SENSEVAL-2: Overview

Philip Edmonds
Sharp Laboratories of Europe
Oxford Science Park
Oxford OX4 4GB, UK
phil@sharp.co.uk

Scott Cotton
Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104, USA
cotton@linc.cis.upenn.edu

Abstract

SENSEVAL-2: The Second International Workshop on Evaluating Word Sense Disambiguation Systems was held on July 5-6, 2001. This paper gives an overview of SENSEVAL-2, discussing the evaluation exercise, the tasks, the scoring system, and the results. It ends with some recommendations for future evaluation exercises.

1 Introduction

Word sense disambiguation (WSD) is the problem of automatically deciding which sense a word has in any particular context. The success of any project in WSD is clearly tied to the evaluation of WSD systems. **SENSEVAL** was started in 1997, under the auspices of ACL-SIGLEX, to bring together researchers to discuss and solve the WSD-evaluation problem. Its aim is to evaluate the strengths and weaknesses of WSD algorithms and systems with respect to different words, different varieties of language, and different languages.

SENSEVAL is independent from other evaluation programs in the language technology community, such as TREC and MUC. Unlike these programs, SENSEVAL is a 'freelance' program is run entirely by volunteers. We'd like to remind everyone that while SENSEVAL takes the guise of a competition, its main function is not to determine a winner but to explore the scientific aspects of word sense disambiguation.

SENSEVAL held its first evaluation exercise in the summer of 1998, culminating in a workshop at Herstmonceux Castle, England on September 2–4 (Kilgarriff and Palmer 2000). Following the success of the first workshop, SENSEVAL-2, supported by EURALEX, ELSNET, EPSRC, and ELRA, was organized in 2000–2001. The Second International Workshop on Evaluating Word Sense Disambiguation Systems was held in conjunction with ACL-2001 on July 5–6, 2001 in Toulouse.

This paper gives an overview of SENSEVAL-2, discussing the evaluation exercise, the tasks, the scoring system, and the results. It ends with some recommendations for future evaluation exercises.

2 Tasks and participants

A main goal of SENSEVAL-2 was to encourage new languages to participate. We were successful: SENSEVAL-2 evaluated WSD systems on three types of task on 12 languages as follows:

All-words	Czech, Dutch, English, Estonian
Lexical sample	Basque, English, Italian, Japanese, Korean, Spanish, Swedish
Translation	Japanese

In the **all-words** task, systems must tag almost all of the content words in a sample of running text. In the **lexical sample** task, we first carefully select a sample of words from the lexicon; systems must then tag several instances of the sample words in short extracts of text. The **translation** task (Japanese only) is a lexical sample task in which word sense is defined according to translation distinction. Task design is discussed in section 3 below.

93 systems were submitted from 34 different research teams. Table 1 gives a breakdown of the number of submissions and teams who participated in each task. Note that some teams submitted multiple systems to the same task, and some submitted systems to multiple tasks.

Several tasks had no submissions: the Chinese and Danish tasks could not find enough time to complete the data in time for the exercise, and the available Dutch data was misplaced in the process of making it public. The Dutch data is available, and the Chinese and Danish data will be prepared in due course.

Language	Task	No. of submissions	No. of teams
Chinese	LS	0	0
Danish	LS	0	0
Dutch	AW	0	0
Czech	AW	1	1
Basque	LS	3	2
Estonian	AW	2	2
Italian	LS	2	2
Korean	LS	2	2
Spanish	LS	12	5
Swedish	LS	8	5
Japanese	LS	7	3
Japanese	TL	9	8
English	AW	21	12
English	LS	26	15
Total		93	57

Table 1 Submissions to SENSEVAL-2

3 Task design

A task in SENSEVAL consists of three types of data: 1) A lexicon of word-to-sense mappings, with possibly extra information to explain, define, or distinguish the senses (e.g., WordNet); 2) A corpus of manually tagged text or samples of text that acts as the Gold Standard, and that is split into an optional training corpus and test corpus; and 3) An optional sense hierarchy or sense grouping to allow for fine or coarse grained sense distinctions to be used in scoring.

Regardless of the type of task, each system is required to tag the words specified in the test corpus with one or more tags in the lexicon. Supervised systems can train on the training corpus, if one is available.

The SENSEVAL committee issued general guidelines for designing a task (Edmonds 2000). But it was up to the individual task organisers, to design their own tasks since each had different constraints on resource availability (both human and data). Everyone, however, used a common XML data encoding format developed for SENSEVAL-2.

Specific issues in choosing and designing the resources for each task are described in the papers in this proceedings, and, more generally, by Kilgarriff and Rosenzweig (2000).

3.1 Lexicon and lexical samples

Each task organiser chose the lexicon for their task. Notably, WordNet was used for the first time in SENSEVAL. Version 1.7 for the English tasks, and versions of EuroWordNet for Spanish, Italian, and Estonian.

For the lexical sample tasks, the guidelines suggests that words be chosen from different parts of speech, different frequencies in the corpus, and different polysemies (i.e., number of senses). The number of words depended on the available resources. The sample words were kept secret from the wider community until the training data was released; however, the organisers consulted each other so that translations of some of the sample words could be used across tasks.

3.2 Tagged corpora

For the all-words tasks, the guidelines suggest that at least 5000 words of running text be selected, and that all content words be tagged.

For the lexical sample tasks, it was suggested that for each sample word, at least $75+15n$ corpus instances be chosen, where n is the number of senses of the word. Again, lack of resources might have precluded this much tagged data.

The **Gold Standard corpus** must be replicable; the goal is to have human taggers agree at least 90% of the time. Thus, at least two human taggers were required to tag every instance of a word. Taggers are allowed to tag with multiple tags and to use special tags for proper names, and unassignable senses. See the papers in this proceedings for more details.

For the evaluation, the corpus had to be divided into a training set and a test set. The **training set** is a random subset of the Gold Standard corpus, which allows supervised systems to train. Not all tasks supplied training data, so only 'unsupervised' systems could participate (e.g., in the English all-words task – although many systems trained on other corpora such as Semcor). The **test set** is the rest of the corpus, with tags removed, on which the systems would be evaluated. It was suggested that a 2:1

ratio of training to test data be used. Although somewhat different from what is normally used in machine learning, the committee felt that having more test data would give a more realistic indication of a system's performance (since more varied contexts per word would be tested), and, moreover, unsupervised systems would be less 'short-changed'.

All data sets are now in the public domain (on the SENSEVAL website).

3.3 Sense groupings

Since some sense inventories are two fine-grained for plausible sense disambiguation, the scoring program can take into account sense hierarchies or sense groupings. Optionally, a task could provide such a grouping of senses, so that choosing any sense within the group or higher in the hierarchy would count towards a system's overall score. For example, the WordNet hierarchy was used for English nouns, whereas a separate 'grouping' was specially constructed for the English verbs (since the verbs do not have a useful hierarchy in WordNet for scoring purposes). See the paper on the English tasks for more detail.

3.4 Common data format

All tasks used a specially defined common data format for encoding the tagged and untagged corpus examples. Specifically, it accommodated the multi-lingual nature of the data by using an XML document type definition which allowed for a flexible mapping from lexical items to their textual instances. Using XML also allowed for arbitrary character encodings in the corpora. The structure was designed so that individual instances of lexical items could be associated with multiple sense tags, and allowed for discontinuous phrasal lexical items. It did not, however allow for multiple phrasal items with overlapping portions in the surface string.

Another requirement was simplicity. This quality would not only facilitate the logistics of designing a task, but would also ease any hand annotation that may have been necessary. As a result, a standoff annotation system was not feasible. This restricted the format in such a way as to limit the feasibility of embedding extant annotation of the corpora and to require that participants use standoff annotation in submitting their answers for reasons of space efficiency.

The use of the common data format simplified many system's participation in multiple tasks, consequently furthering research into the comparison of WSD in different languages.

4 Evaluation procedure

The evaluation was run centrally from a single website at the University of Pennsylvania and followed the same procedure as used in the first SENSEVAL. For each task, data was released in three stages:

- **Trial data:** A small set of data so that participants can design their systems to use the data formats. No 'real' data was released.
- **Training data.**
- **Test data.**

Each team would register their system, and then download the data sets according to the schedule. After running their system on the test data, each team submitted their answers to the website for automatic scoring. Each team's results were returned to the team before the workshop, but the overall results were unveiled at the workshop.

4.1 Schedule

A schedule was set up for task organisers to prepare and submit their data to the central website, while participants followed a separate, more rigid (and in the end very tight), schedule for downloads and submissions.

Task organisers started preparing their data as far back as September 2000, but the real push occurred in the three months proceeding the competition period.

The competition period ran April 17 – June 18. Within this period, each task had a critical window defined to be the period from when the training data was first made available to the last day for answer submissions to that task. The critical window had to be a minimum of 21 days.

Participants could download and submit answers at any time during the critical window of a particular task, subject to the following constraints. A submission of answers must:

- not have occurred more than 7 days after downloading the test data,

- not have occurred more than 21 days after downloading the training data, and
- have occurred before the end of the critical window for the particular task

This set up allowed participants to have sufficient time to participate in several tasks over the whole competition period, while ensuring that on any particular task, a participant had a maximum of one week to run their system (and 3 weeks to train their system), which we hope did not give any time for tailoring systems to the specific words or the corpora of the competition.

4.2 Data distribution

Data for the tasks was distributed via a website at University of Pennsylvania. Participants were required to register for tasks in order to download the trial, training, and test data for the tasks, and to upload their answers. Each of these operations required authentication via a password chosen at the time of registration. Additionally, timestamps were recorded for each of these operations in order to enforce the timing constraints on a per-participant basis. The system was not secure, as a participant could register multiple times under different names and use the data from the first registration to perform the task at hand. However, there were no signs of security problems in the use of the website.

Use of the distribution center was recommended, not required, of the task organizers. All the tasks with the exception of the Japanese tasks used the distribution center. A nice by-product of this process in concert with the common data format was the development an overarching organization of all the SENSEVAL data, which is evident in the data available to the public domain.

4.3 Scoring and evaluation

The same answer format and scoring program was used for SENSEVAL-2 as was used in the first SENSEVAL.

Systems were allowed to tag a word with as many senses as appropriate, giving probabilities, if desired. If the task had a sense hierarchy or grouping, then fine- and coarse-grained scoring was done. In fine-grained scoring, a system had to give at least one of the Gold Standard senses. In coarse-grained scoring, all senses in the answer key and in system output are collapsed to their highest parent or group identifier. For sense hierarchies, mixed-grained scoring was also done: a system is given partial credit for choosing a sense that is a parent of the required sense according to Melamed and Resnik's (1997) scheme.

Systems were not required to tag all instances of a word, or even all words, thus, as in SENSEVAL-1, we used precision and recall to score the systems, although the metrics are not completely analogous to IR evaluation. **Recall** (percentage of right answers on all instances in the test set) is the basic measurement of accuracy in this task, because it shows how many correct disambiguations the system achieved overall. **Precision** (percentage of right answers in the set of answered instances) favours systems that are very accurate if only on a small subset of cases that the system chose to give answers to; the cases might be particularly easy to disambiguate, but this can be determined by comparing the answers to the baseline on the same subset (a type of analysis that has yet to be done). **Coverage,** the percentage of instances that a system gives any answer to, is also reported. Where available, baseline and inter-tagger agreement numbers are given.

No further data analysis was done. Thus, the question of who 'won' depends on your perspective, but, in fact, that is not the relevant question. The important thing is to examine how each system achieved the performance that it shows. Some of this analysis is given in the papers of this proceedings. (Note that in the results, where appropriate, we distinguished between supervised and unsupervised systems.)

When the results were unveiled at the workshop, it soon became apparent that bugs in the scoring software had potentially affected the results. It was decided by everyone present (on the first day) that all systems should be rescored. Also, owing to the tight schedule, some teams had made serious inadvertent errors in formatting their answers. Thus, it was also agreed that any team could resubmit their (corrected) answers before 31 July 2001. In so doing, the team would have to include an explanation about the modifications and only reasons of 'egregious' bugs would be allowed.

The official results list all original submissions scored with the debugged scorer, and all of the resubmissions, clearly identified. This compromise maintains the professionalism of SENSEVAL, as it does not devalue any team that met the original deadline, while encouraging the scientific purpose of the exercise.

5 Recommendations

Because the results were released so close to the workshop, there had been no time for detailed analysis. Thus, the workshop was structured around a series of panels about WSD and evaluation. Panels were held on domain-specific disambiguation, task design for new languages to SENSEVAL, sense distinctions, applications of WSD, and standardizing WordNets.

Ideally, the majority of the workshop content should have been about the analysis of WSD algorithms, so the major recommendation for future exercises is to allow at least one month for analysis before the workshop. Part of this recommendation is to have a proceedings at the workshop, rather than post-workshop as this one. A related recommendation is to gather information about systems (e.g., supervised / unsupervised, knowledge source, etc.) as they are registered.

Second, the use of different granularities and groupings for the lexicons in question yielded some unnecessary inconsistency across tasks. For example, the English tasks used a grouping which invalidated the mixed-grained scores, whereas the Swedish task used a hierarchy which yielded vacuous coarse-grained scores. This is actually a central issue in WSD, which should be addressed before the next SENSEVAL exercise. The data from SENSEVAL-2 should be invaluable in this research.

Finally, it was felt by some that the SENSEVAL organization up to now has been somewhat autocratic, which is true. This might have been suitable in the past, but we would all like SENSEVAL to become as open and scientifically professional an activity as possible, without sacrificing its grassroots quality. Notably, it's the only 'freelance' evaluation activity in the computational linguistics community, and so we recommend that a more democratic organization should be sought, which should include an official executive committee to oversee the future of SENSEVAL.

4 Acknowledgements

Many people contributed to SENSEVAL-2. The preface to this volume acknowledges everyone's contributions.

5 References

Phil Edmonds (2000). *Designing a task for SENSEVAL-2.* Technical Note. Senseval-2 website.

Adam Kilgarriff and Martha Palmer (2000) Guest editors. Special Issue on SENSEVAL: Evaluating Word Sense Disambiguation Programs. *Computers and the Humanities 34(1–2).*

Adam Kilgarriff and Joseph Rosenzweig (2000) Framework and results for English SENSEVAL. *Computers and the Humanities 34(1–2):15–48.*

Dan Melamed and Phil Resnik (2000) Tagger evaluation given hierarchical tag sets. *Computers and the Humanities 34(1–2).*

SENSEVAL Website:
http://www.itri.bton.ac.uk/events/senseval

SENSEVAL-2 Website:
www.sle.sharp.co.uk/senseval2

The Basque task: did systems perform in the upperbound?

Eneko Agirre, Elena Garcia, Mikel Lersundi, David Martinez, Eli Pociello
IxA NLP group, Basque Country University
649 pk.
20.080 Donostia, Spain
eneko@si.ehu.es

Abstract

In this paper we describe the Senseval 2 Basque lexical-sample task. The task comprised 40 words (15 nouns, 15 verbs and 10 adjectives) selected from *Euskal Hiztegia*, the main Basque dictionary. Most examples were taken from the *Egunkaria* newspaper. The method used to hand-tag the examples produced low inter-tagger agreement (75%) before arbitration. The four competing systems attained results well above the most frequent baseline and the best system scored 75% precision at 100% coverage. The paper includes an analysis of the tagging procedure used, as well as the performance of the competing systems. In particular, we argue that inter-tagger agreement is not a real upperbound for the Basque WSD task.

1 Introduction

This paper reviews the design of the lexical-sample task for Basque. The following steps were taken in order to build the hand-tagged corpus:

1. set the exercise
   a. choose sense inventory
   b. choose target corpus
   c. choose target words
   d. select examples from the corpus
2. hand-tagging
   a. define procedure
   b. tag
   c. analysis of inter-tagger agreement
   d. arbitration

The following section presents the setting of the exercise. Section 3 reviews the hand-tagging, and section 4 the results of the participant systems. Section 5 discusses the design of the task, as well as the results, and section 6 presents some future work.

2 Setting of the exercise

In this section we present the setting of the Basque lexical-sample exercise.

2.1 Basque

Basque is an agglutinative language, that is, for the formation of words, the dictionary entry independently takes each of the elements necessary for the different functions (syntactic case included). More specifically, the affixes corresponding to the determinant, number and declension case are taken in this order and independently of each other (deep morphological structure). One of the main characteristics of Basque is its declension system with numerous cases, which differentiates it from the languages spoken in the surrounding countries. An example follows (the order of the lemmas is the reverse):

etxekoari emaiozu
[Give it] [to the one in the house]

2.2 Sense inventory

We chose a published dictionary, *Euskal Hiztegia* (Sarasola, 1996), for the sense inventory. It is a monolingual dictionary of Basque. It is normative and repository of standard Basque. It was produced based mainly on literary tradition. The dictionary has 30,715 entries and 41,699 main senses (see comment on *nuances* below). The TEI version with all the information for each entry was included in the distribution. As the format was quite complex, another version was also included, which listed a plain list of word senses and multiword terms.

This dictionary has the particularity that word senses can have very specific sub-senses, called *nuances* which sometimes are illustrated with just an example and other times have a full definition. These *nuances* were also included in the set of word senses.

2.3 Corpora used

At first the EEBS balanced corpus was chosen, comprising one million words. Unfortunately this size is too small to provide the number of occurrences per word that was defined in the Senseval task specification. We therefore turned to the biggest corpus at hand, the *Egunkaria* corpus, comprising texts taken from the newspaper. The size of this corpus allowed us to easily reach the number of examples required. On the negative side, it is a specific corpus, and the distribution of the word senses could be highly biased. We used *Egunkaria* as the main corpus, but we also used the EEBS corpus in some cases, as we will see below.

2.4 Words chosen

The criterion to choose the 40 words (15 nouns, 15 verbs and 10 adjectives) was that they should cover all possible combinations of frequency, polysemy and skew[1]. The first two can be objectively determined before starting to hand tag, but skew could only be determined by introspection. After choosing a word, the expected skew was sometimes different from the desired skew.

A secondary criterion was the overlap with the words in other languages, and the overlap with a number of verbs that are being used for subcategorization and diathesis alternation studies in our group.

The English task organizers and the Spanish task organizers provided us with half of the words chosen in their lexical-sample task. This information could be used for cross-language mapping of word senses. Regarding the overlap with verbs, we plan to explore the influence of word senses in subcategorization and diathesis alternations.

We chose the first set of 40 words that covered more or less all combinations of the above phenomena from the set of translations of the words in the other tasks. This was done blindly, without knowing which specific word was chosen.

This first set was used to extract the examples (cf. following section), and the hand-taggers started to tag them. Unfortunately, for a number of words, all examples in the corpus referred to a single word sense. We had not foreseen this situation and took two measures:

1) Search for occurrences in the secondary EEBS corpus.
2) If occurrences of new senses were not found, then the word was discarded and a replacement word was chosen.

In order to find the replacement, the hand-tagger that was doing the arbitration scanned the examples of a word with similar polysemy and frequency and decided whether it had occurrences of more than one sense.

2.5 Selection of examples from corpora

The minimum number of examples for each word according to the task specifications was calculated as follows:

N=75+15*senses+6mword

where senses does not include the *nuances* (cf. section 2.2) and *mword* is the number of multiword terms that included the target word.

The minimum number of examples per word was extracted at random from the *Egunkaria* corpus, plus a 10% buffer. As explained in the previous section, for some words occurring in a single sense in this corpus, additional examples were taken from the secondary *EEBS* corpus. In this case, all available examples from *EEBS* were used, plus the examples from *Egunkaria* to meet the minimum number of examples required.

The context included 5 sentences, with the sentence with the target word appearing in the middle. Links were kept to the source corpus, document, and to the newspaper section when applicable.

The occurrences were split at random in training set (two thirds of all occurrences) and test set.

[1] By skew in this context, we mean the dominance of one sense over the others. It is given as the percentage of occurrences of the most frequent sense over all the others.

3 Hand tagging

Three persons, graduate linguistics students, took part in the tagging. They are familiar with word senses, as they are involved in the development of the Basque WordNet and cleaning the TEI version of the *Euskal Hiztegia* dictionary. The following procedure was defined for each word:

- The three of them would meet, read the definitions and examples given in the dictionary and discuss the meaning of each word sense. They tried to agree the meaning differences among the word senses.
- Two taggers independently tagged all examples for the word. No communication was allowed while tagging the word.
- Multiple tags were allowed, as well as the following tags: B new sense or multiword term, U unassignable. Examples with these tags were removed from the final release.
- A program was used to compute agreement rate and output those occurrences where there was disagreement grouped by the senses assigned.
- The third tagger, the referee, reviewed the disagreements and decided which one was correct.

For the word *itzal* (shadow), the disagreement was specially high. The taggers decided that the definitions and examples were too confusing, and decided to replace it with another word.

Overall, the two taggers agreed 75% of the time. Some words attained an agreement rate above 95% (e.g. nouns *kanal* – *channel* – or *tentsio* –*tension* –), but others like *herri* – town/people/nation – attained only 52% agreement.

All in all, 5284 occurrences of the 40 words were released. On average, one hand-tagger took 0.41 minutes per occurrence and the other 0.55 minutes. The referee took 0.22 minutes per entry, including selection of replacement words. Time for arbitration meeting is also included.

4 Participants and Results

Three different teams and four systems took part in the tagging: John Hopkins University (JHU), Basque Country University (BCU-EHU-dlist-all and BCU-EHU-dlist-best) and University of Maryland (UMD). The third team submitted the results later, out of the Senseval competition. The results for the fine-grain scoring are shown in table 1, including the Most Frequent Sense baseline (MFS). Assuming full coverage, JHU attains the best performance. BCU-EHU-dlist-best has the best precision, but only tags 57% of the occurrences.

Prec.	Recall	Attempted	System
0,849	0,483	56,9%	BCU-ehu-dlist-best
0,757	0,757	100%	JHU
0,732	0,732	100%	BCU-ehu-dlist-all
0,703	0,703	100%	UMD
0,648	0,648	100%	MFS

Table 1: results of systems and MFS baseline. UMD submitted results after the deadline.

5 Discussion

These are the main issues that we think are interesting for further discussion.

Dictionary used. Before designing the task, we had to choose between two possible dictionaries: the Basque WordNet and the *Euskal Hiztegia* dictionary. Another alternative was to start the lexicographer's work afresh, defining the word senses as the tagging proceeded. We thought the printed dictionary would provide clear-cut sense distinctions that would allow the tagging to be easier. After the tagging, the hand-taggers complained that this was not the case. They think that the tagging would be much more satisfactory had they defined the word senses directly from the corpus.

In particular, they were not allowed to introduce new senses or multiword terms, and such examples were discarded.

Corpus used. There was a mismatch between the dictionary and the corpus: the corpus was linked to a specific genre, and this resulted in having some senses which were not included in the dictionary. Besides, many senses in the dictionary did not appear in our corpus, and some words had to be replaced. This caused the taggers some overwork, but did not influence the quality of the result.

Hand-tagging is a very unpleasant task. When asked about future editions, the hand taggers suggested the following: "please do get somebody else". We have to note that the hand taggers are used to repetitive tasks, such as building the Basque WordNet or cleaning-up the TEI version of *Euskal Hiztegia.*

Inter-tagger agreement. Part of the disagreement was caused by typos and mistakes. Nevertheless, we think that the low inter-tagger agreement (75%) was caused mainly by the procedure used to tag the occurrences. The taggers met and tried to understand the word senses, but the fact is that it was only after tagging a few occurrences that they started to really conceptualise the word senses and draw specific lines among one sense and the others. If both taggers had been allowed to meet (at least once) while they were tagging, they could have discussed and agreed on a common conceptualisation. The referee found that most of the times whole sets of examples were systematically tagged differently by each of the taggers, that is, each of the taggers had a different criterion about the word sense applicable to that set of examples. The referee then had to decide on the tag for those sets of examples.

Systems performing as good as inter-tagger agreement. Traditionally, inter-tagger agreement has been used as an upperbound for the performance of machines in cognitive tasks. We think that in this case, a system may perform better on the Basque WSD task than a human, in the sense that if the taggers were evaluated against the gold standard they would score lower that the systems. In fact, current systems, which are still under development for Basque, reach the same performance as humans. Are machines performing better than humans? We think that inter-tagger agreement, at least as derived from the procedure used in this exercise, is not a real upperbound, and that systems can easily perform better.

The gold standard reflects the conceptualization of one human, the referee, which does not have to agree with the conceptualization made by other persons (specially if these are done in isolation). People disagree whether in a certain occurrence this word sense or the other applies, i.e. they can disagree in the meaning of the word senses as defined in the dictionary. In fact, trying to achieve a common ground when reading the dictionary definitions sometimes produced heated debate in the meetings.

If the gold standard reflects a systematic conceptualization of a person, machine learning algorithms can learn to replicate these conceptualization (categorizations in this case), and achieve high degrees of agreement with the person behind the gold standard. This does not mean that the system is smarter than the human taggers, but rather that the system has no opinion on his own, and just imitates one of the persons.

Error reduction similar to English task. The best recall for Basque was 75% vs. 64% of the MFS baseline. In English the best system achieved 64% recall vs. 47% of the most frequent sense baseline (called commonest baseline in the official results). It is clear that the skew of the Basque words allowed for higher results. On the other hand, the error reduction for Basque was 29%, compared to 32% for English. This implies that systems could effectively learn from the data in both tasks.

No use of domain tags, full documents. No system used the extra information provided by the full documents or the domain tags.

6 Future work

First of all, we plan to explore the use of other procedures for the hand-tagging. We think that the data attained high levels of quality (which has been shown by the error reduction attained by the participating systems over the MFS baseline), but still we are not satisfied with the sense inventory used.

Further analysis of the results of the participating systems is also planned, as Kappa statistics and the performance of the combination of the systems.

Bibliography

Sarasola, I., 1996, *Euskal Hiztegia*, Donostia, Gipuzkoako Kutxa.

Dutch Word Sense Disambiguation: Data and Preliminary Results

Iris Hendrickx* and **Antal van den Bosch***,+
* ILK / Computational Linguistics, Tilburg University, NL-5000 LE Tilburg, The Netherlands
+ WhizBang! Labs–Research, 4616 Henry Street, Pittsburgh PA 15213, USA

Abstract

We describe the Dutch word sense disambiguation data submitted to SENSEVAL-2, and give preliminary results on the data using a WSD system based on memory-based learning and statistical keyword selection.

1 Introduction

Solving lexical ambiguity, or word sense disambiguation (WSD), is an important task in Natural Language Processing systems. Much like syntactic word-class disambiguation, it is not a end in itself, but rather a subtask of other natural language processing tasks (Kilgarriff and Rozenzweig, 2000). The problem is far from solved, and research and competition in the development of WSD systems in isolation is merited, preferrably on many different languages and genres.

Here we introduce the first electronic Dutch word-sense annotated corpus, that was collected under a sociolinguistic research project (Schrooten and Vermeer, 1994), and was kindly donated by the team coordinators to the WSD systems community. In this paper we describe the original data and the preprocessing steps that were applied to it before submission to the SENSEVAL-2, in Section 2. We also present the first, preliminary, results obtained with MBWSD-D, the Memory-Based Word-Sense Disambiguation system for Dutch, that uses statistical keyword selection, in Section 3.

2 Data: The Dutch child book corpus

The Dutch WSD corpus was built as a part of a sociolinguistic project, led by Walter Schrooten and Anne Vermeer (1994), on the active vocabulary of children in the age of 4 to 12 in the Netherlands. The aim of developing the corpus was to have a realistic wordlist of the most common words used at elementary schools. This wordlist was further used in the study to make literacy tests, including tests how many senses of ambiguous words were known by children of different ages.

The corpus consists of texts of 102 illustrated children books in the age range of 4 to 12. Each word in these texts is manually annotated with its appropriate sense. The data was annotated by six persons who all processed a different part of the data.

Each word in the dataset has a non-hierarchical, symbolic sense tag, realised as a mnemonic description of the specific meaning the word has in the sentence, often using a related term. As there was no gold standard sense set of Dutch available, Schrooten and Vermeer have made their own set of senses.

Sense tags consist of the word's lemma and a sense description of one or two words (*drogen_nat*) or a reference of the grammatical category (*fiets_N, fietsen_V*). Verbs have as their tag their lemma and often a reference to their function in the sentence (*is/zijn_kww*). When a word has only one sense, this is represented with a simple "=". Names and sound imitations also have "=" as their sense tag.

The dataset also contains senses that span over multiple words. These multi-word expressions cover idiomatic expressions, sayings, proverbs, and strong collocations. Each word in the corpus that is part of such multi-word expression has as its meaning the atomic meaning of the expression.

