags from a small annotated corpus using a seed collection which is then automatically increased with every iteration. Several different possibilities how to make this ‘semi-supervised’ bootstrapping method more unsupervised are mentioned.

3.2.2 Combination of (Linguistic) Information Sources

The main goal of our future research is to systematically investigate which sources of linguistic knowledge work best for WSD and in what combination. An important basis of all future work will be to determine which sources of information can (possibly) be helpful in the disambiguation process.

In a first step, we will use available tools to separately test (most) sources of linguistic information which have been identified in section 3.1 on their value for disambiguation. Considering availability, these will be lemmatising, POS, subcategorization frames, parsing, dependency relations, as well as EuroWordNet.

Once the use of linguistic knowledge for improving the performance of a WSD System has been established and detailed for different groups of words, the combination of the acquired information has to be investigated. There are essentially two different approaches to the combination of linguistic knowledge: The information can either be combined in a hierarchical way, starting at sub-word level with morphological information, going on to word and then sentence level syntactic information, from there to semantic information and finally to pragmatic information—a classical but probably not very efficient nor optimised way of combining knowledge.

Another possibility is to determine the importance of the different sources of information for the disambiguation process and weigh them accordingly. The scores can then be used to decide on the order of combination. The choice of algorithm will also have an influence on the research on the combination of linguistic information.

3.3 Schedule

During the period from April to October 2001, I will interrupt my Ph.D. Project for 6 months to collaborate in the KOP-Project on email classification, a joint venture between the University of Groningen and BCN Costumer Care, Groningen. This will give me the opportunity to work in a related field where similar methods are applied to document/text classification and to get a deeper insight into alternative approaches to what we have been doing so far. I hope to profit from this experience in two ways: First to get experience in working together with a company outside the university, working on a real-world application, and second, be able to try out certain algorithms that might be of interest for our Ph.D. project on Word Sense Disambiguation.

In October 2001, I will then start with the follow-up studies and experiments described above. See the schedule below (table 6) for details.

April 2001	KOP Project on Email classification	6 months
October 2001	Unsupervised WSD -Replicate bootstrapping methods with different algorithm(s) and seed instantiation methods -Implement an unsupervised learning algorithm for WSD	6 months
	Linguistic Information -Investigate individual sources of linguistic knowledge -Investigate the possible modes of combination of linguistic knowledge	6 months
October 2002	Corpus-Based Evaluation -Produce hand-annotated evaluation data -Evaluate the implemented system	4 months
February 2003	Variations of model Specify, implement and evaluate variations of the original model	4 months
June 2003	Assessment -Determine success of experiments to date -Start writing thesis	10 months
April 2004	Thesis ready	
October 2004	Thesis Defense	

Table 6: Overview Schedule

References

- Atkins, Sue. 1993. Tools for computer-aided corpus lexicography: The Hector project. *Acta Linguistica Hungarica*, 41:5–72.
- Brants, Thorsten. 2000. TnT—a statistical part-of-speech tagger. In *Proceedings of the 6th Applied Natural Language Processing Conference ANLP-2000*, Seattle, WA, April 29 -May 3.
- Brill, Eric. 1995. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics*, 21(4):543–565.
- Calzolari, Nicoletta and Ornella Corazzari. 2000. Senseval/romanseval: The framework for italian. *Computers and the humanities*, 34(1-2):61–78.
- Chodorow, Martin, Claudia Leacock, and George Miller. 2000. A topical/local classifier for word sense identification. *Computers and the humanities*, 34(1-2):115–120.
- Duda, R. O. and P. E. Hart. 1973. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York.
- Escudero, Gerard, Lluís Màrquez, and German Rigau. 2000. A comparison between supervised learning algorithms for word sense disambiguation. In *Proceedings of the 14th European Conference on Artificial Intelligence, ECAI-2000*, Berlin.
- Gale, Bill, Kenneth Church, and David Yarowsky. 1992a. One sense per discourse. In *Proceedings of the ARPA Workshop on Speech and Natural Language Processing*, pages 233–237.
- Gale, Bill, Kenneth Church, and David Yarowsky. 1992b. Work on statistical methods for word sense disambiguation. In *AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, pages 54–60, Cambridge, MA.
- Gaustad, Tanja. 2001. Statistical corpus-based word sense disambiguation: Pseudowords vs. real ambiguous words. Submitted to the ACL 2001 Student Research Workshop, to be held July 6-11, 2001, Toulouse, France.
- Hawkins, Paul Martin. 1999. *DURHAM: A Word Sense Disambiguation System*. Ph.D. thesis, Laboratory for Natural Language Engineering, Department of Computer Science, University of Durham.