These are two example sentences in the corpus:

"/= het/het_lidwoord raadsel/= van/van_prepositie de/= verdwenen/verdwijnen regenboog/= kan/kunnen_mogelijkheid alleen/alleen_adv

# tokens	152.758
# types	10.263
# sentences	12.287
# words per sentence	12.4
# unambiguous words	9.095
# words that occurs once	4.949
# sense tags	9319
# word/sense combinations occuring once	6.702
% of ambiguous tokens in corpus	54

Table 1: Basic corpus statistics

met/met_prepositie geweld/= opgelost/oplossen_probleem worden/worden_hww ,"/= zeiden/zeggen_praten de/= koningen/koning ./= toen/toen_adv verklaarden/verklaren_oorlog ze/= elkaar/=de/= oorlog/= ./=

The dataset needed some adaptations to make it fully usable for computational purposes. First, spelling and consistency errors have been corrected for most part, but in the data submitted to SENSEVAL-2, a certain amount of errors is still present. Second, in Dutch, prepositions are often combined with verbs as particles and these combinations have other meanings than the two separate words. Unfortunately the annotations of these cases were rather inconsistent and for that reason it was decided to give all prepositions the same sense tag *"/prepositie"* after their lemma.

The dataset consists of approximately 150,000 tokens (words and punctuation tokens) and about 10,000 different word forms. Nine thousand of these words have only one sense, leaving a thousand word types to disambiguate. These ambiguous types account for 54 % of the tokens in the corpus. The basic numbers can be found in Table 1.

For the SENSEVAL-2 competition, the dataset was divided in two parts. The training set consisted of 76 books and approximately 115.000 words. The test set consisted of the remaining 26 books and had about 38.000 words.

3 The MBWSD-D system and preliminary results

We first describe the representation of the corpus data in examples presented to a memory-based learner in Subsection 3.1. We then describe the architecture of the system in Subsection 3.2, and we then present its preliminary results in Subsection 4.

3.1 Representation: Local and keyword features

As a general idea, disambiguation information is assumed to be present in the not-too-distant context of ambiguous words; the present instantiation of MBWSD-D limits this to the sentence the ambiguous word occurs in. Sentences are not represented as is, but rather as limited sets of features expected to give salient information about which sense of the word applies.

The first source of useful disambiguation information can be found immediately adjacent to the ambiguous word.It has been found that a four-word window, two words before the target word and two words after gives good results; cf. (Veenstra et al., 2000).

Second, information about the grammatical category of the target word and its direct context words can also be valuable. Consequently, each sentence of the Dutch corpus was tagged and the part-of-speech (POS) tags of the word and its direct context (two left, two right) are included in the representation of the sentence. Part-of-speech tagging was done with the Memory Based Tagger (Daelemans et al., 1996).

Third, informative words in the context ('keywords') are detected based on the statistical chi-squared test. Chi-square estimates the significance, or degree of surprise, of the number of keyword occurrences with respect to the expected number of occurrences (apriori probability):

$$X^2 = \sum_{k=1}^{n} \frac{(f_k - e_k)^2}{e_k} \qquad (1)$$

where f_i is the keyword frequency and e_i is the expected frequency. f_i is the word frequency and e_i is the expected word frequency. The expected frequency of the keyword is given in equation 3.1. It must be noted that the Chi-Square method cannot be considered reliable when the expected frequency has a value below 5: $e_i = (f_w i / f_w) * f_k$, where f_i is the frequency the ambiguous word w of sense i, f_w is the frequency of word w and f_k is the frequency of the keyword.

The number of occurrences of a very good keyword will have a strong deviation of its expected number of occurrences divided over the senses. The expected probability with respect to all senses can be seen as a distribution of the keyword. A good keyword is a word that differs from the expected distribution and always co-occurs with a certain sense, or never co-occurs with a certain sense.

In sum, a representation of an instance of an ambiguous word consists of the two words before the target word, two words after the word, the POS tags of these words and of the target word itself, a number of selected keywords, and of course the annotated sense of the word as the class label.

3.2 System architecture

Following the example of ILK's previous word-sense disambiguation system for English (Veenstra et al., 2000), it was decided to use word experts. Berleant (Berleant, 1995) defines a word expert as follows: "A word expert is a small expert system-like module for processing a particular word based on other words in its vicinity" (1995, p.1). Word experts are common in the field of word sense disambiguation, because words are very different from each other. Words all have different numbers of senses, different frequencies and need different information sources for disambiguation. With word experts, each word can be treated with its own optimal method.

Making word experts for every ambiguous word may not be useful because many words occur only a few times in the corpus. It was decided to create word experts for wordforms with a threshold of minimal 10 occurrences in the training set. There are 524 of such words in the training set. 10 is a rather low threshold, but many words can be easily disambiguated by knowledge a single feature value, such as of their part-of-speech tag.

The software for emulating memory-based learning used in this research is TiMBL (Tilburg Memory-Based Learner). TiMBL (Daelemans et al., 2001) is a software package developed by the ILK research group at Tilburg University. TiMBL implements several memory-based classifiers. In essence, memory-based classifiers use stored classified examples to disambiguate new examples.

For each word a TiMBL word expert was trained on that portion of the training corpus that consisted of sentence representations containing that word. TiMBL was trained 300 times, each time with another combination of parameters. Each of these training sessions was evaluated with leave-one-out cross validation (Weiss and Kulikowski, 1991) to select the optimal TiMBL setting for a particular word, to be used eventually for classifying the test material.

For each word expert a total of 300 experiments were performed, each with another combination of parameter settings. In this study the following options were used (cf. (Daelemans et al., 2001) for first pointers to descriptions of these metrics and functions):

distance-weighted voting : (1) all neighbors have equal weight; (2) Inverse Distance weighting; (3) Inverse Linear weighting

feature weighting : (1) no weighting; (2) Gain Ratio; (3) Information Gain; (4) Chi Square; (5) Shared Variance

similarity metric : (1) Overlap metric; (2) MVDM

number of nearest neighbours : 1, 3, 5, 7, 9, 11, 15, 25, 45, and 75

The last step for each word expert was to test the optimal settings on the test set. To evaluate the results, described in the next Section, the results were compared with a baseline score. The baseline was to select for each word the most frequent sense.

4 Results

The top line of Table 2 shows the mean score of all the word experts together on the test set. The score of the word experts on the test set, 84.1%, is generously higher than the baseline score of 74.1%. These are the results of the word experts only; the second row also includes the best-guess outputs for the lower-frequency words, lowering the system's performance slightly.

The same results, now split on the frequency of the words in the training set, can be seen in Table 3. The first column shows the frequency groups, based on the word frequencies in the training set, the second the number of words in

test selection	#words	baseline	system
word-expert words	15365	74.1	84.1
all ambiguous words	16686	74.6	83.8
all words	37770	88.8	92.9

Table 2: Summary of results on test material

the test set, and the third column shows the mean score of the WSD system. The scores tend to get better as the frequency goes up, except for the group of 40-49, which has the lowest score of all. Note that the baseline score of the group of words with a frequency below 10 is relatively high: 80.5%.

frequency	#words	baseline	system
<10	1321	–	80.5
10-19	868	63.0	76.8
20-29	644	70.3	79.5
30-39	503	75.9	83.3
40-49	390	66.7	75.9
50-99	1873	73.7	85.4
100-199	2289	77.7	83.1
$\geq$ 200	8798	74.6	85.6
> 100	10995	75.3	85.1

Table 3: Results divided into frequency groups

We can also calculate the score on all the words in the text, including the unambiguous words, to give an impression of the overall performance. The unambiguous words are given a score of 100%, because the task was to disambiguate the ambiguous words. It might be useful for a disambiguation system to tag unambiguous words with their lemma, but the kind of tagging this is not of interest in our task. The third row of Table 2 shows the results on all words in which the system was applied with a threshold of 10: The system scores 4 % higher than the baseline.

5 Discussion

This paper introduced a Dutch child book corpus, generously donated to the WSD community by the team leaders of the sociolinguistic project that produced the corpus. The data is annotated with a non-hierarchical mnemonic sense inventory. The data has been cleaned up and split for the SENSEVAL-2 competition.

The data provides an arguably interesting case of a "flat" semantic tagging, where there is obviously no gain from a governing wordnet, but alternatively it is not negatively biased by an inappropriate or badly-structured wordnet either. Learnability results are therefore an interesting baseline to beat when the data would be annotated with a Dutch wordnet.

The system applied to the data as a first indication of its complexity and learnability, consisted of an ensemble of word experts trained to disambiguate particular ambiguous word forms. The score of the system on the 16686 ambiguous words in the test set was 83.8% compared to a baseline score of 74.6%. On free heldout text the system achieved a result of 92.9%; 4% over the baseline of 88.8%, or in other words yielding an error reduction of about 37%. These absolute and relative figures are roughly comparable to performances of other systems on other data, indicating at least that the data represents learnability properties typical for the WSD area.

References

D. Berleant. 1995. Engineering word-experts for word disambiguation. *Natural Language Engineering*, pages 339–362.

W. Daelemans, J. Zavrel, and P. Berck. 1996. Part-of-speech tagging of dutch with mbt, a memory-based tagger generator. In *Congresboek van de Interdisciplinaire Onderzoeksconferentie Informatiewetenchap.*

W. Daelemans, J. Zavrel, K. van der Sloot, and A. van den Bosch. 2001. Timbl: Tilburg memory based learner, version 4.0, reference guide. Technical report, Tilburg University.

A. Kilgarriff and J. Rozenzweig. 2000. Framework and results for english senseval. *Computers and the Humanities*, 34.

W. Schrooten and A. Vermeer. 1994. *Woorden in het basisonderwijs. 15.000 woorden aangeboden aan leerlingen.* TUP(Studies in meertaligheid 6).

J. Veenstra, A. van den Bosch, S. Buchholz, W. Daelemans, and J. Zavrel. 2000. Memory-based word sense disambiguation. *Computers and the Humanities*, 34.

S. Weiss and C. Kulikowski. 1991. *computer systems that learn.* Morgan Kaufmann.

English Lexical Sample Task Description

Adam Kilgarriff
ITRI, University of Brighton
Brighton, UK
adam@itri.bton.ac.uk

The English lexical sample task (adjectives and nouns) for SENSEVAL 2 was set up according to the same principles as for SENSEVAL-1, as reported in (Kilgarriff and Rosenzweig, 2000). (Adjectives and nouns only, because the data preparation for the verbs lexical sample was undertaken alongside that for the English all-words task, and is reported in Palmer et al (this volume). All discussion below up to the Results section covers only adjectives and nouns.)

1 Lexical sample

The lexicon was sampled to give a range of low, medium and high frequency words (see Table 1). These were all different words to the ones used in SENSEVAL 1.

2 Corpus choice

For the most part, the British National Corpus (New edition) was used. (The new edition has the advantage that it is available worldwide, so all participants had the opportunity of obtaining it for system training.) Our goal was to match this source, containing British English, with another, of American English. In the event, only limited quantities of corpus data for American English were available without copyright complications, so the lion's share of the data was from the BNC with a limited quantity from the Wall Street Journal.

In accordance with standard SENSEVAL procedure, the goal was to have $75 + 15n + 6m$ instances for each lexical-sample word, where n is the number of senses the word has and m is the number of multiword expressions that the word is part of (both, of course, relative to a specific lexicon). In practice numbers varied slightly, as instances were deleted because they had the wrong part of speech or were otherwise unusable. See Table 1 for actual numbers of senses, multiwords expressions and instances.

3 Lexicon choice

Here lay the biggest contrast with the SENSEVAL-1 task, which had used Oxford University Press's experimental HECTOR lexicon. This time, in response to popular acclaim, WordNet was used.

Since SENSEVAL was first mooted, in 1997, WordNet-or-not-WordNet has been a recurring theme. In favour was the argument that it was already very widely used, almost a *de facto* standard. The argument against concerned its sense distinctions. WordNet, like thesauruses but unlike standard dictionaries, is organised around groups of words of similar meanings (*synsets*), not around words (with their various meanings). This means that the priority for the lexicographer is building coherent synsets rather than the coherent analysis of the various meanings of a particular word. The writer of a thesaurus does not need to pay as much attention to the distinction between two senses of a word, as the writer of a dictionary. Word sense disambiguation is a task which needs clear and well-motivated sense distinctions. In English SENSEVAL-1, WordNet was not used because of concerns that it did not provide clean enough sense distinctions.

While HECTOR provided good sense distinctions, it was unsatisfactory in that it did not cover the whole lexicon so there was no possibility of scaling up. The case for WordNet – that it was already integrated into so much NLP and WSD work – still stood, so the decision was made to use WordNet. To guard against cases where WordNet made a distinction between two meanings, but it was not clear what the distinction was, all the words in the lexical sample had their entries reviewed by a

Word	Ss	Mwe	inst	ITA
ADJS: lexical sample size: 15				
blind	3	21	163	89.6
colorless	2	0	103	94.2
cool	6	1	158	92.1
faithful	3	0	70	94.6
fine	9	6	212	84.0
fit	3	0	86	85.0
free	8	36	247	79.2
graceful	2	0	85	72.6
green	7	80	284	86.6
local	3	12	113	89.1
natural	10	37	309	72.4
oblique	2	5	86	96.4
simple	7	19	196	67.8
solemn	2	0	77	84.1
vital	4	7	112	93.7
ALL ADJS			2301	83.4
NOUNS: lexical sample size: 29				
art	5	35	294	78.5
authority	7	6	276	84.3
bar	13	57	455	87.3
bum	4	0	137	91.7
chair	4	35	207	92.8
channel	7	10	218	84.8
child	4	16	193	92.3
church	3	21	192	88.0
circuit	6	31	255	93.5
day	9	82	434	76.3
detention	2	5	95	98.7
dyke	2	0	86	96.5
facility	5	9	172	89.5
fatigue	4	6	128	97.7
feeling	6	5	153	77.0
grip	7	3	153	85.2
hearth	3	1	96	85.0
holiday	2	9	93	90.5
lady	3	27	158	74.1
material	5	39	209	85.1
mouth	8	10	179	88.7
nation	3	10	112	90.5
nature	5	8	138	86.7
post	8	33	236	87.7
restraint	6	3	136	80.4
sense	5	37	160	87.1
spade	3	7	98	95.1
stress	5	7	118	74.7
yew	2	15	85	97.1
ALL NOUNS			5266	86.3
ALL			7567	85.5

Table 1: Lexical sample: rubric for column headers: Ss=number of fine-grained senses; Mwe = number of multi-word expressions which the word participates in (as *bear* participates in WordNet headword *polar bear*); inst = number of instances tagged; ITA = inter-tagger agreement (fine-grained).

lexicographer, with a view particularly to merging insufficiently-distinct senses. It was initially unclear how these revisions would relate to the publicly available version of WordNet (at that time, WordNet 1.6). We are very grateful to the Princeton WordNet team (George Miller, Christiane Fellbaum and Randee Tengi) for their help at this point; they agreed to incorporate our proposed revisions into a new version of WordNet (1.7) which was then made available in time (despite some very tight deadlines) for the SENSEVAL competition.

WordNet 1.7 was not available as a complete object at the time of the gold standard production, in Spring 2001, but the entries for the lexical sample words were fixed at that point. For each lexical sample entry, we produced an HTML version for the lexicographers to work from. In addition to all the relevant information in WordNet, this had a mnemonic for each sense, so that taggers could use mnemonics when doing the tagging, rather than easily-forgotten, easily-confused sense numbers. The mnemonics were selected by a lexicographer.

4 Gold standard production

Once the corpus sources and lexical entries were fixed, work could proceed with the Gold-Standard tagging.[1]

First, a team of three professional lexicographers and fourteen students and others was recruited. Recruitment proceeded as follows: an aptitude test was set up on the web. The test involved sense-tagging some corpus instances (taken from SENSEVAL-1, so the gold-standard answers were known). Email postings were made asking interested people to visit the website and take the test. All applicants scoring sufficiently well on the test were then offered work, on a piecework basis.

An HTML version of the corpus for a word was prepared. This comprised a series of ten-sentence stretches of text, with one word in the last of the sentences highlighted; that was the word to be sense-tagged. The files were HTML versions of the XML files used for test and training data.

A tagger was emailed the lexical entry and corpus for a word. They then tagged it, and

[1]The tagging was supported by a grant from EPSRC, the UK funding council, under GR/R02337/01 (MATS).

returned, by email, a file of answers. These files were checked automatically, and if they contained 'answers' which were not possible answers for the word, the suspect items were automatically emailed back to the tagger for correction.

The tagger guidelines are available along with other resources for the English-lexical-sample task. They developed in the course of the exercise; when a tagger asked a pertinent questions, I circulated the question and my answer to all taggers and incorporated them into the guidelines.

As in SENSEVAL-1, "Unassignable" and "Proper-name" tags were always available alongside regular tags, and taggers were told to put down more than one tag, where multiple tags were equally applicable. Taggers were also asked to mark items where the part of speech was wrong; these were then deleted from the dataset.

5 Tagger agreement procedures and scores

As in all exercises where a gold standard corpus is the goal, it was necessary to have all data tagged by more than one person. The question then arises, how many taggings does each item need? The algorithm adopted here was:

1. send item out to two taggers
2. if they agree completely, **stop; return agreed answer**
3. else, send out to another tagger
4. is there one or more tag that two agree on?
5. if yes, **stop; return all tags which two people agree on**
6. if no, return to step 3

Thus, in simple cases, a minimum of effort was used, but in difficult cases, more opinions were obtained. The number of taggings per items is shown below. Note that the algorithm stops at step 2 if both taggers agree on one tag, or if both taggers agree on two or more tags.

Taggings	Number	%
2	5032	66.5
3	2446	32.3
4	86	1.1
5	4	0.05

3 taggers' answers			GS	cases
A	A	B	A	651
A	A;B	A	A	550
A	A;B	B	A;B	209
A	A;B	A;B	A;B	189
A	A;B	C	A	162
A	A	A;B;C	A	67
A	A;B	A;C	A	51
A	A	B;C	A	44
A;B	A;C	C	A;C	41
A;B	A:B;C	C	A;B;C	38

Table 2: Patterns of (dis)agreement for 3-tagger cases. GS = gold standard tagging arising from these human taggings. ";" used as separator where a tagger (or the gold standard) gave multiple tags.

Of the 5032 two-tagger items, in 4688 cases, the taggers agreed on one tag; in 340 cases, on two tags; and in 4 cases, on three tags.

For the 2446 cases which were tagged three times, 136 were cases where all three taggers agreed perfectly (so, had the algorithm been followed to the letter, the item would not have been tagged a third time; such cases were caused by delays in taggers returning answers.) The common patterns amongst the remainder are shown in Table 2.

For the 86 cases with four taggers, half the cases were {A, A, B, C} taggings.

Fine-grained inter-tagger agreement (ITA) figures was calculated using the same scoring algorithm as for the systems.[2] For each pair of taggers tagging an instance, two scores were calculated, one with the one answer as the key, the other with the other. For each instance, scores were normalised so that the maximum score for each corpus instance was one, however many times it had been tagged. The overall ITA was 85.5%. A breakdown by word and by word class is given in Table 1.[3]

[2] All ITA figures and other results reported in this paper refer to fine-grained sense distinctions. The grouping of senses into coarse-grained categories took place independently of the gold-standard preparation, which was based entirely on fine sense distinctions.

[3] Kappa was not calculated because there were various ways in which it might have been calculated, so it was unclear which was appropriate, and it would have introduced more complication than clarification. Also

As argued in (Kilgarriff and Rosenzweig, 2000) (also (Kilgarriff, 1999)) the inter-tagger agreement figure for a gold standard is of less interest than the replicability figure: if a completely different team of taggers used the same methodology to do the same task, what would the agreement level between the two teams' outputs be? It is the replicability figure, rather than ITA, which defines an upper bound for the task. We have not yet had time to conduct such a study.

6 Task organisation

The organisation followed standard SENSEVAL procedure. The data was prepared in XML using SENSEVAL DTDs, with the data for each word split in a ration of 2:1 between training and test data. Data distribution, results uploads, baselines and scoring were handled at UPenn (see paper by Cotton and Edmonds).

7 Results

Results are presented in the table below. Owing to space constraints, where a team submitted multiple systems with similar results, only the best result is shown. Full results are available at the SENSEVAL website, as are decodings of system names. At the SENSEVAL workshop (5–6 July 2001) it was agreed that there should also be a later deadline (end July 2001) so that 'egregious bugs' could be fixed. In order to honour both standard practice in evaluation exercises (eg, no extension of deadlines) and also the agreement made at the workshop, both results sets are presented, with later-deadline results marked with (R) as a suffix to the name.

There has not yet been time for an analysis of the results. The one comment that does seem pertinent is the contrast with the English-lexical-sample task in SENSEVAL-1. The tasks were organised in similar ways, and some of the systems were improved versions of systems participating in 1998. Yet the performance of the best systems has, apparently, dropped around 14%. We may well ask, why?

We believe the drop is due to the choice of lexicon. As discussed above, using WordNet for SENSEVAL has drawbacks. High-accuracy word sense disambiguation is only possible where the lexicon makes clear and well-motivated sense distinctions, and provides sufficient information about the distinctions for the disambiguation algorithm to build on. An implication for future WSD research is that it is time to turn our attention from algorithms, to sense distinctions.

PR	ATT	System
Supervised systems		
.82	28	BCU ehu-dlist-best
.67	25	IRST
.64	100	JHU (R)
.64	100	SMUls
.63	100	KUNLP
.62	100	Stanford-CS224
.61	100	Sinequa-LIA SCT
.59	100	TALP
.57	98	BCU ehu-dlist-all
.57	100	Duluth-3
.57	100	UMD-SST
.50	100	UNED LS-T
.42	98	Alicante
Supervised baselines		
.51	100	Base Lesk
.48	100	Base Commonest
Unsupervised systems		
.58	55	ITRI-WASPS
40	100	UNED-LS-U
.29	100	CLresearch DIMAP
.25	99	IIT-2 (R)
Unsupervised baselines		
.16	100	Base Lesk-defs
.14	100	Base random

Table 3: PR=system precision; ATT= percentage of cases for which an answer was returned ("attempted").

References

Adam Kilgarriff and Joseph Rosenzweig. 2000. Framework and results for English SENSEVAL. *Computers and the Humanities*, 34(1–2):15–48. Special Issue on SENSEVAL, edited by Adam Kilgarriff and Martha Palmer.

Adam Kilgarriff. 1999. 95% replicability for manual word sense tagging. In *Proc. EACL*, pages 277–278, Bergen, June.

the figures shown, unlike kappa figures, have the merit of being directly comparable with system performance scores.

English Tasks: All-Words and Verb Lexical Sample

Martha Palmer, Christiane Fellbaum, Scott Cotton,
Lauren Delfs, and **Hoa Trang Dang**
University of Pennsylvania
{mpalmer,fellbaum,cotton,lcdelfs,htd}@linc.cis.upenn.edu

Abstract

We describe our experience in preparing the lexicon and sense-tagged corpora used in the English all-words and lexical sample tasks of SENSEVAL-2.

1 Overview

The English lexical sample task is the result of a coordinated effort between the University of Pennsylvania, which provided training/test data for the verbs, and Adam Kilgarriff at Brighton, who provided the training/test data for the nouns and adjectives (see Kilgarriff, this issue). In addition, we provided the test data for the English all-words task. The pre-release version of WordNet 1.7 from Princeton was used as the sense inventory. Most of the revisions of sense definitions relevant to the English tasks were done prior to the bulk of the tagging.

The manual annotation for both the English all-words and verb lexical sample tasks was done by researchers and students in linguistics and computational linguistics at the University of Pennsylvania. All of the verbs in both the lexical sample and all-words tasks were annotated using a graphical tagging interface that allowed the annotators to tag instances by verb type and view the sentences surrounding the instances. Well over 1000 person hours went into the tagging tasks.

2 English All-Words Task

The test data for the English all-words task consisted of 5,000 words of running text from three Wall Street Journal articles representing varied domains from the Penn Treebank II. Annotators preparing the data were allowed to indicate at most one multi-word construction for each content word to be tagged, but could give multiple senses for the construction. In some cases, a multi-word construction was annotated with senses associated with just the head word of the phrase in addition to more specific senses based on the entire phrase. The annotations were done under a double-blind scheme by two linguistics students, and were then adjudicated and corrected by a different person.

Task participants were supplied with test data only, in the standard all-words format for SENSEVAL-2, as well as the original syntactic

System	Precision	Recall
SMUaw-	0.690	0.690
AVe-Antwerp	0.636	0.636
LIA-Sinequa-AllWords	0.618	0.618
david-fa-UNED-AW-T	0.575	0.569
david-fa-UNED-AW-U	0.556	0.550
gchao2-	0.475	0.454
gchao3-	0.474	0.453
Ken-Litkowski-clr-aw (*)	0.451	0.451
Ken-Litkowski-clr-aw	0.416	0.451
gchao-	0.500	0.449
cm.guo-usm-english-tagger2	0.360	0.360
magnini2-irst-eng-all	0.748	0.357
cmguo-usm-english-tagger	0.345	0.338
c.guo-usm-english-tagger3	0.336	0.336
agirre2-ehu-dlist-all	0.572	0.291
judita-	0.440	0.200
dianam-system3ospdana	0.545	0.169
dianam-system2ospd	0.566	0.169
dianam-system1	0.598	0.140
woody-IIT2	0.328	0.038
woody-IIT3	0.294	0.034
woody-IIT1	0.287	0.033

Table 1: System performance on English all-words task (fine-grained scores); (*) indicates system results that were submitted after the SENSEVAL-2 workshop and official deadline.

Christiane Fellbaum is at Princeton University, fellbaum@clarity.princeton.edu

and part-of-speech annotations from the Treebank. Table 1 shows the system performance on the task. Most of the systems tagged almost all the content words. This included not only indicating the appropriate sense from the WordNet 1.7 pre-release (as it stood at the time of annotation), but also marking multi-word constructions appropriate to the corresponding sense tags. If given a perfect lemmatizer, a simple baseline strategy which does not attempt to find the satellite words in multi-word constructions, but which simply tags each head word with the first WordNet sense for the corresponding Treebank part-of-speech tag, would result in precision and recall of about 0.57.

3 English Lexical Sample Task

The data for the verb lexical sample task came primarily from the Penn Treebank II Wall Street Journal corpus. However, where that did not supply enough samples to approximate 75+15*n instances per verb, where n is the number of senses for the verb, we supplemented with British National Corpus instances. We did not find sentences for every sense of every word we tagged. We also sometimes found sentences for which none of the available senses were appropriate, and these were discarded. The instances for each verb were partitioned into training/test data using a ratio of 2:1.

We also grouped the nouns, adjectives and verbs for the lexical sample task, attempting to be explicit about the criteria for each grouping. In particular, the criteria for grouping verbs included differences in semantic classes of arguments, differences in the number and type of arguments, whether an argument refers to a created entity or a resultant state, whether an event involves concrete or abstract entities or constitutes a mental act, whether there is a specialized subject domain, etc. All of the verbs were grouped by two or more people, with differences being reconciled. In some cases the groupings of the verbs are identical to the existing WordNet groupings; in some cases they are quite different. The nouns and adjectives were grouped by the primary annotator in the project; WordNet does not have comparable groups for nouns and adjectives.

These groupings were used for coarse-grained scoring, under the framework of SENSEVAL-1. After the SENSEVAL-2 workshop, participants were invited to retrain their systems on the groups; only a handful of participants chose to do this, and in the end the results were uniformly only slightly better than training on the fine-grained senses with coarse-grained scoring.

Table 2 shows the system performance on just the verbs of the lexical sample task. For comparison we ran several simple baseline algorithms that had been used in SENSEVAL-1, including RANDOM, COMMONEST, LESK, LESK-DEFINITION, and LESK-CORPUS (Kilgarriff and Rosenzweig, 2000). In contrast to SENSEVAL-1, in which none of the competing systems performed significantly better than the highest baseline (LESK-CORPUS), the best-performing systems this time performed well above the highest baseline.

Overall, the performance of the systems was much lower than in SENSEVAL-1. Several factors may have contributed to this. In addition to the use of fine-grained WordNet senses instead of the smaller Hector sense inventory from SENSEVAL-1, most of the verbs included in this task were chosen specifically because we expected them to be difficult to tag. There was also generally less training data made available to the systems (ignoring outliers, there were on average twice as many training samples for each verb in SENSEVAL-1 as there were in SENSEVAL-2). Table 3 shows the correspondence between test data size (half of training data size), entropy, and system performance for each verb.

4 Annotating the Gold Standard

The annotators made every effort to match the target word to a WordNet sense both syntactically and semantically, but sometimes this could not be done. Given a conflict between syntax and semantics, the annotators opted to match semantics. For example, the word "train" has an intransitive sense ("undergo training or instruction in preparation for a particular role, function, or profession") as well as a related (causative) transitive sense ("create by training and teaching"). Instances of "train" that were interpreted as having a dropped object were tagged with the transitive sense even though the overt syntax did not match the sense definition.

Some sentences seemed to fit equally well with two different senses, often because of am-

System	P	R
agirre3-ehu-dlist-best	0.846	0.229
magnini-irst-eng-sample	0.660	0.138
kunlp-	0.576	0.576
jhu-english-JHU-final (*)	0.566	0.566
SMUls-	0.563	0.563
LIA-Sinequa-Lexsample	0.535	0.535
manning-cs224n	0.523	0.523
agirre3-ehu-dlist-all	0.514	0.493
talp-TALP	0.513	0.513
umcp-englishl-	0.494	0.493
jhu-english-JHU-ENGLISH	0.489	0.489
montoyo-Univ.-Alicante-System	0.486	0.480
jhu-english-JHU	0.485	0.485
tdp1-duluth3	0.465	0.465
tdp1a-duluthC	0.453	0.453
tdp1-duluth5	0.450	0.450
tdp1-duluth4	0.446	0.446
baseline-lesk-corpus	0.445	0.445
tdp1-duluth2	0.440	0.440
tdp1a-duluthA	0.439	0.439
tdp1-duluth1	0.437	0.437
tdp1a-duluthB	0.404	0.404
baseline-commonest	0.403	0.403
david-fal-UNED-LS-T	0.388	0.387
david-fal-UNED-LS-U	0.288	0.287
Haynes-IIT2	0.233	0.232
Haynes-IIT1	0.220	0.220
Kenneth-Litkowski-clr-ls	0.218	0.218
Haynes-IIT2 (*)	0.199	0.192
Haynes-IIT1 (*)	0.193	0.186
baseline-lesk	0.181	0.181
michael-oakes.suss2	0.094	0.094
baseline-lesk-def	0.088	0.088
baseline-random	0.085	0.085

Table 2: System precision (P) and recall (R) for English verb lexical sample task (fine-grained scores); (*) indicates system results that were submitted after the SENSEVAL-2 workshop and official deadline.

biguous context; others did not fit well under any sense. One of the solutions employed in these cases was the assignment of multiple sense tags. The taggers would choose two senses (on rare occasions, even three) that they felt made an approximation of the correct sense when used in combination. Sometimes this strategy was also used in arbitration, when it was decided that neither tagger's tag was better than the other. The taggers tried to use this strategy sparingly and chose single tags whenever possible.

Often, a particular verb yielded multiple in-

Verb	Size	Entropy	Fine	Coarse
ferret	1	0.00	0.913	0.913
collaborate	30	0.44	0.898	0.898
wander	50	0.96	0.619	0.786
face	93	1.09	0.690	0.785
replace	45	1.62	0.471	0.860
use	76	1.68	0.558	0.682
begin	280	1.76	0.625	0.625
treat	44	2.10	0.453	0.543
live	67	2.35	0.455	0.476
match	42	2.35	0.398	0.620
train	63	2.60	0.394	0.492
drift	32	2.77	0.327	0.354
dress	59	2.89	0.434	0.679
serve	51	3.02	0.404	0.445
drive	42	3.03	0.308	0.528
leave	66	3.06	0.317	0.428
develop	69	3.17	0.301	0.456
see	69	3.28	0.278	0.317
wash	12	3.31	0.343	0.535
work	60	3.54	0.303	0.442
keep	67	3.62	0.336	0.353
call	66	3.68	0.246	0.457
play	66	3.80	0.323	0.345
find	68	3.81	0.178	0.285
carry	66	3.97	0.279	0.332
strike	54	4.06	0.248	0.331
pull	60	4.24	0.255	0.414
draw	41	4.60	0.195	0.264
turn	67	4.79	0.216	0.327

Table 3: Test corpus size, entropy (base 2) of tagged data, and average system recall for each verb, using fine-grained and coarse-grained scoring.

stances of what was clearly a salient sense, but one not found in WordNet. One of the results was that sentences that should have received a clear sense tag ended up with something rather ad hoc, and often inconsistent. One of the most notorious examples was "call," which had no sense that fit sentences like "The restaurant is called Marrakesh." WordNet contains some senses related to this one. One sense refers to the bestowing of a name; another to informal designations; another to greetings and vocatives. But there is no sense in WordNet for simply stating something's name without additional connotations, and the gap possibly caused some inconsistencies in the annotation. All these senses belonged to the same group, and if the annotators had been allowed to tag with the more general group sense, there may

have been less inconsistency.

It has been well-established that sense-tagging is a very difficult task (Kilgarriff, 1997; Hanks, 2000), even for experienced human taggers. If the sense inventory has gaps or redundancies, or if some of the sense glosses have ambiguous wordings, choosing the correct sense can be all but impossible. Even if the annotator is working with a very good entry, unforeseen instances of the word always arise.

The degree of polysemy does not affect the relative difficulty of tagging, at least not in the way it is often thought. Very polysemous words, such as "drive," are not necessarily harder to tag than less polysemous words like "replace." The difficulty of tagging depends much more on other aspects of the entry and of the word itself. Often very polysemous words *are* quite difficult to tag, because they are more likely to be underspecified or occur in novel uses; however, "replace," with four senses, proved a difficult verb to tag, while "play," with thirty-five senses, was relatively straightforward.

In many ways, the grouped senses are very helpful for the sense-tagger. Grouping similar senses allows the sense-tagger to study side-by-side the senses that are perhaps most likely to be confused, which is helpful when the differences between the senses are very subtle. However, it would be a poor idea to attempt to tag a corpus using *only* the groups, and not the finer sense distinctions, because often some of the senses included in a group will have some properties that the others do not; it is always better to make the finest distinction possible and not just assign the same tag to everything that seems close.

Inter-annotator agreement figures for the human taggers are quite low. However, in some respects they are not quite as low as they seem. Some of the apparent discrepancies were simply the result of a technical error: the annotator accidentally picked the wrong tag, perhaps choosing one of its neighbors. Other differences resulted from the sense inventories themselves. Sometimes the taggers interpreted the wording of a given sense definition in different ways, which caused them to choose different tags, but does not entail that they had interpreted the instances differently; in fact, discussion of such cases usually revealed that the taggers had interpreted the instances themselves in the sam way. Additional apparent discrepancies resulte from the various strategies for dealing with case in which there was no single proper sense i WordNet. This was the case when an instanc in the corpus was underspecified so as to al low multiple appropriate interpretations. Thi resulted in (a) multiple tags by one or bot taggers, and (b) each tagger making a differ ent choice. Here, again, the taggers often hac the same interpretation of the instance itself but because the sense inventory was insufficien for their needs, they were forced to find differen strategies. Sometimes, in fact, one tagger would double-tag a particular instance while the second tagger chose a single sense that matched one of the two selected by the first annotator. This is considered a discrepancy for statistical purposes, but clearly reflects similar interpretations on the part of the annotators.

In the most recent evaluation, with two new annotators tagging against the Gold Standard, the best fine-grained agreement figures for verbs were in the 70's, similar to Semcor figures. However, when we used the groupings to do a more coarse-grained evaluation, and counted a match between a single tag and a member of a double tag as correct, the human annotator agreement figures rose to 90%.

5 Acknowledgments

Support for this work was provided by the National Science Foundation (grants NSF-9800658 and NSF-9910603), DARPA (grant 535626), and the CIA (contract number 2000*SO53100*000). We would also like to thank Joseph Rosenzweig for building the annotation tools, and Susanne Wolff for contribution to the manual annotation.

References

Patrick Hanks. 2000. Do word meanings exist? *Computers and the Humanities*, 34(1-2), April. Special Issue on SENSEVAL.

A. Kilgarriff and J. Rosenzweig. 2000. Framework and results for English SENSEVAL. *Computers and the Humanities*, 34(1-2), April. Special Issue on SENSEVAL.

Adam Kilgarriff. 1997. I don't believe in word senses. *Computers and the Humanities*, 31(2).

Sensiting inflectionality: Estonian task for SENSEVAL-2

Neeme Kahusk and **Heili Orav** and **Haldur Õim**
University of Tartu
Research Group of Computational Linguistics
Tiigi 78, 50410 Tartu, Estonia
{nkahusk,horav,hoim}@psych.ut.ee

Abstract

This paper describes the all-word sense disambiguation task provided by Estonian team at SENSEVAL-2. About 10,000 words are manually disambiguated according to Estonian WordNet word senses. Language-specific problems and lexicon features are discussed.

1 Introduction

We got interested in word sense disambiguation (WSD) for two reasons. First, already a couple of years ago it was evident that WSD is becoming one of the new "hot" topics in computational linguistics and language engineering as our knowledge of how to handle semantic parameters of texts and semantic features of words in texts increased. The second reason was purely practical. Since 1996 we have been involved in a large project of building a semantic database of Estonian; participating in the EuroWordNet project has been a part of it (but a very important part, of course). The main source of building this database have been different corpora of Estonian, and in working with corpora the question of whether we are dealing with different meanings of a word in case of its concrete occurrences or not arises constantly. So we got interested in the possibility to use some objective methods here.

Our task was all-words task. This choice is explained with our "practical" interests explained above.

A large amount of work was done to provide training data where disambiguation was done manually. The same kind of work had to be done with test data, of course. The description of this work is given below. Let us note already here that this work appeared to be very useful and informative for us as builders of Estonian WordNet (EstWN).

And let us stress that this was our first attempt of WSD at all.

2 Corpora and lexicon

The test and training texts come from Corpus of the Estonian Literary Language (CELL), the 1980-s. We used this part of the corpus, that was morphologically disambiguated, initially for the syntactic analysis.

The morphological analysis was made with ESTMORF (Kaalep, 1997). Lemma and word class in the output of the program are relevant to our task, but it is impossible to get them without morphological disambiguation, because of frequent homonymy among word forms.

All training texts and most of test texts (5 of 6 total) are fiction. One of the test texts is from newspaper. Six training and six test files provided for the task contain about 2000 tokens each. More information about the texts used in the task is in Table 1.

Table 1: Statistics on training and test corpora

Corpus	Training	Test
Total words	12162	11440
Words to disambiguate	5854	5650
of them being		
verbs	2431	2191
nouns	3423	3459

2.1 Lexicon

The Estonian part of EuroWordNet[1] served as the lexicon. Like other wordnets, EstWN is a lexical-semantic database, the basic unit of which is concept. Concepts are represented as synonym sets (synsets) that are linked to each other by semantic relations. The description of

[1] http://www.hum.uva.nl/~ewn/

EstWN is given in the final document of EuroWordNet (Vider et al., 1999).

EstWN is supposed to cover the Estonian base vocabulary in its initial version. The base vocabulary will be determined by statistical analysis of the reference corpus. Even so it is not always easy (nor appropriate) to stop encoding words with frequencies below a certain threshold. For this reason we expect EstWN to cover more than just the base vocabulary.

Still the EstWN is rather small, there were 9436 synsets, 13277 words and 16961 senses (literals) in it when the disambiguation was done. That makes about 1.28 senses per word as average.

Most of synsets are connected with hyperonym-hyponym relations building corresponding hierarchies.

2.2 Procedure

Four linguists disambiguated the texts, each text was disambiguated by two persons. Only nouns and verbs were disambiguated, as entering adjectives into EstWN is in the very beginning. The sense number was marked according to sense number in EstWN. If the word was missing from the EstWN, "0" was marked as sense number, and if the word was in EstWN, but missed the appropriate sense, "+1" was marked.

If inconsistencies were met, they were discussed until agreement was achieved. On about 28% of the cases the disambiguators had different opinions.

One of the problems that the disambiguators ran into concerned dividing words into different senses in EstWN. It turned out as over-differentiation—word meaning marked as too specific, or over-generalisation—word meaning marked as too general.

2.3 How much the lexicon covers

Not all senses found in EstWN are represented in texts. Maximum number of senses per word found in texts is 13. This is more than appropriate senses in lexicon (see Table 3), but we must remeber about the "+1" that disambiguators had, if they found that there are not enough meanings in EstWN. Table 2 describes distribution of senses in usage and Table 3 shows the top of lemmas according to number of senses.

Table 2: Distribution of lemmas according t number of senses in texts

Corpus	Training	Test
Total number of lemmas	2340	2268
Number of lemmas not in lexicon	819	948
Number of lemmas with 1 sense in texts	2040	2003
Lemmas with 2 senses in texts	215	183
Lemmas with 3 senses in texts	51	50
Lemmas with 4 senses in texts	17	17
Lemmas with more than 4 senses in texts	17	15

Table 3: Comparison of richest words in sense

POS	No of senses in text	Lemma	No of senses in lexicon
verb	13	saama	12
verb	10	pidama	12
noun	10	asi	11
verb	9	olema	9
verb	9	käima	23
verb	7	võtma	7
verb	7	panema	11
verb	7	nägema	7
verb	7	minema	17
verb	7	leidma	8
noun	7	elu	7

It would be the best, if all words to disambiguate were in the lexicon with all their possible meanings. Apparently this presumption is not met.

The number of compounds in Estonian is indefinite. It is quite easy for a writer to invent new compounds that are not in any dictionary, but nevertheless are easily understood by readers. That is one reason, why there are so many sense numbers "0" in the texts. About 46 % of words that are not in EstWN, are compounds.

Another remarkable class of words not in lexicon are proper names, as there are no proper names in EstWN. There are 17.5 % of words proper names.

If we will postpone phrasal verbs and some strange words that contain hyphens (about

7 %), it leaves us with about half thousand words to check why they are not in EstWN.

But why are there missing senses (tagged with "+1")? The reason is simply historical: such words were included into EstWN as synonyms of some base vocabulary word and the other senses of them are not considered yet.

2.4 Phrases and multi-word units

The initial format of text was as it came from ESTMORF and semantic disambiguation: every word on separate line, followed by an aditional line of morphological analysis and sense number, with multi-word phrase marked if word was part of it. The task to convert into Senseval XML format seemed trivial at first, but phrases turned out to be problematic. Unfortunately enough, all the story about phrases is concerning the training corpus only, because in test corpus the multi-word phrases were unmarked.

Estonian is a flective language with a free word order and that makes it complicated to figure out all phrases. The elements of a phrase can be scattered around the sentence in an unpredictable order.

In the initial texts, the disambiguators marked down the whole phrase on the line where the phrase occured. They were not told to mark it on each line, where the non-disambiguatable parts of the phrase were, and it happened that the phrase was not marked on the line, where the head of the phrase was. The algorithm of calculating head or satellite took into account the part of speech and the form. For verb phrases, if both components were verbs, declinable form of verb infinitive was marked as satellite. For noun phrases, substantive makes head and adjective satellite. If both words are substantives, head is the second one... well, mostly.

However, it is known that expressions tend to contain frozen forms, including inflectional endings. For example, one may not say "*Human Right" or "*Humans Right". "Human Rights" is the only correct expression and should be added into thesauri in such form. Phrasal verbs like "ära maksma" (to pay off) and idiomatic verbal expressions like "end tükkideks naerma" (to laugh oneself into pieces) represent a situation that is different from the occasion described above: the verb part may inflect freely, but the other word(s) are frozen forms. Hereby, even if we have determined what is phrase or collocational multi-word unit, we still have a question— are they commonly used and should we add them into the lexicon.

Multiword expressions are included into EstWN if they build up a conceptual unit and are commonly used as lexical units.

3 Results

There were two systems to solve the task on Estonian. The results are in Table 4. Table 5 shows the recall and precision of the COMMONEST baseline

Table 4: Estonian all-words fine-grained scoring results

System	Precision	Recall	Attempted
JHU	0.67	0.67	100
est-semyh	0.66	0.66	100

Table 5: COMMONEST baseline for Estonian all-words task

Data	Recall	Precision
Overall	0.85	0.73
Polysemous	0.69	0.51

As this is the first attempt to disambiguate Estonian nouns and verbs in text, there is no comparison data. These results will set the level that future systems will try to outgo.

4 Conclusions

Results of WSD of corpus texts turned to be a good way to add missing synsets and senses into our wordnet. There were significant inconsistencies in opinions of these people, who disambiguated the texts. This shows us the most problematic entries in EstWN, the need to reconsider the borders of meaning of some concepts. By now, the last version of EstWN contains 9524 synsets, 13344 words and 17076 senses.

For an inflectional language like Estonian, morphological analysis is extremely important and morphological and semantic disabiguation can help each other.

References

H.-J. Kaalep. 1997. An estonian morphological analyser and the impact of a corpus on its development. *Computers and the Humanities*, 31:115–133.

K. Vider, L. Paldre, H. Orav, and H. Õim. 1999. The Estonian Wordnet. In C. Kunze, editor, *Final Wordnets for German, French, Estonian and Czech.* EuroWordNet (LE-8328), Deliverable 2D014.

The Italian Lexical Sample Task

Francesca BERTAGNA
Consorzio Pisa Ricerche
Via S. Maria 40
56100 Pisa, Italy,
f.bertagna@ilc.pi.cnr.it

Claudia SORIA
Istituto di Linguistica Computazionale-CNR
Via Moruzzi 1
56100 Pisa, Italy,
soria@ilc.pi.cnr.it

Nicoletta CALZOLARI
Istituto di Linguistica Computazionale-CNR
Via Moruzzi 1
56100 Pisa, Italy,
glottolo@ilc.pi.cnr.it

Abstract

In this paper we give an overall description of the Italian lexical sample task for SENSEVAL-2, together with some general reflections about on the one hand the overall task of lexical-semantic annotation and on the other about the adequacy of existing lexical-semantic reference resources.

Introduction

In this paper we give an overall description of the Italian lexical sample task for SENSEVAL-2. In the first two sections, the corpus and reference lexicon used are illustrated; the last section contains some general reflections on the basis of the Senseval experience about on the one hand, the overall task of lexical-semantic annotation and on the other, about the adequacy of existing lexical-semantic reference resources.

Dictionary and Corpus

The dictionary and corpus used for the Italian lexical sample task were provided by the resources developed in the framework of the SI-TAL project[1]. The data had not been adapted in order to be used for the Senseval task, apart from the necessary format conversions. A common encoding format (XML) proved to facilitate re-use and sharing of the data.

The lexical sample corpus

The Italian lexical sample corpus (test data only) consisted of about 3900 instances for 83 lexical entries (46 nouns, 21 verbs, and 16 adjectives), with an average of 47 contexts per entry.
The lexical samples were taken from the SI-TAL Italian Syntactic-Semantic Treebank (ISST[2]), which was still under development when the Senseval task was organized. This fact implied as a main disadvantage of the ISST material that corpus instances were associated with very little context. For each instance, the context corresponded to the sentence containing the target word and in our experience sometimes this proved to be not enough for a WSD task.
The ISST consists of two sub-components: a generic and a domain-specific (financial) corpus, of about 215,000 and 90,000 tokens, respectively. The annotated material comprises instances of newspaper articles, representing everyday journalistic Italian language. As far as annotation is concerned, the ISST has a three-level structure: two levels of syntactic annotation (a constituency-based and a functional-based annotation level) and a lexical-semantic level of annotation. ISST is supposed to be used in different types of applications, ranging from training of grammars and sense disambiguation systems, to the evaluation of language technology systems.
For its use in the SENSEVAL-2 task, only the semantic annotation was used, even if it is

[1] SI-TAL ('Integrated System for the Automatic Treatment of Language') is a National Project, coordinated by Antonio Zampolli at the 'Consorzio Pisa Ricerche' and involving several research centers in Italy, aiming at developing large linguistic resources and software tools for the Italian written and spoken language processing.

[2] See Montemagni et al. (2000a) and Montemagni et al. (2000b).

conceivable that a system could make use of the syntactic information as well.

In the ISST, this was performed manually using the ItalWordNet lexicon (henceforth IWN, see Roventini et al. 2000) as a reference resource (see below for a description). Semantic annotation consisted in assigning to each full word or sequence of words corresponding to a single unit of sense (such as compounds, idioms, etc.) a given sense number (referring to a specific synset) taken from IWN, plus specific features created for the annotation task to account for idioms, compounds and multi-words, figurative uses, evaluative suffixation, foreign words, proper nouns and titles, among the others. From this point of view, the semantic annotation of the corpus enriches the information available in the lexical resource.

However, in order to comply with the SENSEVAL-2 lexical sample format, the only semantic information used was the sense number of ISST, corresponding to the sense number of IWN synset variants, while the supplementary features had to be discarded. This fact obviously resulted in a loss of the overall semantic information available.

For instance, the semantic annotation gave no information about the specific domain or about possible metaphoric senses.

Although the original ISST contained multiwords expressions, no one of them was included in the Senseval lexical sample.

The selection of the lemmas has been carried out starting from the analysis of part of the words chosen for the English lexical sample, since we wanted to share a minimal overlapping core with the English list, in order to make the final results more comparable in a multilingual perspective". At the end, the overlap between English and Italian consisted of only 8 entries, unfortunately.[3]

The criteria for the selection were the polysemy of the word in the lexicon, the frequency, and the actual occurrence in the annotated resource with more than one meaning.

The average polysemy was of 5 senses per word (5 for the nouns subset, 6 for the verbs and 3 for the adjectives).

The average frequency turned out to be rather low, since the Italian treebank from which the lexical sample was extracted was still not complete and we had to select the most frequent words with at least two senses in the lexicon and used at least in two of their senses in the annotated corpus. This led to select mainly words with quite high polysemy and rather generic senses. For instance, only 12 of the 46 nouns had also concrete sense.

More importantly, since we had at our disposal a rather low number of occurrences, no training data were available for the Italian task. This makes the results for the Italian task hardly comparable with those which used similarly structured data, such as the Spanish, Swedish, Basque and Korean tasks as all of them had training data available. This is particularly significant in evaluating the results for the Italian task if we consider that the two systems participating to the task were supervised and needed sense-tagged training instances of each word. For the next Senseval, a larger annotated corpus will be available and hence a training corpus will be provided.

The reference lexicon.

As it was said before, the occurrences provided for the WSD lexical sample task were annotated according to the lexical-semantic database ItalWordNet, developed within the framework of the SI-TAL Project[4].

ItalWordNet is an extension of the Italian wordnet built during the EuroWordNet project (Vossen, 1999).

The IWN database is constituted by:

i) a generic wordnet containing about 64,000 word senses corresponding to about 49,000 synsets;
ii) a (generic) Interlingual-Index (ILI) which is an unstructured version of WordNet 1.5, also used in EWN to link wordnets of different languages;
iii) a terminological wordnet, containing about 5,000 synsets of the economic-financial domain;
iv) a terminological ILI, to which the terminological wordnet is linked;
v) the Top Ontology, a hierarchy of language-independent concepts, built within EWN and partially modified in IWN to account for adjectives (Alonge et al., 2000). Via the ILIs, all the concepts in the generic and specific wordnets are directly or indirectly linked to the Top Ontology;
vi) the Domain Ontology, containing a set of domain labels. Via the ILIs, all the concepts in the generic and specific wordnets are

[3] The entries that are in common were: *arte-art, chiamare-call, colpire-hit, giocare/gioco-play, lavorare/ lavoro-work, senso-sense, trovare-find.*

[4] ItalWordNet is a joint effort between the Consorzio Pisa Ricerche and IRST (Istituto per la Ricerca Scientifica e Tecnologica), Trento, Italy.

directly or indirectly linked to the Domain Ontology.

For the 83 lexical entries we provided to the competitors a hierarchical basic data structure: all the senses of the lemma organized in groups of synonyms (synset) plus their direct hyperonyms and a brief Italian definition.
We also provided a set of semantic relations (belonging to the set of Euro(/Ital)WordNet relations: hyponymy, role/involved, holo/meronymy, derivational relations etc.), but we didn't supply the target entries of the relations (and all their semantic and ontological information) since we provided only a portion of the whole wordnet[5].
All the entries were provided with equivalence relations to at least one record of the EuroWordNet Interlingual Index and with the link to the EuroWordNet Top Concepts.
The entries have been used as they were in the wordnet, without making any adjustment specific for the task at hand. Although the domain information, so useful in a WSD task, is available in the model (only with few labels), none of the provided entries had it, because it has not been systematically codified and also because almost all the entries were quite generic. This was a main disadvantage for at least one of the two systems competing for the Italian Senseval task.
We are now in the process of evaluating whether a linking between ItalWordNet and SIMPLE[6] would be feasible; such a linking could allow ItalWordNet to inherit the rich domain information available in the SIMPLE database.
We didn't consider the POS-tagging a part of the task and we provided as corpus instances only those with the same POS as the previously selected lexical items, i.e. we eliminated occurrences of homographs belonging to different parts of speech.

Results for the Italian lexical sample task

Only two systems took part in the Italian task, namely the IRST and JHU systems.
The results for fine, mixed and coarse-grained WSD are illustrated in the following tables:

System	Precision	Recall	Attempted
IRST	0.406	0.389	95.783%
JHU	0.353	0.353	100%

Table 1: Fine-grained scoring

[5] The whole of the new version of IWN could be obtained through ELRA.

[6] See Lenci et al. (2000)

System	Precision	Recall	Attempted
IRST	0.482	0.461	95.783%
JHU	0.421	0.421	100%

Table 2: Mixed-grained scoring

System	Precision	Recall	Attempted
IRST	0.483	0.463	95.783%
JHU	0.423	0.423	100%

Table 3: Coarse-grained scoring

The low scores are mainly due to the lack of training data and of domain information. It is also possible that for some entries of the lexicon the subtlety of sense distinctions contributed to low performance of the systems, as it's shown by better results obtained with the coarse-grained scoring.

General remarks

Starting from the SENSEVAL-2 experience, we would like to make a few general remarks, both about the adequacy of available lexical-semantic reference resources for WSD tasks and about the overall task of lexical-semantic annotation.
One of the well-known problems of WordNet is the fine-grainedness of its entries in terms of sense distinction. This is true also for the Italian net, even if maybe at a lower level: a brief analysis of the entries highlights the presence of some very subtle distinctions among the senses. Actually, during the SI-TAL project, corpus annotators set up a specific annotation strategy for handling cases where synsets are numerous and reflect fine-grained sense distinctions not easily mappable to the corpus contexts. The strategy allowed the assignment of multiple senses connected through logical operators of conjunction (when IWN senses cannot be distinguished) vs. disjunction (when the ambiguous context does not allow a choice among the different IWN senses).
Nonetheless, in the Italian lexical sample used for Senseval, there are about only 140 cases of multiple key assignment out of about 3900 corpus instances.
This suggests that vague or too fine-grained distinctions are still unproblematic for humans, but may become problematic for machines. It could be useful to investigate what kind of sense distinctions are hardest for systems to make, and whether or not systems have problems with the same senses that human annotators have problems with.
When a stable version of the annotated resource is available, we will be able to start a more detailed analysis of the results of the annotation.

It will be possible, for example, to evaluate the impact of the presence of figurative/rhetorical nuances of a sense in the corpus or to consider the quality and types of the multiwords that, found in the corpus, have been proposed to the IWN lexicographers in order to have them added to the lexicon.

But, above all, by analysing the level of confidence in the sense assignment, it will be possible to evaluate the correctness/suitability of the sense distinction in those cases that generated doubts in the human annotators. This kind of analysis would be particularly useful under the perspective of the organization of future Senseval tasks.

Another issue to inquiry is whether the adoption of the wordnet model and the use of the synsets as information core can lead to a proliferation of word meanings according to the kind of synonyms which may replace a given word in a context[7]. Apart from this, however, it is a fact that use of wordnet or wordnet-like resources significantly correlates with an overall worsening in the performance of WSD systems compared with the previous results obtained using traditional dictionaries. This certainly is an issue to reflect upon.

Other, more general considerations concern the issue of semantic annotation in general. It does not seem correct to talk about the "right sense distinction", and to think at the word sense as a task–independent information (Kilgarriff, 1997): the greater vs. lesser granularity depends also on the task/domain/situation and in principle there is no upper or lower limit to sense granularity.

It seems that there are areas of meaning that cannot be easily encoded at the lexical-semantic level of annotation: sense interpretation may require appeal to e.g. extra-linguistic (world) knowledge which cannot be encoded/captured at the lexical-semantic level of description. We refer here to metaphors even extended to entire sequences and not limited to the single word; to words acquiring a specific sense, strictly dependent on the context, that cannot be encoded at the lexical-semantic level; or to the complexity and variety of nuances implied e.g. by a verb, according to the type of direct object co-occurring with it. Not all these shifts of meaning can or must be captured through lexical-semant annotation.