- Ide, Nancy and Jean Véronis. 1998. Introduction to the special issue on word sense disambiguation: The state of the art. *Computational Linguistics*, 24(1):1–40.
- Karov, Yael and Shimon Edelman. 1996. Learning similarity-based word sense disambiguation from sparse data. In *Proceedings of the 4th Workshop on very large corpora*, Copenhagen, August 4.
- Karov, Yael and Shimon Edelman. 1998. Similarity-based word sense disambiguation. *Computational Linguistics*, 24(1):41–59.
- Kehler, Andrew and Andreas Stolcke. 1999. Preface. In *Proceedings of the Workshop “Unsupervised learning in natural language processing”*, Maryland.
- Kilgarriff, Adam. 1998a. Gold standard datasets for evaluating word sense disambiguation programs. *Computer Speech and Language, Special Issue on Evaluation*, 12(4):453–472.
- Kilgarriff, Adam. 1998b. SENSEVAL: An exercise in evaluating word sense disambiguation programs. In *Proceedings LREC*, pages 581–588, Granada, May.
- Kilgarriff, Adam and Joseph Rosenzweig. 2000. Framework and results for English SENSEVAL. *Computers and the humanities*, 34(1-2):15–48.
- Krovetz, Robert. 1998. More than one sense per discourse. In *Proceedings of the ACL SIGLEX Workshop*.
- Langley, Pat, Wayne Iba, and Kevin Thompson. 1992. An analysis of bayesian classifiers. In *Proceedings of the 10th National Conference on Artificial Intelligence (AAAI-92)*, San Jose.
- Leacock, Claudia, Martin Chodorow, and George A. Miller. 1998. Using corpus statistics and wordNet relations for sense identification. *Computational Linguistics*, 24(1):147–165.
- Mooney, Raymond. 1996. Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 82–91, University of Pennsylvania.
- Ng, Hwee Tou and Hian Beng Lee. 1996. Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In

- Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL-96)*, pages 40–47, Santa Cruz, CA, June 24-27.
- Pedersen, Ted. 2001. A decision tree of bigrams is an accurate predictor of word sense. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-01)*, Pittsburgh, PA, June 2-7.
- Resnik, Philip. 1993. *Selection and Information: A Class-based Approach to Lexical Relationships*. Ph.D. thesis, Institute for Research in Cognitive Science IRCS, University of Pennsylvania.
- Schütze, Hinrich. 1992. Context space. In *AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, pages 113–120, Cambridge, MA.
- Schütze, Hinrich. 1998. Automatic word sense disambiguation. *Computational Linguistics*, 24(1):97–123.
- Segond, Frédérique. 2000. Framework and results for french. *Computers and the humanities*, 34(1-2):49–60.
- Wilks, Yorick. 2000. Is word sense disambiguation just one more nlp task? *Computers and the humanities, Special Issue SENSEVAL*, 34(1-2):235–243.
- Wilks, Yorick and Mark Stevenson. 1997. Combining independent knowledge sources for word sense disambiguation. In *Proceedings of the Conference Recent Advances in Natural Language Processing*, pages 1–7, Tzigrav Chark, Bulgaria.
- Wilks, Yorick and Mark Stevenson. 1998. Word sense disambiguation using optimised combinations of knowledge sources. In *Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual meeting of the Association for Computational Linguistics (COLING-ACL '98)*, pages 1398–1402, Montreal.
- Yarowsky, David. 1992. Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proceedings of the 15th International Conference on Computational Linguistics, COLING-92*, pages 454–460, Nantes.
- Yarowsky, David. 1993. One sense per collocation. In *Proceedings ARPA Human Language Technology Workshop*, Princeton.

Yarowsky, David. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, Cambridge, MA, June.