[7] This is the case of the verb *dire* (to say/to tell) which has the following synsets, among others, in IWN:
dire, enunciare, proferire (utter, mouth, etc.)
spiegare, dire (explain, tell)
dire, far sapere (tell, let it be known).

References

Antonietta Alonge, Francesca Bertagna, Nicoleti Calzolari, Adriana Roventini, Antonio Zampol (2000) *Encoding information on adjectives in lexical-semantic net for computational application.* Proceedings of the 1st NAACL Meeting, Seattle, pp 42-49.

Adam Kilgarriff (1997) *"I don't believe in wor senses"*. ITRI-97-12.

Alessandro Lenci, Nuria Bel, Federica Busa, Nicolett Calzolari, Elisabetta Gola, Monica Monachini Antoine Ogonowsky, Ivonne Peters, Wim Peters Nilda Ruimy, Marta Villegas, Antonio Zampoll (2000) *SIMPLE: A General Framework for th Development of Multilingual Lexicons*. Internationa Journal of Lexicography, XIII (4): pp. 249-263.

Simonetta Montemagni, Francesco Barsotti, Marc Battista, Nicoletta Calzolari, Ornella Corazzari Antonio Zampolli, Francesca Fanciulli, Maria Massetani, Remo Raffaelli, Roberto Basili, Maria Teresa Pazienza, Dario Saracino, Fabio Zanzotto, Nadia Mana, Fabio Pianesi, Rodolfo Delmonte (2000) *The Italian Syntactic-Semantic Treebank: Architecture, Annotation, Tools and Evaluation.* LINC-2000, Luxembourg.

Simonetta Montemagni, Barsotti Francesco, Battista Marco, Calzolari Nicoletta, Corazzari Ornella, Lenci Alessandro, Zampolli Antonio, Fanciulli Francesca, Maria Massetani, Remo Raffaelli, Roberto Basili, Maria Teresa Pazienza, Dario Saracino, Fabio Zanzotto, Nadia Mana, Fabio Pianesi, Rodolfo Delmonte, (2000) *Building the Italian Syntactic-Semantic Treebank.* In "Building and Using Syntactically Annotated Corpora", A. Abeillé, ed., Language and Speech Series, KLUWER, Dordrecht.

Adriana Roventini, Antonietta Alonge, Nicoletta Calzolari, Bernardo Magnini, Francesca Bertagna (2000) *ItalWordNet: a large semantic database for Italian*. Proceedings of the 2nd International Conference on Language Resources and Evaluation, Athens, Greece.

Piek Vossen (ed.) (1999) *EuroWordNet General Document*, http://www.hum.uva.nl/~ewn.

SENSEVAL-2 Japanese Dictionary Task

Kiyoaki Shirai
School of Information Science, Japan Advanced Institute of Science and Technology
kshirai@jaist.ac.jp

Abstract

This paper reports an overview of the SENSEVAL-2 Japanese dictionary task. It was a lexical sample task, and word senses are defined according to a Japanese dictionary, the Iwanami Kokugo Jiten. The Iwanami Kokugo Jiten and a training corpus were distributed to all participants. The number of target words was 100, 50 nouns and 50 verbs. One hundred instances of each target word were provided, making for a total of 10,000 instances for evaluation. Seven systems of three organizations participated in this task.

1 Introduction

In SENSEVAL-2, there are two Japanese tasks, a translation task and a dictionary task. This paper describes the details of the dictionary task.

First of all, let me introduce an overview of the Japanese dictionary task. This task is a lexical sample task. Word senses were defined according to the Iwanami Kokugo Jiten (Nishio et al., 1994), a Japanese dictionary published by Iwanami Shoten. It was distributed to all participants as a sense inventory. Training data, a corpus consisting of 3,000 newspaper articles and manually annotated with sense IDs, was also distributed to participants. For evaluation, we distributed newspaper articles with marked target words as test documents. Participants were required to assign one or more sense IDs to each target word, optionally with associated probabilities. The number of target words was 100, 50 nouns and 50 verbs. One hundred instances of each target word were provided, making for a total of 10,000 instances.

In what follows, Section 2 describes details of data used in the Japanese dictionary task. Section 3 describes the process to construct the gold standard data, including the analysis of inter-tagger agreement. Section 4 briefly introduces participating systems and their results. Finally, Section 5 concludes this paper.

2 Data

In the Japanese dictionary task, three data were distributed to all participants: sense inventory, training data and evaluation data.

2.1 Sense Inventory

As described in Section 1, word senses are defined according to a Japanese dictionary, the Iwanami Kokugo Jiten. The number of headwords and word senses in the Iwanami Kokugo Jiten is 60,321 and 85,870, respectively.

Figure 1 shows an example of word sense descriptions in the Iwanami Kokugo Jiten, the sense set of the Japanese noun "MURI."

MURI
1. lack of reasonableness
   1-a. something not to be rational, not to be sensible [*kimi ga okoru no wa* ***MURI*** *mo nai* (It is natural for you to be angry)]
   1-b. to do something compulsorily [*sigoto no* ***MURI*** *de byouki ni naru* (I become ill from overwork)]

Figure 1: Sense set of "MURI"

As shown in Figure 1, there are hierarchical structures in word sense descriptions. For example, word sense 1 subsumes 1-a and 1-b. The number of layers of hierarchy in the Iwanami Kokugo Jiten is at most 3. Word sense distinctions in the lowest level are rather fine or subtle. Furthermore, a word sense description sometimes contains example sentences including a headword, indicated by italics in Figure 1.

The Iwanami Kokugo Jiten was provided to all participants. For each sense description, a

corresponding sense ID and morphological information were supplied. All morphological information, which included word segmentation, part-of-speech (POS) tag, base form and reading, was manually post-edited.

2.2 Training Data

An annotated corpus was distributed as the training data. It was made up of 3,000 newspaper articles extracted from the 1994 Mainichi Shimbun, consisting of 888,000 words. The annotated information in the training corpus was as follows:

- Morphological information

  The text was annotated with morphological information (word segmentation, POS tag, base form and reading) for all words. All morphological information was manually post-edited.

- UDC code

  Each article was assigned a code representing the text class. The classification code system was the third version (INFOSTA, 1994) of Universal Decimal Classification (UDC) code (Organization, 1993).

- Word sense IDs

  Only 148,558 words in the text were annotated for sense. Words assigned with sense IDs satisfied the following conditions:

  1. Their POSs were noun, verb or adjective.
  2. The Iwanami Kokugo Jiten gave sense descriptions for them.
  3. They were ambiguous, i.e. there are more than two word senses in the dictionary.

  Word sense IDs were manually annotated. However, only one annotator assigned a sense ID for each word.

2.3 Evaluation Data

The evaluation data was made up of 2,130 newspaper articles extracted from the 1994 Mainichi Shimbun. The articles used for the training and evaluation data were mutually exclusive. The annotated information in the evaluation data was as follows:

- Morphological information

  The text was annotated with morphological information (word segmentation, POS tag, base form and reading) for all words. Note that morphological information in the training data was manually post-edited, but not in the evaluation data. So participants might ignore morphological information in the evaluation data.

- UDC code

  As in the training data. each article was assigned a UDC code

- Word sense IDs (gold standard data)

  Word sense IDs were annotated manually for the target words [1]. Note that word sense IDs in the evaluation and training data were given in different ways: (1) a sense ID was assigned for each word by at least two annotators in the evaluation data, while by only one annotator in the training data, (2) only 10,000 instances in the articles were annotated with sense IDs in the evaluation data, while all words were annotated which satisfied the conditions described in 2.2 in the training data.

3 Gold Standard Data

Except for the gold standard data, the data described in Section 2 have been developed by Real World Computing Partnership (Hasida et al., 1998; Shirai et al., 2001) and already released to public domain [2]. On the other hand, the gold standard data was newly developed for the SENSEVAL-2. This section presents the process of preparing the gold standard data, and the analysis of inter-tagger agreement.

3.1 Sampling Target Words

When we chose target words, we considered the following:

- POSs of target words were either nouns or verbs.
- Words were chosen which occurred more than 50 times in the training data.

[1] They were hidden from participants at the contest.

[2] Notice that the training data had been released to the public before the contest began. This violated the SENSEVAL-2 schedule constraint that answer submission should not occur more than 21 days after downloading the training data.

Table 1: Number of Target Words

	D_a	D_b	D_c	all
nouns	**10** (9.1/1.19)	**20** (3.7/0.723)	**20** (3.3/0.248)	**50** (4.6/0.627)
verbs	**10** (18/1.77)	**20** (6.7/0.728)	**20** (5.2/0.244)	**50** (8.3/0.743)
all	**20** (14/1.48)	**40** (5.2/0.725)	**40** (4.2/0.246)	**100** (6.5/0.685)

(average polysemy / average entropy)

- The relative "difficulty" in disambiguating the sense of words was considered. Difficulty of the word w was defined by the entropy of the word sense distribution $E(w)$ in the training data. Obviously, the higher $E(w)$ was, the more difficult the WSD for w was.

  We set up three word classes, D_a ($E(w) \geq 1$), D_b ($0.5 \leq E(w) < 1$) and D_c ($E(w) < 0.5$), and chose target words evenly from them.

Table 1 reveals details of numbers of target words. Average polysemy (i.e. average number of word senses per headword) and average entropy are also indicated.

One hundred instances of each target word were selected from newspaper articles, making for a total of 10,000 instances.

3.2 Manual Annotation

Six annotators assigned the correct word sense IDs for 10,000 instances. They were not experts, but had knowledge of linguistics or lexicography to some degree. The process of manual annotating was as follows:

Step 1. Two annotators chose a sense ID for each instance separately in accordance with the following guidelines:

- Only one sense ID was to be chosen for each instance.
- Sense IDs at any layers in hierarchical structures could be assignable.
- The "UNASSIGNABLE" tag was to be chosen only when all sense IDs weren't absolutely applicable. Otherwise, choose one of sense IDs in the dictionary.

Table 2: Inter-tagger Agreement

	D_a	D_b	D_c	(all)
nouns	0.809	0.786	0.957	0.859
verbs	0.699	0.896	0.922	0.867
all	0.754	0.841	0.939	0.863

Step 2. If the sense IDs selected by 2 annotators agreed, we considered it to be a correct sense ID for an instance.

Step 3. If they did not agree, the third annotator chose the correct sense ID between them. If the third annotator judged both of them to be wrong and chose another sense ID as correct, we considered that all 3 word sense IDs were correct.

According to Step 3., the number of words for which 3 annotators assigned different sense IDs from one another was a quite few, 28 (0.3%).

Table 2 indicates the inter-tagger agreement of two annotators in Step 1. Agreement ratio for all 10,000 instances was 86.3%.

4 Results for Participating Systems

In the Japanese dictionary task, the following 7 systems of 3 organizations submitted answers. Notice that all systems used supervised learning techniques.

- Communications Research Laboratory and New York University (CRL1 ~ CRL4)

  The learning schemes were simple Bayes and support vector machine (SVM), and two kinds of hybrid models of simple Bayes and SVM.

- Tokyo Institute of Technology (Titech1, Titech2)

  Decision lists were learned from the training data. The features used in the decision lists were content words and POS tags in a window, and content words in example sentences contained in word sense descriptions in the Iwanami Kokugo Jiten.

- Nara Institute of Science and Technology (Naist)

  The learning algorithm was SVM. The feature space was reconstructed using Principle Component Analysis(PCA) and Independent Component Analysis(ICA).

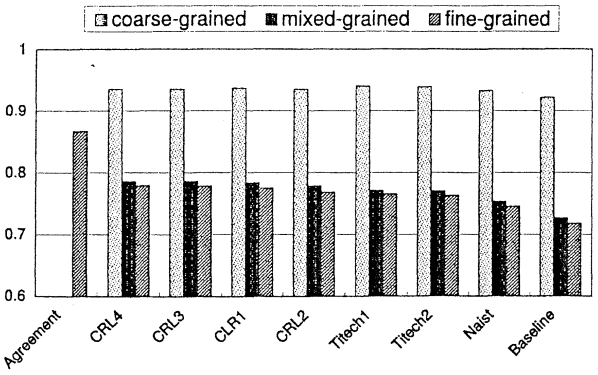

Figure 2: Results

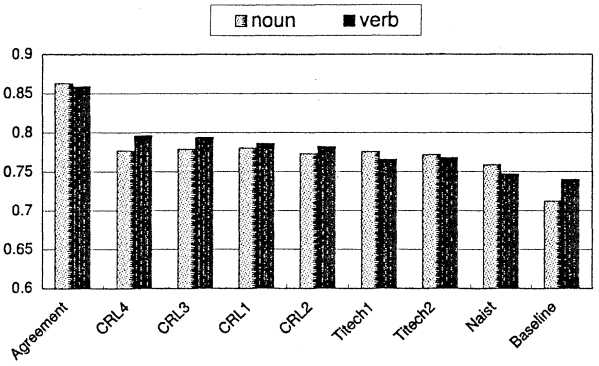

Figure 3: Mixed-grained scores for nouns and verbs

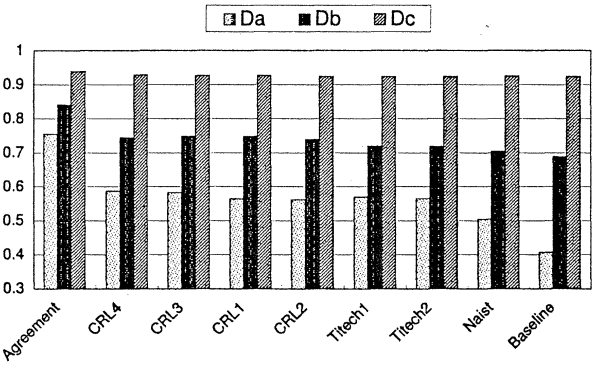

Figure 4: Mixed-grained scores for word classes

The results of all systems are shown in Figure 2. "Baseline" indicates the system which always selects the most frequent word sense ID, while "Agreement" indicates the agreement ratio between two annotators. All systems outperformed the baseline, and there was no remarkable difference between their scores (differences were 3 % at most).

Figure 3 indicates the mixed-grained scores for nouns and verbs. Comparing baseline system scores, the score for verbs was greater than that for nouns, even though the average entropy of verbs was higher than that of nouns (Table 1). The situation was the same in CRL systems, bu not in Titech and Naist. The reason why the av erage entropy was not coincident with the scor of the baseline was that the entropy of som verbs was so great that it raised the average er tropy disproportionately. Actually, the entrop of 7 verbs was greater than the maximum er tropy of nouns.

Figure 4 indicates the mixed-grained score for each word class. For word class D_c, ther was hardly any difference among scores of a systems, including Baseline system and Agree ment. On the other hand, appreciable differenc was found for D_a and D_b.

5 Conclusion

This paper reports an overview of th SENSEVAL-2 Japanese dictionary task. Th data used in this task are available on th SENSEVAL-2 web site. I hope this valuabl data helps all researchers to improve their WSI systems.

Acknowledgment

I wish to express my gratitude to Mainich Newspapers for providing articles. I would alsc like to thank Prof. Takenobu Tokunaga (Tokyc Institute of Technology) and Prof. Sadao Kuro hashi (University of Tokyo) for valuable advise about task organization, the annotators for con structing gold standard data, and all partici pants.

References

Koiti Hasida et al. 1998. The RWC text databases. In *Proceedings of the the firs International Conference on Language Re sources and Evaluation*, pages 457–462.

INFOSTA. 1994. *Universal Decimal Classifica tion.* Maruzen, Tokyo. (in Japanese).

Minoru Nishio, Etsutaro Iwabuchi, and Shizuc Mizutani. 1994. *Iwanami Kokugo Jiten Da Go Han.* Iwanami Publisher. (in Japanese).

British Standards Organization. 1993. *Guide to the Universal Decimal Classification (UDC).* BSI, London.

Kiyoaki Shirai et al. 2001. Text database with word sense tags defined by Iwanami Japanese dictionary. *SIG notes of Information Pro cessing Society of Japan*, 2001(9):117–122. (in Japanese).

SENSEVAL-2 Japanese Translation Task

Sadao Kurohashi
University of Tokyo
kuro@kc.t.u-tokyo.ac.jp

Abstract

This paper reports an overview of SENSEVAL-2 Japanese translation task. In this task, word senses are defined according to translation distinction. A translation Memory (TM) was constructed, which contains, for each Japanese head word, a list of typical Japanese expressions and their English translations. For each target word instance, a TM record best approximating that usage had to be submitted. Alternatively, submission could take the form of actual target word translations. 9 systems from 7 organizations participated in the task.

1 Introduction

In written texts, words which have multiple senses can be classified into two categories; homonyms and polysemous words. Generally speaking, while homonymy sense distinction is quite clear, polysemy sense distinction is very subtle and hard. English texts contain many homonyms. On the other hand, Japanese texts in which most content words are written by ideograms rarely contain homonyms. That is, the main target in Japanese WSD is polysemy, which makes Japanese WSD task setup very hard. What sense distinction of polysemous words is reasonable and effective heavily depends on how to use it, that is, an application of WSD.

Considering such a situation, in addition to the ordinary dictionary task we organized another task for Japanese, a translation task, in which word sense is defined according to translation distinction. Here, we set up the task assuming the example-based machine translation paradigm (Nagao, 1981). That is, first, a translation memory (TM) is constructed which contains, for each Japanese head word, a list of typical Japanese expressions (phrases/sentences) involving the head word and an English translation for each (Figure 1). We call a pair of Japanese and English expressions in the TM as a TM record. Given an evaluation document containing a target word, participants have to submit the TM record best approximating that usage.

Alternatively, submissions can take the form of actual target word translations, or translations of phrases or sentences including each target word. This allows existing rule-based machine translation (MT) systems to participate in the task, and we can compare TM based systems with existing MT systems.

For evaluation, we distributed newspaper articles. The number of target words was 40, and 30 instances of each target word were provided, making for a total of 1,200 instances.

2 Construction of Translation Memory

The translation memory (TM) was constructed in two steps:

1. By referring to the KWIC (Key Word In Context) of a target word, its typical Japanese expressions are picked up by lexicographers.
2. The Japanese expressions are translated by a translation company.

KWIC was made from the nine years volume of Mainichi Newspaper corpus. They are morphologically analyzed and segmented into phrase sequences, and then the 100 most frequent phrase uni-grams, bi-grams (two types; the target word is in the first phrase or the second phrase) and tri-grams (the target word is in the middle phrase) are provided to lexicographers (Figure 2).

無理 *muri*	
参加は無理だ	It is impossible to participate.
今から図書館の利用は無理だ	It is impossible to make use of the library in this hour.
今回の法案には無理がある	This bill is hard to pass.
彼が怒るのも無理はない	It is no wonder he got angry.
一番無理のない方法	the most natural way
無理を重ねる	to work too much
無理な話	unreasonable demand
無理な追い越し	passing by force
無理心中を図る	to commit a forced double suicide
...	...

Figure 1: An example of Translation Memory.

Phrase uni-gram	Phrase bi-gram		Phrase tri-gram
597 無理な	151 無理は ない。	19 ことには 無理が	7 ことには 無理が ある。
551 無理が	138 無理が ある。	14 とても 無理。	6 求めるのは 無理が ある。
416 無理やり	106 無理も ない。	13 ことは 無理と	5 ことには 無理からぬ 理由が
413 無理に	101 無理 なく	10 求めるのは 無理が	5 嘆くのも 無理は ない。
403 無理を	67 無理の ない	10 とても 無理」と	5 同署は 無理心中と みている。
351 無理。	56 無理が ある」と	9 いうのは 無理が	4 しても 無理は ない。
...	...	...	...

Figure 2: An example of KWIC (numbers indicate phrase frequency).

The lexicographers pick up a typical expression of the target word from the KWIC. If its sense is context-independently clear, the expression is adopted as it is. If its sense is not clear, some pre/post expressions are supplemented by referring original sentences in the newspaper corpus.

Then, we asked a translation company to translate the Japanese expressions. As a result, a TM containing 320 head words and 6920 records was constructed (one head word has 21.6 records on average). The average number of words of a Japanese expression is 4.5.

3 Gold Standard Data and the Evaluation of Translations

As a gold standard data of the task, 40 target words were chosen out of 320 TM words. Considering the possible comparison of the translation task and the dictionary task, 40 target words were fully overlapped with 100 target words of the dictionary task.

In the Japanese dictionary task, target words are classified into three categories according to the difficulty (difficult, intermediate, easy), based on the entropy of word sense distribution in the training data of the dictionary task(Shirai, 2001). 40 target words of the trans lation task consists of 20 nouns and 20 verbs: difficult nouns and verbs, 10 intermediate nour and verbs, and 5 easy nouns and verbs.

For each target word, 30 instances were ch sen from Mainichi Newspaper corpus (in tot 1,200 instances) and they are also overlappe with the dictionary task. Since the dictionar task uses 100 instances for each target wor the translation task used 1st, 4th, 7th, ... 90t instances of the dictionary task.

As a gold standard data, zero or more ap propriate TM records were assigned to each in stance by the same translation company. Ap propriate TM records were classified into th following three classes:

◎ : A TM record which can be used t translate the instance. POS, tense, plura singular, and subtle nuance do not nece sarily match.

○ : If the instance is considered alone, th English translation is correct, but usin the TM record in the given context is nc so good, for example, making very round about translation.

△ : If the instance is considered alone, the English translation is correct, but using the TM record in the given context is inappropriate.

Out of 1,200 instances, 34 instances (2.8%) were assigned no TM records (there was no appropriate TM record). To one instance, on average, 6.6 records were assigned as ◎, 1.4 records as ○, and 0.1 records as △, in total 8.1 records. If a system chooses a TM record randomly as an answer, the accuracy becomes 36.8% in case that all of ◎, ○ and △ records are regarded as correct, and 29.0% in case that only ◎ is regarded as correct (they are the baseline scores used in the next section).

In the gold standard data construction, 90 instances (9 words × 10 instances) were dealt with by two annotators doubly, and then their agreement were checked. For each instance one record is chosen randomly from annotator B's answers, and it was checked whether it is contained in annotator A's answers (annotator A made the whole gold standard data). The agreement was 86.6% in case that all of ◎, ○ and △ records are regarded as correct, and 80.9% in case that only ◎ is regarded as correct.

In the case that the submission is in the form of translation data, translation experts (the same company as constructed the TM and the gold standard data) were asked to rank the supplied translation ◎, ○ or ×. This evaluation does not pay attention to the total translation, but just the appropriateness of the target instance translation.

4 Result

In the Japanese translation task, 9 systems from 7 organizations submitted the answers. The characteristics of the systems are summarized as follows:

- AnonymX, AnonymY
  Commercial, rule-based MT systems.
- CRL-NYU (Communications Research Laboratory & New York Univ.)
  TM records are classified according to the English head word, and each cluster is supplemented by several corpora. The system returns a TM record when the similarity between a TM record and an input sentence is very high. Otherwise, it returns the English head word of the most similar cluster by using several machine learning techniques.
- Ibaraki (Ibaraki Univ.)
  A training data was constructed manually from newspaper articles, 170 instances for each target word. Features were collected in 7-word window around the target word, and decision list method was used for learning.
- Stanford-Titech1 (Stanford Univ. & Tokyo Institute of Technology)
  The system selects the appropriate TM record based on the character-bigram-based Dice's coefficient. It also utilized the context of the other target word instances in the evaluation text.
- AnonymZ
  A sentence (TM records for learning, and an input for testing) is morphologically analyzed and converted into a semantic tag sequence, and maximum entropy method was used for learning.
- ATR
  The system selects the most similar TM record based on the cosine similarity between context vectors, which were constructed from semantic features and syntactic relations of neighboring words of the target word.
- Kyoto (Kyoto Univ.)
  The system selects the most similar TM record by bottom-up, shared-memory based matching algorithm.
- Stanford-Titech2 (Stanford Univ. & Tokyo Institute of Technology)
  The system selects the appropriate TM record based on the case-frame-based similarity, using NTT Goi-Taikei thesaurus.

The results of all systems are shown in Figure 3. The left bar charts indicate the accuracy based on the lenient evaluation (◎, ○ and △ in TM selection and ◎ and ○ in MT are regarded as correct); the right bar charts indicate the accuracy based on the strict evaluation (◎ is only regarded as correct both in TM selection and MT). Note that since the TM does not have

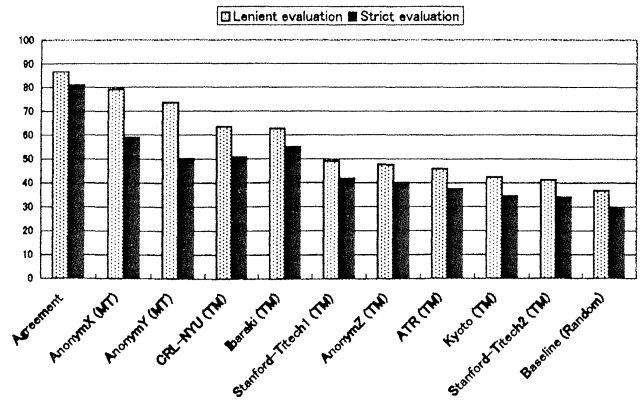

Figure 3: Result of the Japanese translation task.

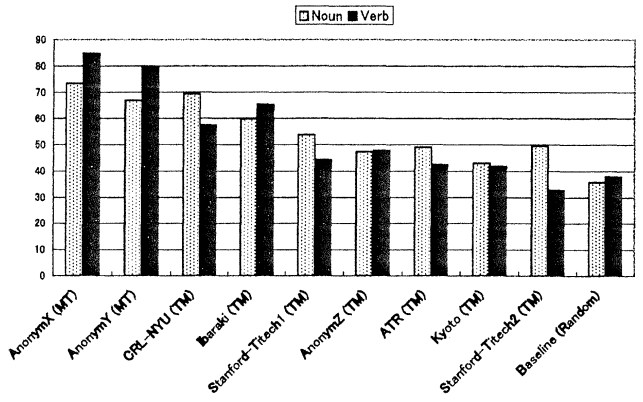

Figure 4: Scores for nouns and verbs.

a hierarchical structure, there is no evaluation options such as fine, coarse, and mixed.

Figure 4 shows scores for nouns and verbs separately, and Figure 5 shows scores for difficult/intermediate/easy words. Both of them were evaluated by the lenient criteria.

In these figures, "Agreement" and "Baseline" were as described in the previous section. When the system judges that there is no appropriate TM record for an instance, it can return "UNASSIGNABLE". In that case, if there is no appropriate TM record assigned in the gold standard data, the answer is regarded as correct.

Among TM selection systems, systems using some extra learning data outperformed other systems just using the TM. The comparison between TM selection systems and MT systems is not easy, but the result indicates the effectiveness of the accumulated know-how of MT systems. However, the performance of the best TM selection system is not so different from MT

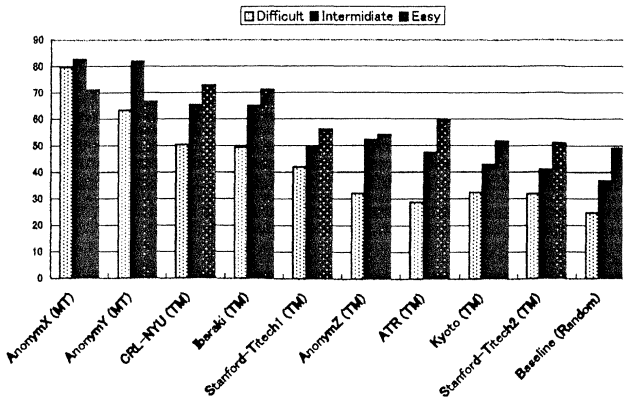

Figure 5: Scores for difficulty classes.

systems, which indicates the promising future of TM based techniques.

5 Conclusion

This paper described an overview of SENSEVAL-2 Japanese translation task. The data used in this task are available at SENSEVAL-2 web site. We hope this valuable data helps improve WSD and MT systems.

Acknowledgment

I wish to express my gratitude to Mainichi Newspapers for providing articles. I would also like to thank Prof. Takenobu Tokunaga (Tokyo Institute of Technology) and Prof. Kiyoaki Shirai (JAIST) and Dr. Kiyotaka Uchimoto (CRL) for their valuable advise about task organization, Yuiko Igura (Kyoto Univ.) and Intec Group Corp. for data construction, and all participants to the task.

References

Makoto Nagao. 1981. A framework of mechanical translation between Japanease and English by analogy priciple. In *Proc. of the International NATO Symposium on Artificia and Human Intelligence.*

Kiyoaki Shirai. 2001. SENSEVAL-2 Japanese dictionary task. In *Proceedings of the SENSEVAL-2 Workshop.*

Framework and Results for the Spanish SENSEVAL

German Rigau, Mariona Taulé, Ana Fernandez and **Julio Gonzalo**
g.rigau@lsi.upc.es, TALP Research Center, Universitat Politècnica de Catalunya
mtaule@lingua.fil.ub.es, CLiC, Universitat de Barcelona
ana.fernandez@uab.es, CLiC, Universitat Autònoma de Barcelona
julio@lsi.uned.es, GPLN, Universidad Nacional de Educación a Distancia

Abstract

In this paper we describe the structure, organisation and results of the SENSEVAL exercise for Spanish. We present several design decisions we taked for the exercise, we describe the creation of the gold-standard data and finally, we present the results of the evaluation. Twelve systems from five different universities were evaluated. Final scores ranged from 0.56 to 0.65.

1 Introduction

In this paper we describe the structure, organisation and results of the Spanish exercise included within the framework of SENSEVAL-2.

Although we closely follow the general architecture of the evaluation of SENSEVAL-2, the final setting of the Spanish exercise involved a number of choices detailed in section 2. In the following sections we describe the data, the manual tagging process (including the inter-tagger agreement figures), the participant systems and the accuracy results (including some baselines for comparison purposes).

2 Design Decisions

2.1 Task Selection

For Spanish SENSEVAL, the lexical-sample variant for the task was chosen. The main reasons for this decision are the following:

- During the same tagging session, it is easier and quicker to concentrate only on one word at a time. That is, tagging multiple instances of the same word.
- The all-words task requires access to a full dictionary. To our knowledge, there are no full Spanish dictionaries available (with low or no cost). Instead, the lexical-sample task required only as many dictionary entries as words in the sample task.

2.2 Word Selection

The task for Spanish is a "lexical sample" for 39 words[1] (17 nouns, 13 verbs, and 9 adjectives). See table 1 for the complete list of all words selected for the Spanish lexical sample task. The words can belong only to one of the syntactic categories. The fourteen words selected to be translation-equivalents to English has been:

- Nouns: *arte* (=art), *autoridad* (= authority), *canal* (= channel), *circuito* (= circuit), and *naturaleza* (= nature).
- Verbs: *conducir* (= drive), *tratar* (= treat), and *usar* (= use).
- Adjectives: *ciego* (= blind), *local* (= local), *natural* (= natural), *simple* (= simple), *verde* (= green), and *vital* (= vital).

2.3 Corpus Selection

The corpus was collected from two different sources: "El Periódico"[2] (a Spanish newspaper) and LexEsp[3] (a balanced corpus of 5.5 million words). The length of corpus samples is the sentence.

2.4 Selection of Dictionary

The lexicon provided was created specifically for the task and it consists of a definition for each sense linked to the Spanish version of EuroWordNet and, thus, to the English WordNet 1.5. The syntactic category and, sometimes, examples and synonyms are also provided. The connections to EuroWordNet have been provided in order to have a common language independent conceptual structure. Neither proper nouns nor multiwords has been considered. We have also provided the complete mapping between WordNet 1.5 and 1.6 versions[4]. Each dictionary entry have been constructed consulting the cor-

[1] The noun "arte" was not included in the exercise because it was provided to the competitors during the trial phase.

[2] The working corpus of the HERMES project CICYT TIC2000-0335-C03-02. More details at http://http://terral.ieec.uned.es/hermes.

[3] Provided by LEXESPIII project DGICYT APC 99-0105

[4] http://www.lsi.upc.es/~nlp/mapping.html

pus and multiple Spanish dictionaries (including the Spanish WordNet).

2.5 Annotation procedure

The Spanish SENSEVAL annotation procedure was divided into three consecutive phases.

- Corpus and dictionary creation
- Annotation
- Referee process

All these processes have been possible thanks to the effort of volunteers from three NLP groups from Universitat Politècnica de Catalunya[5] (UPC), Universitat de Barcelona[6] (UB) and Universidad Nacional de Educación a Distancia[7] (UNED).

2.5.1 Corpus and Dictionary Creation

The most important and crucial task was carried out by the UB team of linguists, headed by Mariona Taulé. They were responsible for the selection of the words, the creation of the dictionary entries and the selection of the corpus instances. First, this team selected the polysemous words for the task consulting several dictionaries including the Spanish WordNet and a quick inspection to the Spanish corpus. For the words selected, the dictionary entries were created simultaneously with the annotation of all occurrences of the word. This allowed the modification of the dictionary entries (i.e. adapting the dictionary to the corpus) during the annotation and the elimination of unclear corpus instances (i.e. adapting the corpus to the dictionary).

2.5.2 Annotation

Once the Spanish SENSEVAL dictionary and the annotated corpus were created, all the data was delivered to the UPC and UNED teams, removing all the sense tags from the corpus. Having the Spanish SENSEVAL dictionary provided by the UB team as the unique semantic reference for annotation both teams performed in parallel and simultaneously a new annotation of the whole corpus. Both teams where allowed to provide comments/problems on the each of the corpus instances.

2.5.3 Referee Control

Finally, in order to provide a coherent annotation, a unique referee from the UPC team collate both annotated corpus tagged by the UPC and the UNED teams. This referee was not integrated in the UPC team in the previous annotating phase. The referee was in fact providing a new annotation for each instance when occurring a disagreement between the sense tags provided by the UPC and UNED teams.

[5] http://www.lsi.upc.es/~nlp
[6] http://www.ub.es/ling/labing.htm
[7] http://rayuela.ieec.uned.es/

3 The Spanish data

3.1 Spanish Dictionary

The Spanish lexical sample is a selection of high medium and low polysemy frequent nouns, verbs an adjectives. The dictionary has 5.10 senses per wor and the polysemy degree ranges from 2 to 13. Noun has 3.94 ranging from 2 to 10, verbs 7.23 from 4 t 13 and adjectives 4.22 from 2 to 9 (see table 1 fo further details).

The lexical entries of the dictionary have the fol lowing form:

$< HEADWORD >\#$
$< POS >\#$
$< SENSENUMBER >\#$
$< GLOSS : EXAMPLEs >\#$
$SIN :< SINONYMWORDs >\#$
$< SYNSETNUMBERs >\#$

Figure 1: Dictionary entry format

For instance, the dictionary for noun headwor *arte* (= art) is:

arte#NCMS#1#Actividad humana o producto d tal actividad que expresa simbólicamente un as pecto de la realidad: el arte de la música; el art precolombino #SIN:?#00518008n/02980374n

arte#NCMS#2#Sabiduría, destreza o habilida de una persona en una actividad o con ducta determinada: tiene mucho arte bai lando; desplegó todo su arte para convencerl #SIN:?#03850627n#

arte#NCMS#3#Aparato que sirve par pescar#SIN:?#02005770n#

3.2 Spanish Corpus

We adopted, when possible, the guidelines propose by the SENSEVAL organisers (Edmonds, 2000). Fo each word selected having n senses we provided a least $75 + 15n$ instances. For the adjective *popular* larger set of instances has been provided to test per formance improvement when increasing the numbe of examples. These data has been then ramdoml divided in a ratio of 2:1 between training and tes set.

The corpus was structured following the standar SENSEVAL XML format.

3.3 Major problems during annotation

In this section we discuss the most frequent and reg ular types of disagreement between annotators.

In particular, the dictionary proved to be not suf ficiently representative of the selected words to b annotated. Although the dictionary was built fo the task, out of 48% of the problems during the sec ond phase of the annotation where due to the lacl

of the appropriate sense in the corresponding dictionary entry. This portion includes 5% of metaphorical uses not explicitly described into the dictionary entry. Furthermore, 51% of the problems reported by the annotators were concentrated only on five words (*pasaje*, *canal*, *bomba*, *usar*, and *saltar*).

Selecting only one sentence as a context during annotation was the other main problem. Around 26% of the problems where attributed to insufficient context to determine the appropriate sense.

Other sources of minor problems included different Part-of-Speech from the one selected for the word to be annotated, and sentences with multiple meanings.

3.4 Inter-tagger agreement

In general, disagreement between annotators (and sometimes the use of multiple tags) must be interpreted as misleading problems in the definition of the dictionary entries. The inter-tagger agreement between UPC and UNED teams was 0.64% and the Kappa measure 0.44%.

4 The Systems

Twelve systems from five teams participated in the Spanish task.

- Universidad de Alicante (UA) combined a Knowledge-based method and a supervised method. The first uses WordNet and the second a Maximum Entropy model.
- John Hopkins University (JHU) presented a metalearner of six diverse supervised learning subsystems integrated via classifier. The subsystems included decision lists, transformation-based error-driven learning, cosine-based vector models, decision stumps and feature-enhanced naive Bayes systems.
- Stanford University (SU) presented a metalearner mainly using Naive Bayes methods, but also including vector space, n-gram, and KNN classifiers.
- University of Maryland (UMD) used a margin-based algorithm to the task: Support Vector Machine.
- University of Manitoba (d6-10,dX-Z) presented different combinations of classical Machine Learning algorithms.

5 The Results

Table 1 presents the results in detail for all systems and all words. The best scores for each word are highlighted in boldface. The best average score is obtained by the JHU system. This system is the best in 12 out of the 39 words and is also the best for nouns and verbs but not for adjectives. The SU system gets the highest score for adjectives.

The associated agreement and kappa measures for each system are shown in Table 2. Again JHU system scores higher in both agreement and Kappa measures. This indicates that the results from the JHU system are closer to the corpus than the rest of participants.

6 Conclusions and Further Work

Obviously, an in deep study of the strengths and weaknesses of each system with respect to the results of the evaluation must be carried out, including also further analysis comparing the UPC and UNED annotations against each system.

Following the ideas described in (Escudero et al., 2000) we are considering also to add a cross-domain aspect to the evaluation in future SENSEVAL editions, allowing the training on one domain and the evaluation on the other, and vice-versa.

In order to provide a common platform for evaluating different WSD algorithms we are planning to process the Spanish corpus tagged with POS using MACO (Carmona et al., 1998) and RELAX (Padró, 1998).

7 Acknowledgements

The Spanish SENSEVAL has been possible thanks to the effort of volunteers from three NLP groups from UPC, UB, and UNED universities.

References

J. Carmona, S. Cervell, L. Màrquez, M.A. Martí, L. Padró, R. Placer, H. Rodríguez, M. Taulé, and J. Turmo. 1998. An Environment for Morphosyntactic Processing of Unrestricted Spanish Text. In *Proceedings of the First International Conference on Language Resources and Evaluation, LREC*, Granada, Spain.

P. Edmonds. 2000. Designing a task for SENSEVAL-2. Draft, Sharp Laboratories, Oxford.

G. Escudero, L. Màrquez, and G. Rigau. 2000. A Comparison between Supervised Learning Algorithms for Word Sense Disambiguation. In *Proceedings of the 4th Computational Natural Language Learning Workshop, CoNLL*, Lisbon, Portugal.

L. Padró. 1998. *A Hybrid Environment for Syntax-Semantic Tagging.* Phd. Thesis, Software Department (LSI). Technical University of Catalonia (UPC).

words	p	e	s	MF	UA	SU	JHU	UMD	d6	d7	d8	d9	d10	dX	dY	dZ
actuar	v	155	6	0.28	0.27	0.60	0.56	0.45	0.25	0.27	0.40	0.36	0.35	0.22	**0.67**	0.22
apoyar	v	210	4	0.64	0.63	**0.70**	0.68	0.67	0.64	0.63	0.67	0.64	0.66	0.66	0.64	0.64
apuntar	v	191	8	0.47	0.55	0.55	**0.65**	0.53	0.49	0.49	0.51	0.51	0.55	0.49	0.47	0.49
autoridad	n	122	6	0.49	**0.68**	0.50	0.53	0.47	0.50	0.56	0.56	0.47	0.62	0.47	0.62	0.50
bomba	n	113	2	0.71	0.27	0.70	0.68	0.73	0.78	0.71	0.79	**0.80**	0.74	0.78	0.59	**0.80**
brillante	a	256	2	0.52	0.63	0.76	**0.83**	0.76	0.81	0.76	0.81	0.76	0.78	0.73	0.78	0.78
canal	n	156	5	0.33	0.34	0.63	0.68	**0.76**	0.49	0.59	0.56	0.51	0.56	0.56	0.46	0.59
ciego	a	114	4	0.54	**0.71**	0.69	0.62	0.62	0.64	0.55	0.57	0.60	0.60	0.60	0.55	0.57
circuito	n	123	4	0.34	0.43	**0.59**	0.57	0.37	0.49	0.55	0.61	0.31	0.53	0.53	0.29	0.49
claro	a	204	7	0.83	0.82	**0.88**	0.82	0.83	0.83	0.85	0.85	0.83	0.86	0.85	0.85	0.85
clavar	v	131	9	0.44	0.50	0.64	0.48	0.64	0.61	**0.68**	0.64	0.52	0.61	0.57	0.57	0.57
conducir	v	150	9	0.35	0.35	0.43	0.44	**0.46**	0.41	0.43	0.43	0.35	0.41	0.37	0.41	0.41
copiar	v	147	8	0.32	0.42	0.55	0.45	0.47	0.45	0.40	0.42	0.53	0.43	0.38	**0.62**	0.42
corazon	n	146	5	0.36	0.23	0.53	0.77	0.68	0.66	0.74	**0.79**	0.53	0.77	0.64	0.68	0.62
corona	n	119	4	0.45	0.53	**0.80**	0.70	0.53	0.55	0.62	0.57	0.55	0.57	0.55	0.53	0.55
coronar	v	244	6	0.32	0.49	0.65	**0.70**	0.65	0.55	0.62	0.61	0.64	0.61	0.59	0.41	0.62
explotar	v	133	6	0.32	0.49	0.56	0.56	0.56	0.46	0.39	0.41	0.49	0.41	0.44	**0.61**	0.41
gracia	n	160	6	0.30	0.28	**0.79**	0.74	0.61	0.69	0.66	0.79	0.59	0.72	0.70	0.70	0.80
grano	n	78	3	0.44	0.37	0.32	0.50	0.45	0.36	0.50	0.32	0.32	0.45	0.36	**0.64**	0.36
hermano	n	135	5	0.61	0.74	0.58	0.74	0.72	0.70	0.74	0.74	0.70	**0.75**	0.70	0.74	0.74
local	a	139	3	0.74	0.84	0.78	**0.89**	0.75	0.76	0.84	0.85	0.73	0.84	0.78	0.82	0.82
masa	n	131	5	0.45	0.39	0.63	**0.68**	0.61	0.54	0.54	0.61	0.56	0.66	0.56	0.41	0.59
natural	a	137	6	0.25	0.34	0.48	**0.60**	0.45	0.36	0.41	0.40	0.31	0.47	0.41	0.38	0.41
naturaleza	n	167	10	0.44	0.45	0.66	0.59	0.54	0.64	0.70	0.66	0.52	**0.68**	0.57	0.64	0.59
operacion	n	142	5	0.35	**0.71**	0.60	0.55	0.49	0.43	0.45	0.40	0.57	0.45	0.47	**0.60**	0.47
organo	n	212	4	0.52	0.73	**0.83**	0.81	0.73	0.70	0.64	0.64	0.70	0.64	0.64	0.53	0.68
partido	n	159	2	0.55	0.81	0.84	**0.86**	0.81	0.74	0.74	0.74	0.67	0.75	0.72	0.67	0.77
pasaje	n	112	4	0.39	**0.83**	0.44	0.56	0.34	0.39	0.39	0.39	0.32	0.56	0.41	0.29	0.39
popular	a	661	3	0.65	0.77	**0.90**	0.83	0.75	0.77	0.78	0.80	0.71	0.77	0.77	0.68	0.75
programa	n	142	6	0.49	0.36	0.49	**0.64**	0.49	0.49	**0.64**	0.55	0.47	0.40	0.40	0.49	0.45
saltar	v	137	14	0.15	0.51	0.49	0.57	0.51	0.16	0.35	0.32	0.11	0.54	0.32	0.65	0.30
simple	a	217	5	0.61	0.67	**0.77**	0.63	0.65	0.68	0.70	0.72	0.65	0.72	0.67	0.67	0.65
tabla	n	119	3	0.51	**0.88**	0.73	0.66	0.71	0.66	0.59	0.73	0.76	0.68	0.73	0.59	0.76
tocar	v	236	12	0.31	0.51	0.61	**0.66**	0.59	0.41	0.51	0.49	0.39	0.47	0.42	0.34	0.42
tratar	v	192	13	0.21	0.39	0.46	**0.60**	0.56	0.27	0.39	0.37	0.30	0.43	0.30	0.24	0.34
usar	v	167	4	0.68	0.77	0.73	**0.79**	0.70	0.70	0.68	0.70	0.70	0.64	0.70	0.70	0.70
vencer	v	183	8	0.63	**0.72**	0.69	0.62	0.69	0.69	**0.72**	0.71	0.69	0.71	0.69	0.71	0.69
verde	a	109	9	0.37	0.48	0.61	0.52	0.64	0.58	0.58	0.61	0.61	**0.67**	0.48	0.55	**0.67**
vital	a	256	4	0.45	0.65	0.68	**0.77**	0.68	0.54	0.67	0.68	0.51	0.66	0.47	0.53	0.51
NOUNS	n	2336	4	0.45	0.55	0.63	**0.66**	0.59	0.58	0.61	0.61	0.55	0.62	0.58	0.56	0.60
VERBS	v	2276	7	0.40	0.51	0.59	**0.60**	0.58	0.47	0.5.	0.51	0.48	0.52	0.47	0.54	0.48
ADJS	a	2093	4	0.58	0.66	**0.73**	0.72	0.68	0.66	0.68	0.70	0.63	0.71	0.64	0.65	0.67
TOTAL	T	6705	5	0.48	0.56	0.64	**0.65**	0.61	0.56	0.59	0.60	0.55	0.61	0.56	0.57	0.57

Table 1: Evaluation of Spanish words. **p** stands for Part-of-Speech; **e** for the total number of examples (including train and test sets); **s** for the number of senses; **MF** for the Most Frequent Sense Classifier and the rest are the system acronyms.

words	UA	SU	JHU	UMD	d6	d7	d8	d9	d10	dX	dY	dZ
Agreement	0.51	0.63	**0.65**	0.61	0.55	0.57	0.59	0.53	0.59	0.55	0.51	0.57
Kappa	0.20	0.34	0.47	0.20	0.13	0.19	0.23	0.06	0.24	0.15	-0.03	0.15

Table 2: Agreement and Kappa measures

SENSEVAL-2: The Swedish Framework

Dimitrios KOKKINAKIS
Språkdata, Göteborg University
Box 200, SE-405 30
Göteborg, Sweden
Dimitrios.Kokkinakis@svenska.gu.se

Jerker JÄRBORG
Språkdata, Göteborg University
Box 200, SE-405 30
Göteborg, Sweden
Jerker.Jaerborg@svenska.gu.se

Yvonne CEDERHOLM
Språkdata, Göteborg University
Box 200, SE-405 30
Göteborg, Sweden
Yvonne.Cederholm@svenska.gu.se

Abstract

In this paper we describe the organisation and results of the SENSEVAL-2 exercise for Swedish. We present some of the experiences we gained by participating as developers and organisers in the exercise. We particularly focus on the choice of the lexical and corpus material, the annotation process, the scoring scheme, the motivations for choosing the lexical-sample branch of the exercise, the participating systems and the official results.

Introduction

Word sense ambiguity is a potential source for errors in human language technology applications, such as Machine Translation, and it is considered as *the* great open problem at the lexical level of Natural Language Processing (NLP). There are, however, several computer programs for automatically determining which sense of a word is being used in a given context, according to a variety of semantic, or defining dictionaries as demonstrated in the SENSEVAL-1 exercise; (Kilgarriff and Palmer, 2000). The purpose of SENSEVAL is to be able to say which programs and methods perform better, which worse, which words, or varieties of language, present particular problems to which programs; when modifications improve performance of systems, and how much and what combinations of modifications are optimal. Specifically for Swedish, we would also like to investigate to what extent sense disambiguation can be accomplished and the potential resources available for the task. We would thus be creating a framework that can be shared both within the exercise and for future evaluation exercises of similar kind, national and international.

1 Choice of Task

Three tasks were identified for SENSEVAL-2, namely: *the lexical-sample*, *the all-words* and *the 'in a system'* tasks. In the lexical sample task, first, we sample the lexicon, then we find instances in context of the sample words and the evaluation is carried out on the sampled instances. In the all-word task a system will be evaluated on its disambiguation performance on every word in the test collection. Finally, in the third type of task, a word sense disambiguation (WSD) system is evaluated on how well it improves the performance of a NL system (MT, IR etc). The reasons we chose the lexical-sample task for Swedish are summarised below:

1. Cost-effectiveness of annotation: it is easier and quicker for the human annotators to sense-tag multiple occurrences of one word at a time, particularly when robust interactive means are utilized (Section 3);
2. The lexical-sample reduces the work of preparing training data since only a subset of the sense inventory is used;
3. More systems can/could (eventually) participate;
4. The all-words task requires access to a full dictionary, which is problematic from the copyright point of view, since industrial partners were also allowed to participate; and, as Kilgarriff and Palmer (2000) noted:
5. Provided that the sample is well chosen, the lexical sample strategy would be more informative about the current strengths and failings of sense disambiguation research than the all-words task.

2 Development Process

In this section we will give a concise description of how the whole exercise (for Swedish) was set up, putting more emphasis on some of the main ingredients of the work, i.e. sampling, resources, annotation and scoring.

A number of likely participants were invited to express their interest and participate in the Swedish SENSEVAL (summer, 2000). A plan for selecting the evaluation material was agreed in Språkdata, and human annotators were set on the task of generating the training and testing material. The material was released to the participants at the end of April 2001 and during the second week of June, 2001 the results were returned for scoring. The Swedish SENSEVAL material was divided into three parts and released in stages:

- **Trial data**: freezing and showing the data formatting conventions (lexicon & corpus);
- **Training data**: the finalised sense inventory and portion of the 'gold standard';
- **Evaluation data**: the rest of the 'gold standard', untagged.

2.1 Dictionary and Corpus

At least three lexical resources were candidates for the Swedish lexicon-sample task. These were the Swedish versions of the WordNet (http://www.ling.lu.se/projects/Swordnet) and the Swedish SIMPLE (http://spraakdata.gu.se/simple/), as well as the Gothenburg Lexical Data Base/semantic Database (GLDB/SDB) (http://spraakdata.gu.se/lb/gldb.html). We chose the GLDB/SDB. The creation of a Swedish version of WordNet, a resource that is extensively used for the semantic annotation of texts in other languages, is under development and had (up to that point) limited coverage, while the SIMPLE lexicon, although available, has limited coverage (in principle it could be used and it is linked to the GLDB/SDB). However, a drawback of the Swedish SIMPLE is that very fine-grained sub-senses are not adequately described (or not described at all) in the material. GLDB/SDB is a generic defining dictionary of 65,000 lemmas available and developed at our department and became the final choice for the lexical inventory (see Allén, 1999[1981] for a description of the model utilized in the dictionary).

For the textual material we chose the Stockholm-Umeå Corpus (SUC), Ejerhed *et al.* (1992). The particular corpus was chosen for three main reasons. It is available to the research community; it is considered the "standard, reference" corpus for contemporary written Swedish; and, third, it is the corpus utilised in the SemTag project (next section).

2.2 Sampling

There is no standard method for sampling the lexical data. However, certain features were considered. These were: frequency, polysemy, part-of-speech and distribution of senses. Words were chosen based not so much on intuition, but rather on their frequency and polysemy. Still, it was hard to find a balance between these two features since high frequency words tend to be monosemous in a corpus, while highly polysemous words tend to have few senses in a corpus. In the case that a word was frequent and polysemous we tried to provide more data (context), than for words that were less frequent. Part-of-speech information was consulted for the decision of choosing more nouns in the sample (highest portion in the GLDB/SDB), than verbs (less than nouns, but more than adjectives in the GLDB/SDB) and adjectives (which are fewer than nouns and verbs in GLDB/SDB). We chose a sample of words where the amount of senses was evenly distributed, i.e. lemmas (dictionary entries) with 2-7 lexemes (senses) and 1-23 cycles (subsenses).

2.3 SemTag

Creating a sense-annotated reference corpus is a laborious task. Therefore, we developed the majority of the test and reference material within an ongoing project highly relevant for our mission, namely SemTag (*Lexikalisk betydelse och användningsbetydelse* – "Lexical Sense and Sense in Context", financed by the *Swedish Council for Research in the Humanities and Social Sciences* (HSFR)); see Järborg (1999). In brief, the purpose of the project is to create a large sample of sense-annotated corpus (several hundreds of thousands of words), which can be used among other things for:

- measuring the performance of automatic methods for WSD;

- testing, in practice and on a large scale, the validity of the lemma-lexeme model implemented in GLDB/SDB;
- the improvement of lexicographic descriptions, and the production of (new and) more fine-grained senses in GLDB/SDB;
- the adjustment of the definitions in GLDB/SDB to better fit the textual use;
- describing new words, not covered by the content of the GLDB/SDB;
- producing material, adequate for training supervised methods to sense disambiguation.

2.4 Corpus/Sense Inventory

Table 1 shows information on the sense inventory, the amount of corpus instances (training/testing) and the distribution of senses and sub-senses (Lexemes/Cycles) in the material for the twenty nouns (N), fifteen verbs (V) and the five adjectives (A). The total amount of training and testing corpus instances was: 8716/1525. The average polysemy in the sample is 3,5/7,6 for lexemes and cycles respectively.

Word	POS	Corpus Instances	Lexemes/ Cycles
barn/1	N	656/115	3/6
betydelse/1	N	295/52	2/1
färg/1	N	110/19	4/11
konst/1	N	77/13	3/6
kraft/1	N	152/27	4/11
kyrka/1	N	154/27	2/3
känsla/1	N	142/25	2/4
ledning/1	N	91/16	4/1
makt/1	N	128/22	3/4
massa/1	N	93/16	6/3
mening/1	N	168/29	4/1
natur/1	N	90/16	3/4
program/1	N	139/24	4/10
rad/1	N	145/25	4/3
rum/1	N	223/39	3/7
scen/1	N	101/17	4/7
tillfälle/1	N	117/20	2/4
uppgift/1	N	174/30	2/3
vatten/1	N	285/50	2/3
ämne/1	N	198/34	4/4
betyda/1	V	198/35	4/4
flytta/1	V	188/33	2/4
fylla/2	V	96/17	4/11
följa/1	V	345/61	5/19
förklara/1	V	169/30	2/9
gälla/1	V	843/148	4/6
handla/1	V	250/44	4/5
höra/1	V	523/92	5/14
måla/1	V	96/16	2/7
skjuta/1	V	79/14	6/15
spela/1	V	267/47	6/23
vänta/1	V	248/43	3/15
växa/1	V	203/36	2/9
öka/1	V	436/77	2/2
öppna/1	V	147/25	4/16
bred/1	A	103/18	3/1
klar/1	A	307/54	4/11
naturlig/1	A	139/24	4/5
stark/1	A	352/62	5/11
öppen	A	189/33	7/21

Table 1. Data for the Swedish Lexical Sample

3 Annotation

The annotation was carried out interactively using a concordance-based interface (developed in SemTag) and which interacts with the corpus and the dictionary; (see http://svenska.gu.se/~svedk/SENSEVAL/images/semtag.gif for a screenshot of this tool). Due to our limited financial resources only two professional lexicographers and a trained Phd student were involved in the tagging process, which was preferred to (untrained) students doing the annotation. High replicability between the human annotators was observed (>95%). The uncertain cases were not used in the training or testing material, while the provided dictionary descriptions for the 40 lemmas were revised (extended and/or modified) prior to their release.

4 Scoring

Prior to SENSEVAL, evaluating WSD performance was based solely on the exact match criterion, which is not consider a "fair" metric, and has a lot of drawbacks (e.g. it does not account for the semantic distance between senses when assigning penalties for incorrect labels, and it does not offer a mechanism to offer partial credit; *cf.* Resnik & Yarowsky (2000)) Instead, in SENSEVAL-2 three scoring policies are adopted:

1. **Fine-grained**: answers must match exactly
2. **Coarse-grained**: answers are mapped to coarse-grained senses and compared to the gold standard tags, also mapped to coarse-grained ones (sense map is required; see below)
3. **Mixed-grained**: if a sense subsumption hierarchy is available, then the mixed-

grained scoring gives some credit to choosing a more coarse-grained sense than the gold standard tag, but not full credit (also using a sense map; see below).

A "sense map" containing a complete list of all sense-ids involved in the evaluation was provided in order to perform the two last types of scoring policies. Each line in the sense map included sense subsumption information and contained a list of the subsumer senses and branching factors.

5 Participants and Results

Five groups showed interest in participating in the Swedish task (eight systems in total). Table 2 provides information for the participating systems, while their average performance is given in Table 3, the score in parenthesis concerns: Verbs/Noun/ Adjectives. All systems returned answers for all instances, thus precision equals recall, all used supervised methods and all systems scored lower on the adjectives and higher on the nouns.

Group (Systems)	Method	Contact Person(s)
Uppsala Univ. (PWE, 3)	TBL-tränade Prolog word experts	T. Lager, N. Zinovjeva
Linköping Univ. (LIU, 1)	Multilevel decision list approach	L. Ahrenberg, M. Merkel, M. Andersson
Göteborg Univ. (Språkdata, 2)	Machine learning & feature overlap	D. Kokkinakis
John Hopkins Univ. (JHU, 1)	---	D. Yarowsky
Maryland Univ. (UMD, 1)	Support vector machine	P. Resnik, J. Stevens, C. Cabezas

Table2. Participants

System	Results	
	Fine-Grained	**Mixed-Grained**
JHU	70,1(63,4/76,9/51,8)	74,7(70,9/79,8/59,5)
PWE-Vote	63,0(58,5/72,7/48,7)	68,6(65,9/75,0/57,9)
Språkdata-ML	62,0(57,8/71,3/48,2)	68,2(66,1/74,9/54,4)
PWE-Simple	61,1(55,4/73,2/43,5)	66,8(63,2/75,7/51,7)
UMD	61,1(56,4/71,4/45,5)	65,6(61,7/73,6/54,3)
LIU	56,5(47,8/71,6/40,8)	61,6(54,7/73,3/49,6)
PWE-Disj	54,0(46,3/67,7/38,4)	60,7(55,3/71,0/47,5)
Språkdata-Overlap	46,0(36,6/57,8/43,1)	55,8(47,8/65,7/53,8)

Table 3. Results. *Overall Precision* followed by precision for *(Verb/Noun/Adective)* instances

Conclusion

The process of WSD is a complex, controversial matter, but relevant for a number of NLP applications. Our contribution to the exercise will eventually sharpen the focus of WSD in Sweden; the material developed in SENSEVAL-2 can be used as benchmark for other researchers that need to measure their system's WSD performance against a concrete reference point (although the dictionary is limited). We think that WSD opens up exciting opportunities for linguistic analysis, contributing with very important information for the assignment of lexical semantic knowledge to polysemous and homonymous content words. The existence of sense ambiguity (polysemy and homonymy) is one of the major problems affecting the usefulness of basic corpus exploration tools. In this respect, we regard WSD as a very important process when it is seen in the context of a wider and deeper NLP system.

Acknowledgements

We would like to thank the *Swedish Council for Research in the Humanities and Social Sciences* (HSFR) for providing financial support for the coordination of the task.

References

Allén S. (1999[1981]). The Lemma-Lexeme Model of the Swedish Lexical Database. *Empirical Semantics*, 376-387. Rieger B. (ed). Bochum. (Reprinted in: Allén S. (1999). *Modersmålet i Fäderneslandet. Ett urval uppsatser under fyrtio år av Sture Allén*, 268-278. Meijerbergs Arkiv 25).

Ejerhed E., Källgren G., Wennstedt G. and Åström M. (1992). *The Linguistic Annotation of the Stockholm-Umeå Corpus project.* Technical Report No. 33, Univ. of Umeå.

Järborg J. (1999). *Lexikon i konfrontation.* Research Reports from the Department of Swedish, Språkdata, GU-ISS-99-6. Available from: http://svenska.gu.se/~svedk/resrapp/konfront.pdf. (In Swedish).

Kilgarriff A. and Palmer M. (2000). Introduction to the Special Issue on SENSEVAL. *Computer and the Humanities*, 00:1-13, Kluwer Acad. Publishers.

Resnik P. and Yarowsky D. (2000). Distinguishing Systems and Distinguishing Senses: New Evaluation Methods for Word Sense Disambiguation. *Natural Language Engineering*, 5(2):113-133, Cambridge.

The SENSEVAL-2 Panel on Domains, Topics and Senses

Paul Buitelaar

DFKI GmbH
Stuhlsatzenhausweg 3
D-66123 Saarbruecken, Germany
paulb@dfki.de

1 Introduction: Why Domains Matter in Sense Disambiguation

An important aspect of sense disambiguation is the wider semantic space (domain, topic) in which the ambiguous word occurs. This may be most clearly illustrated by some cross-lingual examples, as they would appear in (machine) translation. Consider for instance the English word *housing*. In a more general "sense", this translates in German into *Wohnung*. In an engineering setting however it translates into *Gehäuse*. Also verbs may be translated differently (i.e. have a different sense) according to the semantic space in which they occur. For instance, English *warming up* translates into *erhitzen* in a more general sense, but into *aufwärmen* in the sports domain.

Because of the apparent relevance then of domains or topics on sense disambiguation, a panel was organized at SENSEVAL-2 to discuss some current and previous work in this area. The paper presents a more extended overview based on the relevant literature, besides giving a summary of the discussion that developed after the panel presentations.

2 Domains, Topics and Senses

2.1 Subject Codes

A semantic space may be indicated in a dictionary by use of a so-called "subject code". In LDOCE for instance, subject codes like MD, for the medical domain, or ML, for meteorology are used to define which senses of a word are used in which domains. Three of the senses of the word *high* for instance correspond to three different domains: music (a high tone), drugs (the experience of being high) and meteorology (a high pressure area).

Subject codes can be used to detect the topic of a text segment by simply counting their frequency over all content words (Walker and Amsler 1986). At the same time, however, subject codes can be used in sense disambiguation by constructing topic specific context models (Guthrie et. al 1991). Such "neighborhoods" can be constructed by taking into account all words in the definitions and in sample sentences of all words in the dictionary that share the same subject code. For instance, the word *bank* has the following neighborhoods for the financial and medical domains:

write	*safe*	*sum*
account	*person*	*put*
take	*money*	*order*
keep	*pay*	*supply*
paper	*draw*	*cheque*

Table 1: Financial neighborhood of *bank*

medicine	*product*	*hold*
origin	*place*	*human*
treatment	*blood*	*hospital*
use	*store*	
organ	*comb*	

Table 2: Medical neighborhood of *bank*

Using subject codes in sense disambiguation has been shown to be fruitful, relative to using other sources of knowledge. As reported in (Stevenson and Wilks 1999), the performance of using only subject codes (79% precision) was much better than that of using only dictionary definition words (65%), or selection restrictions (44%). Given these results it seems worthwhile to identify also the semantic space of WordNet synsets more explicitly by the introduction of subject codes (Magnini and Cavaglià 2000). This allows for grouping together synsets across part-of-speech, as in the medical domain

(doctor#1, hospital#1; operate#7) and across sub-hierarchies, as in the sports domain (life_form#1: athlete#1; physical_object#1: game_equipment#1; act#2: sport#1; location#1: playing_field#1).

2.2 Topic Signatures and Variation

The topic specific context models as constructed by (Guthrie et al. 1991) can be viewed as "signatures" of the topic in question. Such topic signatures can, however, be constructed even without the use of subject codes by generating them (semi-) automatically from a lexical resource and then validating them on topic specific corpora (Hearst and Schütze 1993).

An extension of this idea is to treat senses, or rather WordNet synsets, as topics for which a signature can be constructed. One approach to this is to retrieve relevant documents through search engines on the web by defining queries for each synset (Agirre et al. 2000, Agirre et al. 2001). For instance, the following query can be defined for the first WordNet sense of *boy*:

#1 (*boy* AND (*altar boy* OR *ball boy* OR ...)
#2 AND NOT (*man* OR ... OR *broth of a boy* OR
#3 *son* OR ... OR *mama's boy* OR
#4 *nigger* OR ... OR *black*)

The document collections retrieved are then analysed and a list of the most relevant words for each synset is generated as its topic signature. Examples (abridged) for the first three senses of *boy* are:

Sense 1	Sense 2	Sense 3
child	gay	human
Child	reference	son
person	tpd-results	Human

Constructing topic signatures for senses implies that a dominant sense can be identified given a certain topic or domain. This may be true for clearly ambiguous words (i.e in the case of homonymy). For instance, *sentence* will be dominant in the judicial sense in the law domain and in the syntactic sense in the linguistics domain. However, for words with related senses (i.e in the case of systematic polysemy) the topic signatures will overlap, as with the results on *boy* in sense 1: *young male person* and sense 3: *son*. This has been shown also from a somewhat different viewpoint in reaction to (Gale et al. 1992), in which it was stated that one sense will be uniquely used within a discourse (which we can equate with a topic or domain for our purposes here). Instead, many words have overlapping senses that will be used simultaneously throughout one discourse (Krovetz 1998).

The main question that remains now is, what exactly constitutes a discourse / subject / topic / domain? We can get closer at answering this question by looking at some empirical sense disambiguation results that involve a variation of topic. More specifically, we can observe some effects of topic variation by training a sense disambiguation system on one topic and applying it to another. For instance, training on Wall Street Journal while testing on SemCor and vice versa shows a degrading of 12% and 19% in precision (Escudero et al. 2000). On the other hand, applying context information (collocations) extracted from Wall Street Journal to a financial text in SemCor shows significantly higher precision than on texts in other domains in SemCor (Martinez and Agirre 2000).

These results therefore suggest that a discourse / subject / topic / domain corresponds to a larger or smaller chunk of text (a corpus, a text or a text segment) with a homogeneous distribution of senses and corresponding collocations.

2.3 Tuning

But even with a clearly defined domain, it is far from certain that any general sense inventory will be appropriate. "The usual scenario ... has been that the word senses are taken from a general purpose dictionary, ... whereas the material to be disambiguated is ... Wall Street Journal. ... So, the profiles [Signatures, Collocations] ... will be for general English senses according to the WSJ ..." (Kilgarriff 1998). Instead, a general sense inventory needs to be tuned to the domain at hand. This involves selecting only those senses that are most appropriate for the domain, as well as extending the sense inventory with novel words (terms) and novel senses, specific to the domain (Basili et al. 1997; Cucchiarelli and Velardi 1998; Turcato et al. 2000; Buitelaar and Sacaleanu 2001; Vossen 2001).

According to the method described in (Cucchiarelli and Velardi 1998), a domain specific sense inventory that is balanced (even distribution of words to senses) and at the right

level of abstraction (ambiguity vs. generalization) can be selected automatically given the following criteria: "Generality", "Discrimination Power", "Domain Coverage" and "Average Ambiguity." Applying these criteria in a quantitative way to a general sense inventory (i.e the WordNet hierarchy) and a given domain specific corpus automatically selects a set of relevant categories (i.e. top level synsets). For instance, this method selects following categories for the financial domain:

person, individual,...
instrumentality,...
written_communication,...
possession,...

Only senses that are subsumed by these categories are included in the domain specific sense inventory. For instance, for the word *stock*, only 5 out of 16 senses are selected:

#1 capital > asset > possession
#2 support > device > instrumentality
#4 document > ... > written_communication
#5 accumulation > asset > possession
#6 ancestor > relative > person,individual

Senses that are discarded include:

#7 soup > ...
#9 plant_part > ...
#12 lineage,line,line_of_descent > ...
#14 lumber,timber > ...

The method described above uses a top down approach that propagates the domain relevance of certain top level synsets down through the (WordNet) hierarchy. A somewhat different approach would be to assign a domain relevance to each concept (i.e. word sense, synset) from the bottom up (Buitelaar and Sacaleanu 2001). This method determines the domain specific relevance of (WordNet, GermaNet) synsets on the basis of the relevance of their constituent synonyms that co-occur within representative domain corpora.

Next to selecting domain relevant concepts from the general sense inventory, novel terms (those not covered by the sense inventory) need to be accounted for also. This includes adding morphological and syntactic variants of known terms (Vossen 2001) as well as extending the inventory with semantically related terms through classification and/or clustering.

3 Panel Discussion

In the panel presentations most of the issues discussed above were addressed. Central to the discussion were the following two questions:

- Is generic sense disambiguation possible?
- Is sense disambiguation always necessary?

The first question concerns the influence of the semantic space (topic, domain, etc.) on the disambiguation process. Unlike with PoS tagging, it seems hard and perhaps even theoretically impossible to define a 'general' training corpus and sense inventory for sense disambiguation. Instead, it seems necessary to tightly connect sense disambiguation to topic detection or text classification in order to recognize the wider semantic space of ambiguous words. The second question is concerned with the even more fundamental observation that sense disambiguation is unneccessary if one sense (or more than one, in the case of systematic polysemy) can be assigned unambiguously within a certain semantic space. The disambiguation problem then shifts towards an appropriate modelling of such semantic spaces (i.e. domain modelling). In summary, it may not be feasible to separate sense disambiguation from the domain in which it operates, which in turn implies that modelling this domain is the first priority for sense disambiguation. In the discussion, however, several arguments were raised against such a view of sense disambiguation.

First of all, such an approach drives us back to earlier domain specific methods. These were not very robust and required major efforts in adapting to new domains. As a counter argument to this point, it was noted that there are now many robust, machine-learning based methods available for lexical acquisition, which would allow for a rapid adaptation of the disambiguation resources to a new domain. The second main issue raised was that, from an evaluation point of view, it is important to evaluate the performance of different algorithms, independent from a specific domain or application. As a counter argument to this, the question was asked what such an evaluation

would then prove. Sense disambiguation evaluated without a particular (application) domain can only show an artificial result which is hard to interpret and to generalize over. This is illustrated in particular by low interannotator agreement scores obtained when disambiguating without the context of a certain domain.

The discussion did not reach a consensus on these points, although there was general agreement that future evaluation efforts in sense disambiguation should take applications (and hence certain domains) into account. The following table gives an overview of those teams that participated at SENSEVAL-2 and declared to be using domains, topical context or the „One Sense per Discourse" heuristic.

Team	Domain Information	Topical Context	One Sense / Discourse
Lexical Sample Task (English)			
IRST	✔		
TALP	✔	✔	
BCU-EHU		✔	
KUNLP		✔	
All Words Task (English)			
IRST	✔		
BCU-EHU		✔	
Sheffield		✔	
Sussex			✔
UCLA			✔

On the lexical sample task, *KUNLP* and *TALP* had both high precision and recall, while *BCU-EHU* and *IRST* reached the highest precision of all participating systems, but at a low recall. On the all words task, all teams in the table scored average to low, except for *IRST*, which reached again a very high precision at a low recall.

These results are unfortunately still inconclusive about the general merit of domain and topic information. Only the anomalous results of *IRST* may indicate the advantage of domain information for reaching a high precision in sense disambiguation.

4 Acknowledgements

Many thanks to Eneko Agirre, Nancy Ide, Bernardo Magnini and Piek Vossen for their contributions to the panel, and to the SENSEVAL-2 audience for their active participation in the discussion. This research has in part been supported by EC/NSF grant *IST-1999-11438* for the MUCHMORE project.

References

Agirre E., Ansa O., Hovy E., Martinez D. *Enriching very large ontologies using the WWW.* In: Proceedings of the Ontology Learning Workshop ECAI 2000.

Agirre E., Ansa O., Martinez D., Hovy E. *Enriching WordNet concepts with topic signatures.* In: Proceedings NAACL WordNet Workshop, 2001.

Basili R., Della Rocca M., Pazienza M.-T. *Contextual Word Sense Tuning and Disambiguation.* Applied Artificial Intelligence, vol. 11, 1997.

Buitelaar P., Sacaleanu B. *Ranking and Selecting Synsets by Domain Relevance.* In: Proceedings NAACL WordNet Workshop, 2001.

Cucchiarelli A., Velardi P. *Finding a Domain-Appropriate Sense Inventory for Semantically Tagging a Corpus.* In: Journal of Natural Language Engineering, 1998

Escudero G., Màrquez L., Rigau G. *An Empirical Study of the Domain Dependence of Supervised Word Sense Disambiguation Systems.* In: EMNLP 2000.

Gale W., Church K., Yarowsky D. *One Sense per Discourse.* In: Proceedings of the 4th DARPA Speech and Natural Language Workshop, 1992.

Hearst M., Schütze H. *Customizing a Lexicon to Better Suit a Computational Task.* In: Proceedings ACL SIGLEX Workshop 1993.

Guthrie J. A., Guthrie I., Wilks Y., Aidinejad H. *Subject Dependent Co-Occurrence and Word Sense Disambiguation.* In: Proceedings of ACL 1991.

Kilgarriff A. *Bridging the gap between lexicon and corpus: convergence of formalisms.* In: Proceedings of LREC Workshop on Adapting Lexical Resources, 1998.

Krovetz R. *More than one sense per discourse.* NEC Research Memorandum, 1998.

Magnini B., Cavaglià G. *Integrating Subject Field Codes into WordNet.* In: Proceedings LREC 2000.

Martinez D., Agirre E. *One Sense per Collocation and Genre/Topic Variations.* In: Proceedings EMNLP 2000.

Stevenson M., Wilks Y. *Combining Weak Knowledge Sources for Sense Disambiguation.* In: Proceedings IJCAI 1999.

Turcato D., Popowich F., Toole J., Fass D., Nicholson D., Tisher G. *Adapting a synonym database to specific domains.* In: Proceedings of the ACL workshop on recent advances in NLP and IR. Hong Kong, 2000.

Vossen P. *Extending, Trimming and Fusing WordNet for Technical Documents.* In: Proceedings NAACL WordNet Workshop, 2001.

Walker D., Amsler R. *The use of machinereadable dictionaries in sublanguage analysis* In: Analyzing Language in Restricted Domains, 1986.

The Japanese Translation Task: Lexical and Structural Perspectives

Timothy Baldwin,* Atsushi Okazaki,† Takenobu Tokunaga† and Hozumi Tanaka†

* CSLI, Stanford University <tbaldwin@csli.stanford.edu>

† Tokyo Institute of Technology <{okazaki,take,tanaka}@cl.cs.titech.ac.jp>

Abstract

This paper describes two distinct attempts at the SENSEVAL-2 Japanese translation task. The first implementation is based on lexical similarity and builds on the results of Baldwin (2001b; 2001a), whereas the second is based on structural similarity via the medium of parse trees and includes a basic model of conceptual similarity. Despite its simplistic nature, the lexical method was found to perform the better of the two, at 49.1% accuracy, as compared to 41.2% for the structural method and 36.8% for the baseline.

1 Introduction

Translation retrieval is defined as the task of, for a given source language (L1) input, retrieving the target language (L2) string which best translates it. Retrieval is carried out over a **translation memory** made up of **translation records**, that is L1 strings coupled with an L2 translation. A single translation retrieval task was offered in SENSEVAL-2, from Japanese into English, and it is this task that we target in this paper.

Conventionally, translation retrieval is carried out by way of determining the L1 string in the translation memory most similar to the input, and returning the L2 string paired with that string as a translation for the input. It is important to realise that at no point is the output compared back to the input to determine its "translation adequacy", a job which is left up to the system user.

Determination of the degree of similarity between the input and L1 component of each translation record can take a range of factors into consideration, including lexical (character or word) content, word order, parse tree topology and conceptual similarity. In this paper, we focus on a simple character-based (**lexical**) method and more sophisticated parse tree comparison (**structural**) method.

Both methods discussed herein are fully unsupervised. The lexical method makes use of no external resources or linguistic knowledge whatsoever. It treats each string as a "bag of character bigrams" and calculates similarity according to Dice's Coefficient. The structural method, on the other hand, relies on both morphological and syntactic analysis, in the form of the publicly-available JUMAN (Kurohashi and Nagao, 1998b) and KNP (Kurohashi and Nagao, 1998a) systems, respectively, and also the Japanese Goi-Taikei thesaurus (Ikehara et al., 1997) to measure conceptual distance. A parse tree is generated for the L1 component of each translation record, and also each input, and similarity gauged by both topological resemblance between parse trees and conceptual similarity between nodes of the parse tree.

Translation records used by the two systems were taken exclusively from the translation memory provided for the task.

In the proceeding sections, we briefly review the Japanese translation task (§ 2) and detail our particular use of the data provided for the task (§ 3). Next, we outline the lexical method (§ 4) and structural method (§ 5), and compare and discuss the performance of the two methods (§ 6).

2 Basic task description

The Japanese translation task data was made up of a translation memory and test set. The translation memory was dissected into 320 disjoint segments according to **headwords**, with an average of 21.6 translation records per headword (i.e. 6920 translation records overall). The purpose of the task was to select for a given headword which (if any) of the translation records gave a suitable translation for that word. The task stipulated that a maximum of one translation record could be selected for each input (allowing for the possibility of an `unassignable` output, indicating that no appropriate translation could be found). Translations were selected by way of a translation record ID, and systems were not required to actually identify what part of the L2 string in the selected translation record was the translation for the headword.

Translation records took the form of Japanese–English pairings of word clusters, isolated phrases, clauses or sentences containing the headword, at an average of 8.0 Japanese characters[1] and 4.0 English words per translation record. In some instances, multiple semantically-equivalent translations were given for a single expression, such as "corporation

[1] Ignoring punctuation but including each numeric digit as a single character.

which is in danger of bankruptcy" and "unsound corporation" for *abunai kigyō*; all such occurrences were marked by the annotator. For some other translation records, the annotator had provided a list of lexical variants or a paraphrase of the L1 expression to elucidate its meaning (not necessarily involving the headword), or made a note as to typical arguments taken by that expression (e.g. "refers to a person").

In the test data, inputs took the form of paragraphs taken from newspaper articles, within which a single headword had been identified for translation. The average input length was 697.9 characters, nearly 90 times the L1 component of each translation record. In its raw form, therefore, the translation task differs from a conventional translation retrieval task in that translation records and inputs are not directly comparable, in the sense that translation records are never going to provide a full translation approximation for the overall input.

3 Data preparation

In adapting the task data to our purposes, we first carried out limited normalisation of both the translation memory and test data by: (a) replacing all numerical expressions with a common NUM marker, and (b) normalising punctuation.

In order to maximise the disambiguating potential of the translation memory, we next set about automatically deriving as many discrete translation records as possible from the original translation memory. Multiple lexical variants of the same basic translation record (indexed identically) were generated in the case that: (a) a lexical alternate was provided (in which case all variants were listed in parallel); (b) a paraphrase was provided by the annotator (irrespective of whether the paraphrase included the headword or not); (c) syntactic or semantic preferences were listed for particular arguments in the basic translation record (in which case lexical variants took the form of strings expanded by adding in each preference as a string). At the same time, for each headword, any repetitions of the same L1 string were completely removed from the translation record data. This equates to the assumption that the translation listed first in the translation memory is the most salient or commonplace.

This method of translation record derivation resulted in a total of 152 new translation records, whereas the removal of duplicate L1 strings for a given headword resulted in the deletion of 670 translation records; the total number of translation records was thus 6402, at an average of 20.0 translation records per headword.

We experimented with a number of methods for abbreviating the inputs, so as to achieve direct comparability between inputs and translation records. First, we extracted the clause containing the headword instance to be translated. This was achieved through a number of ad hoc heuristics driven by the analysis of punctuation. These clause-level instances served as the inputs for the *structural* method. We then further "windowed" the inputs for the *lexical* method, by allowing a maximum of 10 characters to either side of the headword. No attempt was made to identify or enforce the observation of word boundaries in this process.

4 The lexical method

As stated above, the lexical method is based on character-based indexing, meaning that each string is naively treated as a sequence of characters. Rather than treat each individual character as a single segment, however, we chunk adjacent characters into bigrams in order to capture local character contiguity. String similarity is then determined by way of Dice's Coefficient, calculated according to:

$$sim_1(IN^*_m, TR_i) = \frac{2 \times \sum_{e \in IN^*_m, TR_i} \min\left(freq_{IN^*_m}(e), freq_{TR_i}(e)\right)}{len(IN^*_m) + len(TR_i)}$$

where IN^*_m is the abbreviated version of the input string IN_m (see above) and TR_i is a translation record; each e is a character bigram occurring in either IN^*_m or TR_i, $freq_{IN^*_m}(e)$ is defined as the weighted frequency of bigram type e in IN^*_m, and $len(IN^*_m)$ is the character bigram length of IN^*_m.[2] Bigram frequency is weighted according to character type: a bigram made up entirely of hiragana characters (generally used in functional words/particles) is given a weight of 0.2 and all other bigrams a weight of 1. Note that Dice's Coefficient ignores segment order, and that each string is thus treated as a "bag of character bigrams".

Our choice of the combination of Dice's Coefficient, character-based indexing and character bigrams (rather than any other n-gram order or mixed n-gram model) is based on the findings of Baldwin (2001b; 2001a), who compared character- and word-based indexing in combination with both segment order-sensitive and bag-of-words similarity measures and with various n-gram models. As a result of extensive evaluation, Baldwin found the combination of character bigram-based indexing and a bag-of-words method (in the form of either the vector space model or Dice's Coefficient) to be optimal. Our choice of Dice's Coefficient over the vector space model is due to the vector space model tending to blithely prefer shorter strings in cases of low-level character overlap, and the ability of Dice's Coefficient to pick up on subtle string similarities under such high-noise conditions.

Given the limited lexical context in translation records (8.0 Japanese characters on average), our method is highly susceptible to the effects of data sparseness. While we have no immediate way of reconciling this shortcoming, it is possible to make use of the rich lexical context of the full inputs (i.e. in original paragraph form rather than clause or windowed clause form). Direct comparison of the full

[2] $freq_{TR_i}(e)$ and $len(TR_i)$ are defined similarly.

inputs with translation records is undesirable as high levels of spurious matches can be expected outside the scope of the original translation record expression. Inter-comparison of full inputs, on the other hand, provides a primitive model of domain similarity. Assuming that high similarity correlates with a high level of domain correspondence, we can apply a cross-lingual corollary of the "one sense per discourse" observation (Gale et al., 1992) in stipulating that a given word will be translated consistently within a given domain. By ascertaining that a given input closely resembles a second input, we can use the combined translation retrieval results for the two inputs to hone in on the optimal translation for the two. We term this procedure **domain-based similarity consolidation**.

The overall retrieval process thus involves: (1) carrying out standard translation retrieval based on the abbreviated input, (2) using the original test set to determine the full input string most similar to the current input, and (3) performing translation retrieval independently using the abbreviated form of the maximally similar alternate input. Numerically, the combined similarity is calculated as:

$$sim_2(IN_m, TR_i) = 0.5 \left(sim_1(IN^*_m, TR_i) + \max_{n \neq m} sim_1(IN_m, IN_n)\ sim_1(IN^*_n, TR_i) \right)$$

where IN_m is the current input (full form), IN^*_m is the abbreviated form of IN_m, sim_1 is as defined above, and IN_n is any input string other than the current input. Note that the multiplication by 0.5 simply normalises the output of sim_2 to the range $[0, 1]$. For each input IN_m, the ID for that translation record which is deemed most similar to IN_m is returned, with translation records occurring earlier in the translation memory selected in the case of a tie.[3]

5 The structural method

The structural method contrasts starkly with the lexical method in that it is heavily resource-dependent, requiring a morphological analyser, parser and thesaurus. It operates over the same translation memory data as the lexical method, but uses only the abbreviated forms of the inputs (to the clause level) and does not consider inter-input similarity.

JUMAN (Kurohashi and Nagao, 1998b) is first used to segment each string (translation records and inputs), based on the output of which, the KNP parser (Kurohashi and Nagao, 1998a) is used to derive a parse tree for the string. The reason for abbreviating inputs only as far as the clause level for the structural method, is to enhance parseability. Further pruning takes place implicitly further downstream as part of the parse tree matching process.

KNP returns a binary parse tree, with leaves corresponding to optionally case-marked phrases. Each leaf node is simplified to the phrase head and the (optional) case marker normalised (according to the KNP output).

As for the lexical method, all translation records corresponding to the current headword are matched against the parse tree for the input, and the ID of the closest-matching tree returned. In comparing a given pair of parse trees T^1 and T^2, we proceed as follows in direction $d \in \{\text{up}, \text{down}\}$:

1. Set p^1 to the leaf node containing the headword in T^1, and similarly initialise p^2 in T^2; initialise n to 0
2. If $p^1_m \neq p^2_m$, return $(n, 0)$
3. If $p^1_f \neq p^2_f$, return $(n, concept\_sim(p^1_f, p^2_f))$
4. Increment n by 1, set p^1 and p^2 to their respective adjacent leaf nodes in direction d within the parse tree; goto step 2.

Here, p^i_m is the case marker associated with node p^i, p^i_f is the filler associated with node p^i, and the $\neq$ operator represents lexical inequality; *concept_sim* calculates the conceptual similarity of the two fillers in question according to the Goi-Taikei thesaurus (Ikehara et al., 1997). We do this by, for each sense pairing of the fillers, determining the least common hypernym and the number of edges separating each sense node from the least common hypernym. The conceptual distance of the given senses is then determined according to the inverse of the greater of the two edge distances to the hypernym node, and the overall conceptual distance for the two fillers as the minimum such sense-wise conceptual distance.

We match both up and down the tree structure from the headword node, and evaluate the combined similarity as the sum of the individual elements of the returned tuples. That is, if an upward match returned (i, m) and a downward match (j, n), the overall similarity would be $(i + j, m + n)$. The translation output is the ID of the translation record producing the greatest such similarity, where $(w, x) > (y, z)$ iff $w > y$ or $(w = y \wedge x > z)$. As a result, conceptual similarity is essentially a tie-breaking mechanism, and the principal determining factor is the number of phrase levels over which the parse trees match. In the case that there is a tie for best translation, the translation record with the longest L1 string is (arbitrarily) chosen, and in the case that this doesn't resolve the stalemate, a translation record is chosen randomly. In the case that all translation records score $(0, 0)$, we deem there to be no suitable translation in the translation memory, and return `unassignable`.

As mentioned in Section 2, crude selectional preferences (of the form PERSON or BUILDING) were provided on certain argument slots in trans-

[3] Based on the observation that translation records are roughly ordered according to commonality. Ties were observed 7.5% of the time, with the mean number of top-scoring translation records being 1.12.

Method	*Accuracy*
Lexical	49.1%
Structural	41.2%
Baseline	36.8%

Table 1: Results

lation records. These were supported by semi-automatically mapping the preference type onto the Goi-Taikei thesaurus structure, and modifying the $\neq$ operator to non-sense subsumption of the translation record filler by the input selectional preference, in step 3 of the parse tree match algorithm. Selectional preferences were automatically mapped onto nodes of the same name if they existed, and manually linked to the thesaurus otherwise.

6 Results and discussion

The translation retrieval accuracy for the two methods is given in Table 1, along with a baseline accuracy arrived at through random translation record selection for the given headword. Note that as we attempt to translate all inputs, the presented accuracy figures correspond to both recall and precision.

The most striking feature of the results is that the lexical method has a clear advantage over the structural method, while both methods outperform the baseline. Obviously, it would be going too far to discount structural methods outright based on this limited evaluation, particularly as the lexical method has undergone extensive testing and tuning over other datasets, whereas the structural method is novel to this task. It is surprising, however, that a technique as simple as the lexical method, requiring no external resources and ignoring even word boundaries and word order, should perform so well.

The main area in which the structural method fell short was `unassignable` inputs where no translation record displayed even the same case marking on the headword. Indeed 130 or 10.8% of inputs were tagged `unassignable`, despite them comprising only 0.3% of the solution set. Note, however, that even for only those inputs where the structural method was able to produce a match, the lexical method significantly outperformed the structural method (50.2% vs. 45.4%, respectively).

Conversely for the lexical method, at present, a translation record is selected irrespective of the magnitude of the similarity value, and it would be a trivial process to implement a similarity cutoff, below which an `unassignable` result would be returned. Preliminary analysis of the correlation between the lowest similarity values and inputs annotated as `unassignable` indicates that this method could be moderately successful (see Baldwin et al. (to appear)).

The translation task was designed such that participants didn't get access to annotated inputs until after the submission of final results, meaning that parameter settings and fine-tuning of techniques had to be carried out according to intuition only. Post hoc evaluation of methods such as domain-based similarity consolidation suggests that it does have a significant impact on system performance (Baldwin et al., to appear), although even in its basic configuration (using clause inputs and no domain-based similarity consolidation), the lexical method is superior to the structural method as presented herein.

In conclusion, this paper has served to describe each of a lexical and structural translation retrieval method, as applied to the SENSEVAL-2 Japanese translation task. The lexical method modelled strings as a bag of character bigrams, but incorporated a number of novel techniques including domain-based similarity consolidation in reaching a final decision as to the translation record most similar to the input. The structural method, on the other hand, compared parse trees and had recourse to conceptual similarity, but in a relatively rudimentary form. Of the two proposed methods, the lexical method proved to be clearly superior, although both methods were well above the baseline performance.

Acknowledgements

This paper was supported in part by the Research Collaboration between the Nippon Telegraph and Telephone Company (NTT) Communication Science Laboratories and CSLI, Stanford University.

References

T. Baldwin, A. Okazaki, T. Tokunaga, and H. Tanaka. to appear. The successes and failures of lexical and structural translation retrieval. In *Transactions of the IEICE.*

T. Baldwin. 2001a. Low-cost, high-performance translation retrieval: Dumber is better. In *Proc. of the 39th Annual Meeting of the ACL and 10th Conference of the EACL (ACL-EACL 2001)*, pages 18–25.

T. Baldwin. 2001b. *Making Lexical Sense of Japanese–English Machine Translation: A Disambiguation Extravaganza.* Ph.D. thesis, Tokyo Institute of Technology.

W. Gale, K. Church, and D. Yarowsky. 1992. One sense per discourse. In *Proc. of the 4th DARPA Speech and Natural Language Workshop*, pages 233–7.

S. Ikehara, M. Miyazaki, A. Yokoo, S. Shirai, H. Nakaiwa, K. Ogura, Y. Ooyama, and Y. Hayashi. 1997. *Nihongo Goi Taikei – A Japanese Lexicon.* Iwanami Shoten. 5 volumes. (In Japanese).

S. Kurohashi and M. Nagao. 1998a. Building a Japanese parsed corpus while improving the parsing system. In *Proc. of the 1st International Conference on Language Resources and Evaluation (LREC'98)*, pages 719–24.

S. Kurohashi and M. Nagao. 1998b. *Nihongo keitaikaiseki sisutemu JUMAN* [Japanese morphological analysis system JUMAN] version 3.5. Technical report, Kyoto University. (In Japanese).

Supervised Sense Tagging using Support Vector Machines

Clara Cabezas, Philip Resnik, and **Jessica Stevens**
Dept. of Linguistics and Institute for Advanced Computer Studies
University of Maryland, College Park, MD 20742 USA
{clarac,resnik,stevenjc}@umiacs.umd.edu

Abstract

We describe the University of Maryland's supervised sense tagger, which participated in the SENSEVAL-2 lexical sample evaluations for English, Spanish, and Swedish; we also present unofficial results for Basque. We designed a highly modular combination of language-independent feature extraction and supervised learning using support vector machines in order to permit rapid ramp-up, language independence, and capability for future expansion.

1 Introduction

The SENSEVAL-2 exercise provided an unprecedented opportunity to explore word sense disambiguation (WSD) in a common evaluation framework for a large number of languages. In past work, we have focused on unsupervised methods for English, taking advantage of the WordNet hierarchy and sometimes also selectional preferences between predicates and arguments (Resnik, 1997; Resnik, 1999). In the current exercise, however, WordNet-like sense hierarchies were not necessarily going to be available for all languages, and the predominance of lexical selection tasks (rather than all-words tasks) suggested adopting a disambiguation approach capable of exploiting manually annotated training data. These considerations motivated a system design based on supervised learning, where senses to be predicted did not need to be treated as part of a semantic hierarchy.

Our design was also motivated by the role of semantic selection techniques in our longer term research agenda. In the context of our group's work on cross-language information retrieval and machine translation applications (Resnik et al., 2001; Cabezas et al., 2001), lexical selection — that is, choosing the right target-language word given a source-language word in context — is a crucial task. Because the lexical selection problem is extremely similar to sense selection, and because this was our first foray into supervised methods, we took advantage of the opportunity to construct an architecture that will support both tasks.

In the sections that follow, we lay out our system architecture, briefly summarize our SENSEVAL-2 results, and discuss our plans for future work.

2 System Architecture

UMD's system follows the classic supervised learning paradigm that, for WSD, is perhaps best exemplified by Yarowsky's (1993) work. Each word in the vocabulary is considered an independent classification problem. First, annotated training instances for the ambiguous word are analyzed so that each instance can be represented as a collection of feature-value pairs labeled with the correct category. Then, these data are used for parameter estimation within a supervised learning framework in order to produce a trained classifier. Finally, the trained classifier is given previously unseen test instances and for each instance it predicts what the appropriate category label should be.

2.1 Contextual Features

We began by tokenizing all the training instances using a simple language-specific tokenizer. Features were then defined in terms of the presence of tokens either within a wide context or at a certain position to the right or left of the word being disambiguated.

In detail, let $\mathcal{T}$ be the set of unique tokens found in the full set of training data (all training instances), plus the special token UNKNOWN, which replaces any token in test data that was

never seen during training. Define $\mathcal{F}_{\text{wide}} = \mathcal{T}$. A feature $f \in \mathcal{F}_{\text{wide}}$ will be considered present and have a non-zero value if f appears anywhere in the wide context of the word being disambiguated. For example, if we were disambiguating the word *training* that appears in the first sentence of this paragraph, using the entire paragraph as the wide context, then there would be non-zero values for features WE, BEGAN, and every other word in the paragraph. That is, features correspond to surrounding words.[1]

Let $\mathcal{L} = \{L_3, L_2, L_1, R_1, R_2, R_3\}$, signifying the locations "three tokens to the left", "two tokens to the left", ..., "three tokens to the right", and define $\mathcal{F}_{\text{colloc}} = \{\text{l:t} \mid l \in \mathcal{L} \text{ and } t \in \mathcal{T}\}$. A feature l:t $\in \mathcal{F}_{\text{colloc}}$ will be considered present and have a non-zero value if token t appears at position l relative to the word being disambiguated. For example, if we were disambiguating the word *training* that appears in the first sentence of this section, there would be non-zero values for the features L_3:`tokenizing`, L_2:`all`, L_1:`the`, L_1:`instances`, L_2:`using`, and L_3:`a`.

2.2 Feature Weights

The value associated with each feature is a weight indicating how useful the feature is likely to be in disambiguation, analogous to the term weights used in representing documents as feature vectors for information retrieval.

In detail, let us designate the full feature set as $\mathcal{F} = \mathcal{F}_{\text{wide}} \cup \mathcal{F}_{\text{colloc}}$, and let $N_{\mathcal{F}} = |\mathcal{F}|$. Clearly some features are more useful than others. For example, the feature `into` (word *into* appearing anywhere in the context) is unlikely to help distinguish among senses, although the feature R_1:`into` (word *into* appearing one word to the right) might be useful for disambiguating among the senses of some verbs. In order to assign weights to features based on their likely utility, we follow a strategy similar to what is done in information retrieval, defining inverse category frequency (ICF), by analogy with inverse document frequency (IDF), as a function of how many distinct categories a feature appears with in training data.

Specifically, if we are disambiguating a word w with senses $\mathcal{S} = \{s_1, s_2, \ldots, s_{N_w}\}$, then we define $\text{ICF}_w(f) = -\log(N_w^f/N_w)$ where N_w^f is the number of distinct elements of $\mathcal{S}$ that ever co-occur with feature f in the training data for word w. For example, if a word has five senses, and the feature L_1:`the` appears in some training instance for each of the five senses, then $\text{ICF}_w(L_1\text{:the}) = -\log(5/5) = 0$, correctly indicating that this feature is not at all useful for disambiguating among the five senses of this word. The lower N_w^f is, the greater the value of the $\text{ICF}_w(f)$ value and hence the greater weight accorded this feature.

Training and test instances are represented as $N_{\mathcal{F}}$-ary feature vectors: given a training or test instance for a word w, the vector representation is defined by $v_w[f] = \text{ICF}_w(f)$ if $f \in \mathcal{F}$ is present, and zero otherwise.

2.3 Learning Framework

Once training and test instances are represented as feature vectors, it becomes possible to exploit any number of existing supervised learning algorithms. In general, such algorithms take a set $\{\langle v_1, c_1\rangle, \langle v_2, c_2\rangle, \ldots, \langle v_N, c_N\rangle\}$ of training instances, and produce a classifier that takes a feature vector v as input and return a distribution or confidence function over the possible categories.

For SENSEVAL-2, we selected support vector machines (SVMs) as the supervised learning framework. We were motivated by the fact that SVMs have been shown to achieve high performance and work efficiently in environments where there are very large numbers of features, and also by the existence of a good off-the-shelf implementation, SVM-Light, available for research purposes (Joachims, 1999; Joachims, 1998).[2]

SVM learning is appropriate for binary classification tasks, rather than the multi-way classification needed for disambiguating among n senses. For each word in the lexical sample tasks, therefore, we constructed a family of SVM classifiers, one for each of the word's N_w senses. All positive training examples for a

[1] For SENSEVAL-2, we defined the surrounding context for wide contexts as being anywhere within the test instance, because instances comprised only a sentence or two. In a more general setting the context could be defined as a window of ±50 words, ±100 words, the entire document, etc.

[2] Hearst (1998) presents a collection of brief and illuminating discussions of SVMs; see http://www.computer.org/intelligent/ex1998/pdf/x4018.pdf. SVM-Light is available at http://www-ai.cs.uni-dortmund.de/svm_light.

Language	Precision (%)	Recall (%)
English (coarse)	64.3	64.3
English (fine)	56.8	56.8
Spanish (fine)	62.7	62.7
Swedish (mixed)	65.6	65.6
Swedish (fine)	61.1	61.1
Basque (fine)	70.3	70.3

Table 1: UMD-SST lexical sample results

sense s_i of w were treated as negative training examples for all the other senses s_j, $j \neq i$.

In the testing phase, we convert test instances for word w into feature vectors, and we then we run these vectors through the SVM classifiers for $\{s_1, s_2, \ldots, s_{N_w}\}$. For each instance, we select the sense for which the SVM classifier's response is most strongly "yes" (or, equivalently, most weakly "no").

3 SENSEVAL-2 Results

Table 1 shows the performance of UMD's supervised sense tagger (UMD-SST) for the lexical sample tasks in four languages. The figures for English, Spanish, and Swedish are official SENSEVAL-2 results; the figures for Basque are unofficial results kindly computed by the Basque task organizers after SENSEVAL-2 because our Basque responses were not submitted in time for official evaluation.

In general, we were quite pleased with the results, particularly since this was our first time participating in SENSEVAL. UMD-SST turned in a solid performance in comparison with the baselines and other systems, with essentially no language-specific alterations necessary other than those required for tokenization. This enabled us to participate in system evaluation for more languages than any site except JHU. We consider this a good starting point for our further investigations, which we now briefly describe.

4 Future Work

Using the current system as a starting point, we are engaged in three lines of further investigation: linguistically richer contextual features, corpus-dependent expansion of feature vectors, and lexical selection via supervised learning.

In our preliminary tests using training and development data, we experimented first with using $\mathcal{F}_{\text{wide}}$ as the feature set, and obtained significant improvements when we added $\mathcal{F}_{\text{colloc}}$ in order to capture collocations and other local contextual features. In our follow-up efforts we plan to use broad-coverage parsing to create a set of features augmented further by grammatical relations, thus capturing collocations mediated by syntactic structure. For example, although our current feature vectors could not represent the presence of the word *tagger* as a nearby collocate of the word *describe* in the abstract of this paper, syntactically richer representations of this context for the verb *describe* would include the feature `object='tagger'`. Use of syntactic collocates will require broad-coverage parsing in all the languages of interest in order to identify grammatical relations; for this we will take advantage of our other work at Maryland on bootstrapping stochastic parsers for new languages using parallel corpora (Cabezas et al., 2001).

In our preliminary efforts we were not surprised to find that sparseness of data was a problem. Although we expect that some improvements may be obtained by collapsing across word variants — e.g. via morphological equivalence classes or stemming — we also plan to focus our efforts on semantic expansion, using document expansion techniques we have developed in our research on cross-language information retrieval (Levow et al., 2001). We have implemented a variant of the architecture in which training contexts are used as queries to a comparable corpus in order to retrieve related documents. The features from these documents are then added to the context representations, providing semantically enhanced feature vectors. Evaluation of this approach using SENSEVAL data is in progress.

Our third avenue of investigation focuses on the use of our supervised WSD infrastructure to address problems of lexical selection in machine translation. Empirically, there is a close relationship between sense distinctions and patterns of lexicalization across languages (Resnik and Yarowsky, 1999). And operationally, there is no real difference between labeling a word with a sense tag from a monolingual dictionary and labeling that word with a translation from a bilingual dictionary. Using WSD techniques for lexical selection primarily requires solving two

problems. The first problem is acquisition of annotated training data, and in this case large corpora of translation-labeled words in context can be created by obtaining parallel corpora, performing word-level alignment, and labeling each word with its correspondent in the other language; this problem is already solved as part of our infrastructure for research on statistical machine translation (Cabezas et al., 2001). The second problem is one of scalability: the approach we have described requires a separate classifer for every sense (or, now, every possible word-level translation) of every source language word. This remains an open issue, but we are optimistic about rapid developments in this area since scaling up to large vocabularies is a problem shared by everybody who wishes to use supervised WSD techniques in a broad-coverage setting.

5 Conclusions

University of Maryland's sense tagger represents a classic instance of the supervised learning approach. At the same time, we have made architectural choices that promote language independence, modularity, extensibility, and scalability, and in a relatively short time period we succeeded in putting together an implementation that performs quite credibly among an impressive collection of competitors. We are encouraged by the results and we look forward to participating in further SENSEVAL exercises.

Acknowledgements

This work was supported in part by Department of Defense contract MDA90496C1250 and DARPA/ITO Cooperative Agreement N660010028910. We're very grateful to all the SENSEVAL-2 organizers and task organizers for their hard work, to Thorsten Joachims for making SVM-Light available, and to David Martinez for computing our results for Basque.

References

Clara Cabezas, Bonnie Dorr, and Philip Resnik. 2001. Spanish language processing at University of Maryland: Building infrastructure for multilingual applications. In *Proceedings of the Second International Workshop on Spanish Language Processing and Language Technologies (SLPLT-2)*, Jaen, Spain, September.

Marti A. Hearst. 1998. Trends and controversies: Support vector machines. *IEEE Intelligent Systems*, 13(4):18–28.

Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the European Conference on Machine Learning*. Springer.

Thorsten Joachims. 1999. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press.

Gina-Anne Levow, Douglas Oard, and Philip Resnik. 2001. Rapidly retargetable interactive translingual retrieval. In *Human Language Technology Conference (HLT-2001)*, San Diego, CA, March.

Philip Resnik and David Yarowsky. 1999. Distinguishing systems and distinguishing senses: New evaluation methods for word sense disambiguation. *Natural Language Engineering*, 5(2):113–133.

Philip Resnik, Douglas Oard, and Gina Levow. 2001. Improved cross-language retrieval using backoff translation. In *Human Language Technology Conference (HLT-2001)*, San Diego, March.

Philip Resnik. 1997. Selectional preference and sense disambiguation. In *ANLP Workshop on Tagging Text with Lexical Semantics*, Washington, D.C., April.

Philip Resnik. 1999. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research (JAIR)*, 11:95–130.

David Yarowsky. 1993. One sense per collocation. ARPA Workshop on Human Language Technology, March. Princeton.

Probabilistic Network Models for Word Sense Disambiguation

Gerald Chao and **Michael G. Dyer**
Computer Science Department,
University of California, Los Angeles
Los Angeles, California 90095
gerald@cs.ucla.edu, dyer@cs.ucla.edu

Abstract

We present the techniques used in the word sense disambiguation (WSD) system that was submitted to the SENSEVAL-2 workshop. The system builds a probabilistic network per sentence to model the dependencies between the words within the sentence, and the sense tagging for the entire sentence is computed by performing a query over the network. The salient context used for disambiguation is based on sentential structure and not positional information. The parameters are established automatically and smoothed via training data, which was compiled from the SemCor corpus and the WordNet glosses. Lastly, the One-sense-per-discourse (OSPD) hypothesis is incorporated to test its effectiveness. The results from two parameterization techniques and the effects of the OSPD hypothesis are presented.

1 Problem Formulation

WSD is treated in this system as a classification task, where the i^{th} sense ($W\#i$) of a word (W) is classified as the correct sense tag (M_i), given the word W and usually some surrounding context. In the SENSEVAL-2 English all-words task, all ambiguous content words (nouns, verbs, adjectives, and adverbs) are to be classified with a sense tag from the WordNet 1.7 lexical database (Miller, 1990). For example, the words "great", "devastated", and "region" in the sentence "The great hurricane devastated the region" are classified with the correct sense tags 2, 2, and 2, respectively. We will refer to this task using the following notation:

$$\tilde{M} = M_{best}(S) = arg\ max P(M|S), \qquad (1)$$

where S is the input sentence, and M is the semantic tag assigned to each word. While a context larger than the sentence S can be and is used in our model, we will refer to the context as S. In this formulation, each word W_i in the sentence is treated as a random variable M_i taking on the values $\{1..N_i\}$, where N_i is the number of senses for the word W_i. Therefore, we wish to find instantiations of M such that $P(M|S)$ is maximized.

To make the computation of $M_{best}(S)$ more tractable, it can be decomposed into $M_{best}(S) \approx arg\ max(\Pi_i P(M_i|S))$, where it is assumed that each word can be disambiguated independently. However, this assumption does not always hold, since disambiguating one word often affects the sense assignment of another word within the same sentence. Alternatively, the process can be modeled as a Markov model, e.g., $M_{best}(S) \approx arg\ max(\Pi_i P(W_i|M_i) \times P(M_i|M_{i-1}))$. While the Markov model requires fewer parameters, it is unable to capture the long-distance dependencies that occur in natural languages. Although the first decomposition better captures these dependencies, computing $P(M_i|S)$ using the full sentential context is rarely used, since the number of parameters required grows exponentially with each added context. Therefore, one can further simplify this model by narrowing the context to $2n$ number of surrounding words, i.e., $P(M_i|S) \approx P(M_i|W_{i-n}, ...W_{i-1}, W_{i+1}, ...W_{i+n})$. However, narrowing the context also discards long-distance relationships, making it closer to a Markov model.

Without having to artificially limit the size of the context, another possible simplification is to make independence assumptions between the context words. In the simplest case, every context is assumed to be independent from each other, i.e., $P(M_i|S) \approx \Pi_x P(M_i|W_x)$, like a Naive Bayes classifier. While the parameters can be simply established by a set of bi-grams, the independence assumption is often too strong and thus negatively affects accuracy. The difficulty is in choosing the context that would maximize the accuracy while allowing for reliable parameter estimation from training data.

In our model, we aim to strike this balance by choosing the context words based on *structural* information, rather than positional information. The hypothesis is that an ambiguous word is probabilistically dependent on its structurally related words and is independent of the rest of the sentence. Therefore, long-distance dependencies can still be captured, while the context is kept small. Further-

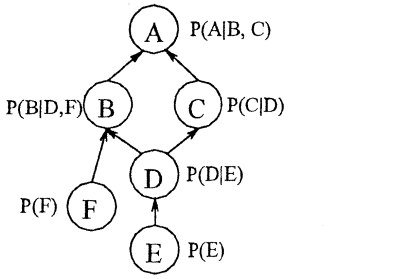

Figure 1: An example of a Bayesian network and the probability tables at each node that define the relationships between a node and its parents. The equation at the bottom shows how the distribution is represented by the network.

more, each word is not classified independently of each other, but is computed as one single query that determines all of the sense assignments that result in the highest overall probability for the whole sentence. Therefore, our model is a combination of the decompositions described above, by selectively making independence assumptions on a per-word basis to best model $P(M_i|S)$, while computing $M_{best}(S)$ in one query to allow for interactions between the word senses M_i.

1.1 Bayesian Networks

This process is achieved by using Bayesian networks to model the dependencies between each word and its contextual words, and based on the parameterization, compute the best overall sense assignments. A Bayesian network is a directed acyclic graph G that represents a joint probability distribution $P(X_1, ..., X_n)$ across the random variables of each node in the graph. By making independence assumptions between variables, each node i is conditionally dependent upon only its parents PA_i (Pearl, 1988): $P(X_1, ..., X_n) = \Pi_i P(X_i|PA_i)$. By using this representation, the number of probabilities needed to represent the distribution can be significantly reduced. Figure 1 shows an example Bayesian network representing the distribution P(A,B,C,D,E,F). Instead of having one large table with 2^6 parameters (with all Boolean nodes), the distribution is represented by the conditional probability tables (CPTs) at each node, such as P(B | D, F) at node B, requiring a total of only 24 parameters for the whole distribution. Not only do the savings become more significant with larger networks, but the sparse data problem becomes more manageable as well. The training set no longer needs to cover all permutations of the feature sets, but only smaller subsets dictated by the sets of variables of the CPTs.

In our model using Bayesian networks for WSD, each word is represented by the random variable M_i as a node in G. We then find a set of parents PA_i that M_i depends on, based on structural information. Using this representation, the number of parameters is significantly reduced. If the average number of parents per node is 2, and if the average number of senses per word is 5, then the joint distribution across the whole sentence $P(M_1, .., M_N)$ is represented by the Bayesian network with $\approx 5^{(2+1)} * N$ parameters. This is in contrast to a full joint distribution table that would contain 5^N entries, which is obviously intractable for any sentence of non-trivial length N. Bayesian networks also facilitate the computation of the instantiations for M_i such that $P(M_1, .., M_N)$ is maximum. Instead of looking for the maximum row in the table with 5^N entries, this computation is made tractable by using Bayesian networks. Specifically, this query, called Maximum A Posteriori (MAP), can be computed in $O(5^w)$, where $w << N$ and indicates the connectiveness of G.

Using the same notation above, the process of a whole-sentence word sense disambiguation using probabilistic networks can be described as the following:

$$M_{best}(S) \approx arg\ max \Pi_i P(M_i|W_i, W_{PA_i}, M_{PA_i})$$

$$\approx arg\ max \Pi_i (P(M_i|M_{PA_i}) P(M_i|W_i, W_{PA_i})). \quad (2)$$

The first approximation is based on our hypothesis of a word's sense is dependent only on structurally related words. It is further decomposed in the second term to minimize the sparse data problem. This process consists of three major steps: 1) defining the structure of the Bayesian network G, 2) quantifying the network with probabilities from training data ($P(M_i|W_i, W_{PA_i})$), and finally, 3) answering the query of the most probable word sense assignments ($arg\ max \Pi_i(...)$).

2 Network Structure

The first step in constructing a Bayesian network is to determine its structure G, which defines each node's dependency relationship with the rest of the network. In our model, we are making these independence assumptions based on the structural relationships between words. Specifically, given the sentence S and its parse tree, we automatically construct a graph G by first creating a node M_i for each word W_i. This process is best illustrated by the example shown in Figure 2. For each node M_i, an edge is added to node M_x, where M_x is the head word of a verb phrase (board $\rightarrow$ approved), the target of the modifier M_i (today's $\rightarrow$ meeting), or the preposition M_x where M_i is the target or a constituent of the prepositional phrase (approved $\rightarrow$ at). One can see that if the parse tree is known, the construction of network G is straight-forward. For SENSEVAL-2, the

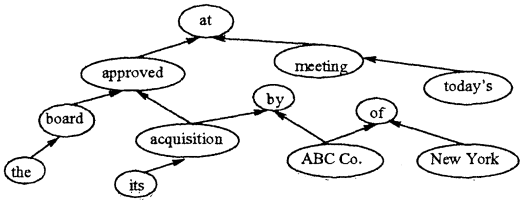

Figure 2: An example of a Bayesian network representing the inter-dependencies between the words of the sentence "The board approved its acquisition by ABC Co. of New York at today's meeting."

parse trees provided in Treebank format were used to build the Bayesian networks' structure.

Once the structure of the Bayesian network is determined, the context, i.e., the parents PA_i, for each word is established. Using the same example, the context for the word "approved" is "board" and "acquisition", and for "at" it is "approved" and "meeting". Our hypothesis is that these structurally related words, among all of the words within the sentence, provide the best contextual information for sense disambiguation. That is, given that the parents' word form W_{PA_i} and senses M_{PA_i} are known, the sense assignment for M_i is independent of all other words in the sentence. This is, of course, a simplification due to the constraint in minimizing the context. However, the use of Bayesian networks allows for easy expansion of context by establishing more edges between nodes or adding new nodes, provided that the parameters can be determined reliably.

3 Establishing the Parameters

Once G is determined, the CPTs at each node need to be quantified. Using the same example above, for the word "approved", its CPT $P(approved\#i|board\#i, acquisition\#i)$ would contain 2 (number of senses for "approved") $\times$ 9 $\times$ 4 = 72 entries. For a word without any parents, such as "today's", its priors are used.

While determining the network structure is relatively simple, establishing accurate parameters is quite difficult, even with a small context such as ours. Due to the limited size of SemCor, our only labeled training data, we used additional sources to quantify and smooth these parameters. Primarily we deployed the same techniques used in our Bayesian Hierarchical Disambiguator (BHD) model (Chao and Dyer, 2000), which uses Internet search engines to estimate parameters based on permutations of synonym words, a method first introduced by Mihalcea and Moldovan (1999). These parameters are then smoothed by training data obtained from SemCor. The details of BHD are omitted here due to space constraints.

Although BHD was only used on adjective-noun pairs, the same principles are used to quantify all of the CPTs in this model. While only one hierarchical network is needed to smooth the parameter for adjective-noun pairs, up to three hierarchical networks are used for each potential parent. Since the smoothing computation is very efficient, being linear in the depth of the network, these additions did not impact the speed of the model. The majority of the time was used to query the Internet search engine.

The BHD model, however, did use additional training data that was collected from the WordNet glosses and manually annotated. While it resulted in good accuracy, this was obviously not an option for SENSEVAL-2. Instead, the example sentences from WordNet are extracted and first tagged by Brill's POS tagger (Brill, 1995). Then an experimental parser and our WSD system were used to parse and disambiguate the sentences to extract additional training data. For example, for the 6th sense of adjective "great", the pair "great#6 time" is extracted from the example sentence fragment "had a great time at the party" and automatically disambiguated. The labeled pair is then added to the training set for great#6.

Lastly, the priors in this model are determined directly from SemCor's occurrence statistics and estimated using Maximum Likelihood Estimation (MLE). This is another simplification over the BHD model, where the priors were determined using the hundred most frequent adjective-noun pairs culled from the Internet and then manually classified. It is well known that MLE is inaccurate when the number of events are low, as is in this case when rarer senses often have only single occurrences.

Nevertheless, we are able to address both of the manual steps used in the BHD model with automated processes. However, it is our belief that they are also the weakest part of our model and contribute the most to the errors.

4 Querying the Network

With both the structure G and the parameters established, the query we pose is to compute the instantiations for each random variable that would result in the highest joint probability, i.e., $arg\ max P(M_i|S)$. This is computed easily using the Maximum A Posteriori (MAP) query. This was implemented using the JointTree algorithm (Darwiche, 1995) and can be computed in $O(|c|^w)$ time, where $|c|$ is the size of the variable (number of senses), and w is the tree width. Given that our networks are sparsely connected, w is usually close to 3, the average number of parents + 1.

The advantage of using the MAP query is that it computes variable instantiations that will maximize the *overall* probability across the whole sentence, rather than the localized context. Further-

Model	Precision	Recall
1	0.500	0.449
2	0.475	0.454
3	0.474	0.453

Table 1: Precision/recall results of the three models submitted to SENSEVAL-2.

more, the resulting instantiation and probability is guaranteed to be maximum. So given the independence assumptions made on the context and the estimated parameters, MAP will always produce the most probable sense tagging for every word in the sentence.

5 Beyond Sentential Context

It is well known that word senses are often influenced by contexts larger than the sentence, such as surrounding sentences or even the whole passage. We experimented with the One-sense-per-discourse (OSPD) hypothesis (Yarowsky, 1993) by applying the probabilities described in Stetina et al. (1998) to words that have previously appeared in the text and thus have been disambiguated. The only modification needed to our model described thus far is to apply OSPD probabilities, which is dependent on the distance between the sentences, to each sense of a re-occurring word before the MAP query. It is our observation that this incarnation of the OSPD hypothesis, chosen for its ease of implementation, tends to propagate erroneous sense tagging from initial sentences to the remainder of the passage. A better approach would be to determine the one sense that would maximize the consensus across the whole passage, as well as within each individual sentence. How this can be achieved efficiently in a probabilistic framework is currently being investigated.

6 Evaluation

For SENSEVAL-2, we submitted three models for comparison, which differ by their methods of parameter estimation. Model 2 uses the training data from SemCor and Hierarchical networks to smooth the parameters from Internet search engines. Model 3 incorporates additional training data gathered automatically from the WordNet glosses. Lastly, model 1 combines all training data, as well as the OSPD hypothesis.

One can see that the model that uses all of the available data achieved best accuracy (model 1) but unfortunately also had the lowest recall due to the added complexity. Some highly polysemous words were omitted due to time and memory constraints. Between the 2 training sets, it was unfortunate that the addition of the automatically generated training set reduced the accuracy slightly, mainly due to the noisy data produced by our experimental system.

Nevertheless, we believe that there is a wealth of information contained within WordNet's glosses. Since one of our aims is to use as much automated processing as possible, we are focusing on improving the accuracy of the automatically generated training data. Our goal is that as the WSD accuracy of our system improves, so will the reliably of these automatically generated training data. Having improved training data will further improve the system's WSD accuracy, i.e., a bootstrapping system. We are at the initial stage of this process, but some fundamental problems such as reliable POS tagging and parsing of sentence fragments need to be addressed first. Furthermore, parameter estimation based on Internet statistics might prove to be too noisy, so we are currently focusing on learning algorithms such as Expectation Maximization to tune the parameters. Lastly, if our context is found to be too limited, additional features can be added to the Bayesian networks to improve the classification accuracy.

References

Eric Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics*, 21:722–727.

Gerald Chao and Michael G. Dyer. 2000. Word sense disambiguation of adjectives using probabilistic networks. In *Proceedings of the Eighteenth International Conference on Computational Linguistics.*

Adnan Darwiche. 1995. Conditional algorithms for exact and approximate inference in causal networks. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, pages 99–107.

Sadao Kurohashi Jiri Stetina and Makoto Nagao. 1998. General word sense disambiguation method based on a full sentential context. In *Proceedings of COLING-ACL Workshop on Usage of WordNet in Natural Language Processing, Montreal, Canada*, pages 1–8, July.

Rada Mihalcea and Dan Moldovan. 1999. A method for word sense disambiguation of unrestricted text. In *Proceedings of the 37th Annual Meeting of the ACL*, pages 152–158, Maryland, NY, June.

G. Miller. 1990. WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4).

Judea Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, CA.

David Yarowsky. 1993. One sense per collocation. In *Proceedings of ARPA Human Language Technology, Princeton*, pages 266–271.

Improving WSD with Multi-Level View of Context Monitored by Similarity Measure

E. Crestan[1,2], M. El-Bèze[1] and C. de Loupy[2]

(1) Laboratoire d'Informatique d'Avignon
339 ch. Des Meinajaries, BP 1228
F-84911 Avignon Cedex 9
{eric.crestan, marc.elbeze}@lia.univ-avignon.fr

(2) Sinequa
51-59 rue Ledru Rollin
F-94200 Ivry-sur-Seine
{crestan, loupy}@sinequa.com

Abstract

The approach presented in this paper for Word Sense Disambiguation (WSD) is based on a combination of different views of the context. Semantic Classification Trees (SCT) are employed over a short and a multi-level view of context, including rough semantic features, while a similarity measure is used in some particular cases to rely on a larger view of the context. We also describe our two-step approach based on HMM for the *all-word* task.

Introduction

In the tracks of SENSEVAL-1 (Kilgarriff and Rosenzweig, 2000), the second edition of the word sense disambiguation evaluation campaign offers a new set of words to test improvements in the domain of WSD. It also includes a new task, aimed at disambiguating each word of a text.

Our approach for the lexical sample task is based on three different views of the context, which allows us to consider more information for sense tagging. In order to deal with short-range view of the context, we have chosen to use Semantic Classification Trees (SCT) (Kuhn and De Mori, 1995), which are binary decision trees. Moreover, based on our experience, we will show, that using rough semantic features as a higher-level view of the context yields substantial increases in performance. Finally, a similarity distance is employed in order to capture longer-range context information.

The paper is organized as follows: in the first part (Section 1), the work we have done on the *lexical sample* task is presented. This part includes a brief overview of the SCT approach (Section 1.1) and we show how the coverage it yields could be increased while using more or less rough semantic features thanks to a multi-level view of the context (Section 1.2). In Section 1.3, we propose to use a similarity measure like those used in document retrieval in order to select a sense among those proposed by the SCT systems. The second part (Section 2) is dedicated to the *all-words* task. A two-step approach based on a trisem-bisem model is presented (Section 2.1). Then, we propose to apply a special process on the most frequent words in the task (Section 2.2). In conclusion, the results for both tasks are presented.

1 Lexical Sample Task

The *lexical sample* task of SENSEVAL-2 is composed of 29 nouns, 29 verbs and 15 adjectives in context. We decided to handle the totality of the words, and always assign one and only one sense to each test word (*recall = precision*). For training purpose, we used the corpus supplied for each word to be disambiguated. However, the number of training sentences supplied was greatly reduced compared to that of the first SENSEVAL exercise. By comparison, the average number of training sentences for the nouns in SENSEVAL-1 data was about 410 sentences/word. Here, the average number of training sentences is only 121 sentences/word. This difference leads us to believe that the present evaluation may be much harder than the previous one. The senses used for this evaluation come from the Wordnet 1.7 pre-release (Miller *et al.*, 1990).

1.1 Applying SCT to WSD

Yarowsky (1993) states that most clues for the purpose of disambiguation are present in a micro-context of *3* or *4* words. SCT seems to be an adequate approach to handle short contexts. Moreover, SCT, which are binary decision trees, permit a simple interpretation of the results, by recovering the successive questions asked along each path from the root to a leaf. Kuhn and

De Mori (1995) have shown that these extracted rules correspond to regular expressions. However, this approach requires a certain amount of data in order for the trees to be grown with reliable questions in its nodes.

Relying on previous work in this field (Loupy *et al.*, 2000), the training corpus was used to build one tree for each word to be disambiguated. While growing the trees, the list of possible questions is built at each node, taking into consideration the position of an element of the context (lemma in this case). The Gini impurity $G(X)$ (Breiman *et al.*, 1984) is then computed (*formula 1*) for each question in the list, in order to extract the one which generates the highest decrease in impurity ΔG_q (*formula 2*).

$$G(X) = 1 - \sum_{s \in S} P(s/X)^2 \quad (1)$$

Where $P(s/X)$ is the probability of sense s given population X,

$$\Delta G_q = G(T) - p_{Yes_q} G(Yes_q) + p_{No_q} G(No_q) \quad (2)$$

Here Yes_q and No_q correspond respectively to the population answering *yes* or *no* to the question q; p_{Yes_q} (respectively p_{No_q}) is the proportion of population T answering *yes* (respectively *no*) to question q.

A more detailed description of our approach to SCT can be found in Crestan and El-Bèze (2001).

The data had to be pre-processed before they could be used. Motivated by conclusions drawn from recent work (see for example Loupy and El-Bèze (2000)), the context was lemmatized, except for the word to be disambiguated. The determiners, possessive pronouns, adverbs and adjectives were removed, because they bring more noise to the tree growing process than they help capture relevant clues. However, some adjectives were preserved, when they were part of a compound noun, as in "*short circuit*". For the part-of-speech (POS) tagging process and lemmatization process, the English Tree-Tagger (Schmid, 1994) was used.

1.2 Rough semantic features as a multi-level view of context

Regarding previous work using SCT, the novelty of our approach consists in the introduction of rough semantic features into the context in order to increase the coverage of the trees. The process of tree growing can quickly suffer from lack of data. The ability of our system to view the context, not only as a succession of lemmas, but also as a multi-level view makes it more robust and reliable.

We used the Semantic Classes (SC) proposed in Wordnet in order to improve the coverage of the trees. There are *26* SC associated with nouns (e.g. <noun.body> for body related nouns) and *15* SC associated with verbs (e.g. <verb.motion> for motion related verbs). Because most of the adjectives and adverbs were removed during the pre-processing phase and because they have only one or two possible SC, their respective SC are not employed.

During the SCT building process, there is now not just one question to ask at a given position in a training sentence, but *n+1* (where *n* is the number of possible SC associated with a lemma). For example, the sentence sample in *figure* 1 leads to 16 possible questions if considering SC, and only 7 questions if considering only lemmas.

Figure 1: Example of SC usage

SC are added regardless of the POS. In the example above, the term *offer* can only be a verb, but we still associate with it the classes *_04* (noun.act) and *_10* (noun.communication), which are associated with the noun-senses of *offer*. There are two reasons for this choice: First, in the case of erroneous POS tagging, we would not be able to characterize a sense using the adequate SC. Second, tests have shown that results obtained using POS related SC or all the SC are comparable. This last point could be explained by the aptitude of SCT to select the best questions. Therefore, SCT are able to partially disambiguate the local context at a coarse-grained sense level when enough data are available. Consequently, it seems useless to make assumptions about POS.

Experiments carried on the SENSEVAL-1 data, has shown an improvement of about 2.5% on nouns and about 3% on verbs when using the Semantic Classes.

1.3 Similarity measure for a long-range view of the context

Experience has shown us that a window size of *WS*=3 is enough for disambiguation in many cases, but there are still numerous cases for which a larger window size is required. However, if a larger window size can provide more information for sense detection, it may also add more noise. In order to cope with this drawback, a similarity measure is employed (a technique usually applied in the field of document retrieval), as a ruler to decide which sense seems the more likely, considering the whole sentence (Figure 2). Firstly, three different Window Sizes (*WS*) are considered and run through the appropriate SCT process (trained on the same WS). Secondly, for each sense proposed by the SCT systems (E_1, E_2, and E_3), a pseudo-document is built with the corresponding sentences from the training corpus. Then, a similarity measure as those used in document retrieval is computed between the test sentence (*WS*=|*S*|) and each of the pseudo-documents (*i.e.* senses). Finally, only the sense having the best score is kept. The similarity measure used here is the Cosine measure (Salton and McGill, 1983).

The analysis of the results has shown that monitoring several SCT based views of the context by using the here above described technique leads to an average precision improvement of about 2%.

2 All-Words Task

The second task proposed in SENSEVAL consists in tagging almost all the words of a text. This is a more difficult task because in the first one, only some words have to be studied, whereas the behavior of all words must be known in order to correctly tag an entire text. Hidden Markov Models (HMM) have shown their efficiency in many NLP domains: part-of-speech tagging (El-Bèze and Merialdo, 1999), speech recognition (Jelinek, 1998), etc. Moreover, they have been used in semantic disambiguation with some success (Loupy *et al.*, 1998). Therefore, we decided to use this method for the all words task.

The test corpus supplied is composed of *2473* words to be disambiguated out of *5836* words. All POS are represented: *1140* nouns, *544* verbs, *453* adjectives and *299* adverbs (according to the supplied TreeBank-tagged file).

2.1 A coarse to fine-grained sense strategy

In a previous experiment (Loupy *et al.*, 1998), HMM were applied directly to disambiguate senses at fine-grained level using a unisem-bisem model, after training on the SemCor (Miller *et al.*, 1993). However, even if this method achieves correct results (72 % of correct assignation), it does not really improve

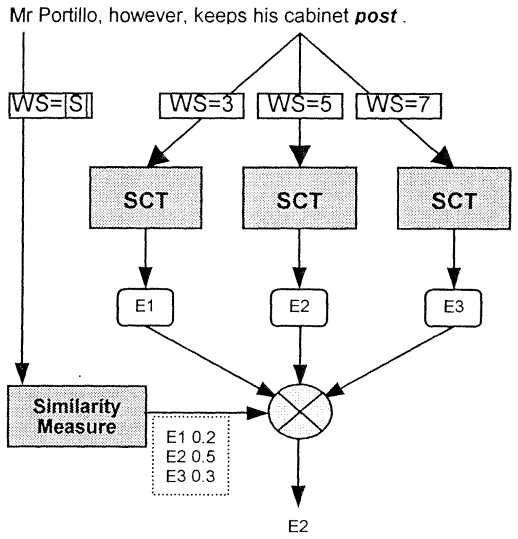

Figure 2: Sense selection using a similarity measure

over the unisem model. Therefore, it is recognized that there are not enough data to correctly learn the transitions between senses.

On the other hand, an HMM unisem-bisem model brings a slight improvement as compared to unisem alone when applied to a coarser semantic level, that is SC (Loupy *et al.*, 1998). We adopt the following two-step strategy:

- Firstly, determine the SC associated with each word in the text (*formula 4)*

$$\begin{aligned}\tilde{G} &= Arg\,\underset{G}{Max}\left[P(G/L)\right] \\ &= Arg\,\underset{G}{Max}\left[P(L/G)P(G)\right]\end{aligned} \quad (4)$$

  where G is the set of possible coarse-grained semantic classes associated to the lemma L.

- Secondly, assign the most probable fine-grained sense according to the word and the previously retrieved SC (*formula 5*).

$$\tilde{S} = Arg\,\underset{S}{Max}\left[P(S/G,L)\right] \quad (5)$$

  where S is the set of possible senses associated with the lemma L and its possible semantic classes G.

To cope with the well-known sparse data problem, some assumptions allow us to use a HMM (trisem-bisem model), in order to estimate *P(G)* (*formula 6*) and *P(L/G)* (*formula 7*).

$$P(G) \approx \prod_i \lambda \times P(g_i | g_{i-2}, g_{i-1}) + (1-\lambda) \times P(g_i | g_{i-1}) \quad (6)$$

and

$$P(L|G) \approx \prod_i P(l_i | g_i) \quad (7)$$

In the same way, assumptions were made in order to estimate the probability *P(S/G,L)* (formula 8).

$$P(S|G,L) \approx \prod_i P(s_i | g_i, l_i) \quad (8)$$

2.2 Using Lexical Sample Task Experience

In view of our experience with the lexical sample task, we decided to take advantage of it. The most frequent words among those to be disambiguated in the all-words task and which were also present in the SENSEVAL-2 lexical sample task were extracted. For those words, the technique presented in Section 1 was applied. In this way, 4 verbs (*call*, *develop*, *find* and *use*) and 2 nouns (*child* and *church*) were disambiguated by the SCT-Cosine method, as described in Section 1.3.

Results and Conclusion

As mentioned in section 1, the scores for the second edition of the *lexical sample* task are much lower than for the first edition (about 20%). However, our system achieved satisfactory results comparing to other participants (see table 1) and even accessed the top-5 systems. The use of SC as a multi-level view of the context has generated significant improvements in the results. As well as, the combination of different window sizes using similarity measure on a larger context as a judge has shown noticeable improvements.

	Lexical Sample		All-Words	
	Precision	Recall	Precision	Recall
Fine	61.3%	61.3%	61.8%	61.8%
Coarse	68.2%	68.2%	62.6%	62.6%

Table 1: Results for fine and coarse-grained senses

For the *all-words* task, our system has proven to be one of the bests, achieving an average precision/recall of 61.8%, and this, despite the absence of mapping between Wordnet 1.6 senses used for training purpose (SemCor) and Wordnet 1.7 senses used as test references.

References

L. Breiman, J. Friedman, R. Olshen, and C. Stone (1984): "*Classification and Regression Trees*", Wadsworth.

E. Crestan and M. El-Bèze (2001): "*Improving Supervised WSD by Including Rough Semantic Features in a Multi-Level View of the Context*", SEMPRO-2001 Workshop, Edinburgh. http://www.lia.univ-avignon.fr/publications/fich_art/LIA-SEMPRO-2001.pdf

M. El-Bèze and B. Mérialdo (1999): "*HMM Based Taggers*", in Syntactic Wordclass Tagging, ed. Hans Van Halteren, Kluwer Academic Publishers, Text and Language Technology, pp 263-284.

F. Jelinek (1998): "*Statistical Methods for Speech Recognition*", MIT Press, Cambridge.

A. Kilgarriff and J. Rosenzweig (2000): "*English SENSEVAL: Report and Results*", In Proc. LREC, Athens, Greece, Vol 3, pp 1239-1244.

R. Kuhn and R. De Mori (1995): "*The Application of Semantic Classification Trees to Natural Language Understanding*", IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(5), pp 449-460.

C. de Loupy and M. El-Bèze (2000): "*Using Few Clues can compensate the small amount of resources available for Word Sense Disambiguation*", LREC, Athens, Vol 1, pp 219-223.

C. de Loupy, M. El-Bèze and P.-F. Marteau (1998): "*Word Sense Disambiguation using HMM Tagger*", LREC, Grenade, Vol 2, pp. 1255-1258.

C. de Loupy, M. El-Bèze and P.-F. Marteau (2000): "*Using Semantic Classification Trees for WSD*", Computer and the Humanities, N° 34, Kluwer Academic Publishers, pp 187-192.

G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller (1990): "*Introduction to WordNet: An on-line lexical database*," International Journal of Lexicography, vol. 3(4), pp 235-244.

G. A. Miller, C. Leacock, R. Tengi, and T. Bunker (1993): "*A Semantic Concordance*", Proceedings of ARPA Workshop on Human Language Technology, Plainsboro, New Jersey, pp 303-308.

G. Salton and M.J. McGill (1983): "*Introduction to Modern Information Retrieval*", McGraw-Hill, New York.

H. Schmid (1994): "*Probabilistic Part-of-Speech Tagging Using Decision Trees*". In Proceedings of the Conference on New Methods in Language Processing. Manchester, UK, pp 44-49.

D. Yarowsky (1993): "*One sense per collocation*", In Proceedings of the ARPA Workshop on Human Language Technology, pp 266-271.

Using LazyBoosting for Word Sense Disambiguation

G. Escudero, L. Màrquez and G. Rigau
TALP Research Center
Universitat Politècnica de Catalunya
Jordi Girona Salgado, 1–3
Barcelona, Catalonia, Spain
{escudero,lluism,g.rigau}@lsi.upc.es

Abstract

This paper describes the architecture and results of the TALP system presented at the SENSEVAL-2 exercise for the English lexical–sample task. This system is based on the LazyBoosting algorithm for Word Sense Disambiguation (Escudero et al., 2000), and incorporates some improvements and adaptations to this task. The evaluation reported here includes an analysis of the contribution of each component to the overall system performance.

1 System Description

The TALP system has been developed on the basis of LazyBoosting (Escudero et al., 2000), a boosting–based approach for Word Sense Disambiguation. In order to better fit the SENSEVAL-2 domain, some improvements have been made on the basic system, including: features that take into account domain information, an specific treatment of multiwords, and a hierarchical decomposition of the multiclass classification problem, similar to that of (Yarowsky, 2000). All these issues will be briefly described in the following sections.

1.1 LazyBoosting

The purpose of boosting–based algorithms is to find a highly accurate classification rule by combining many *weak classifiers* (or weak hypotheses), each of which may be only moderately accurate. The weak hypotheses are learned sequentially, one at a time, and, conceptually, at each iteration the weak hypothesis is biased to classify the examples which were most difficult to classify by the preceding weak hypotheses. The learned weak hypotheses are linearly combined into a single rule called the *combined hypothesis*.

The particular algorithm used in our system to perform the classification of senses is the generalized AdaBoost.MH with confidence–rated predictions (Schapire and Singer, 1999). This algorithm is able to deal straightforwardly with multiclass multi–label problems, and has been previously applied, with significant success, to a number of NLP disambiguation tasks, including, among others: Part–of–speech tagging and PP–attachment (Abney et al., 1999), text categorization (Schapire and Singer, 2000), and shallow parsing (Carreras and Màrquez, 2001). The weak hypotheses used in this work are *decision stumps*, which can be seen as extremely simple decision trees with one internal node testing the value of a single binary feature (e.g. "the word *dark* appears in the context of the word to be disambiguated?") and two leaves that give the prediction of the senses based on the feature value.

The "Lazy" Boosting, is a simple modification of the AdaBoost.MH algorithm, which consists of reducing the feature space that is explored when learning each weak classifier. More specifically, a small proportion of attributes are randomly selected and the best weak rule is selected only among them. This modification significantly increases the efficiency of the learning process with no loss in accuracy (Escudero et al., 2000).

1.2 Feature Space

Three kinds of information have been used to describe the examples and to train the classifiers. These features refer to local and topical contexts, and domain labels.

More particularly, let "$\ldots w_{-3}\ w_{-2}\ w_{-1}\ w\ w_{+1}\ w_{+2}\ w_{+3}\ldots$" be the context of consecutive words around the word w to be disambiguated, and $p_{\pm i}$

($-3 \leq i \leq 3$) be the part–of–speech tag of word $w_{\pm i}$[1]. Feature patterns referring to local context are the following 13:

$p_{-3}, p_{-2}, p_{-1}, p_{+1}, p_{+2}, p_{+3}, w_{-2}, w_{-1}, w_{+1}, w_{+2}, (w_{-2}, w_{-1}), (w_{-1}, w_{+1})$, and (w_{+1}, w_{+2}),

where the last three correspond to collocations of two consecutive words.

The topical context is formed by $c_1, \ldots, c_m$, which stand for the unordered set of open class words appearing in a medium–size 21-word window centered around the target word.

The more innovative use of semantic domain information is detailed in the next section.

1.2.1 Domain Information

We have enriched the basic set of features by adding semantic information in the form of domain labels. These domain labels are computed during a preprocessing step using the 164 domain labels linked to the nominal part of WordNet 1.6 (Magnini and Cavaglia, 2000).

For each training example, a program gathers, from its context, all nouns and their synsets with the attached domain labels, and scores them according to a certain scoring function. The weights assigned by this function depend on the number of domain labels assigned to each noun and their relative frequencies in the whole WordNet. The result of this procedure is the set of domain labels that achieve a score higher than a certain experimentally set threshold, which are incorporated as regular features for describing the example.

1.3 Preprocessing and Hierarchical Decomposition

We began this exercise by selecting a representative sample, containing the most frequent words of the SENSEVAL-2 training data, and applying the LazyBoosting system straightforwardly on this sample. The results achieved after a 10–fold cross–validation procedure were very bad, mainly due to the fact that most of the words contain too many senses and too few examples per sense to induce reliable classifiers. With the aim of improving the performance of the learning algorithm, we have reduced the number of senses by performing a hierarchical decomposition of the multiclass problem, following the idea of (Yarowsky, 2000).

[1]In this work, the English versions of MACO+ morphological analyzer and RELAX part–of–speech tagger have been used for tagging (Carmona et al., 1998).

Two different simplifications have been carried out. Firstly, multiword training examples have been processed separately. During training, multiwords have been saved into a separate file. At test time, all examples found in this multiword file are automatically tagged as multiwords. As an example, the word *bar* appears in the training set with 22 labels. But only the 10 senses showed in the left table of figure 1 are single words. The remaining 12 are multiwords which are considered unambiguous (Yarowsky, 1993).

Full senses	
Senses	**Exs.**
bar%1:06:04::	127
bar%1:06:00::	29
bar%1:06:05::	28
bar%1:14:00::	17
bar%1:10:00::	12
bar%1:06:06::	11
bar%1:04:00::	5
bar%1:06:02::	4
bar%1:23:00::	3
bar%1:17:00::	1

1st level	
Senses	**Exs.**
bar%1:06	199
bar%1:14	17
bar%1:10	12

2nd level	
Senses	**Exs.**
04::	127
00::	29
05::	28
06::	11

Figure 1: Sense treatment for word 'bar'

Secondly, we have reduced the sense granularity, by hierarchically decomposing the learning process in two steps. In the first level, the learning algorithm is trained to classify between the labels corresponding to the WordNet semantic files, and, additionally the semantic–file labels with less than 10 training examples are automatically discarded. If less than two senses remain, no training is performed and, simply, the *Most-frequent-sense Classifier* is applied.

As an example, for the word *'bar'*, in this first step the system is trained to classify between the labels of the top–right table of figure 1. Note that senses *bar%1:04*, *bar%1:23* and *bar%1:17* have been dropped out because there are not enough training examples.

In the second level, one classifier is trained for each of the resulting semantic–file labels of the first step in order to distinguish between their particular senses. Note that the same simplifying rules of the previous level are also applied. For instance, the bottom–right table of figure 1 shows the labels for *bar%1:06*, where *02::* has been rejected.

When classifying a new test example, the classifiers of the two levels are applied sequentially. That

is, the semantic–file classifier is applied first. Then, depending on the semantic–file label output by this classifier, the appropriate 2nd level classifier is selected. The resulting label assigned to the test example is formed by the concatenation of the outputs of both previous levels.

In the official competition, labels 'U' and 'P' have been completely ignored. Thus, the examples labelled with these classes have not been considered during the training, and no test examples have been tagged with them.

Despite the simplifying assumptions and the loss of information, we have observed that all these changes together significantly improved the accuracy on the training set. However, the components of the system were not tested separately due to the lack of time. Next section includes some evaluation about this issue.

2 Evaluation

The official results achieved by the TALP system are presented in table 1. The evaluation setting corresponding to these results contains all the modifications explained in the previous sections, including the hierarchical approach to all words.

	Accuracy
fine–grained	59.4%
coarse–grained	67.1%

Table 1: Official results

After the SENSEVAL-2 event, we added a very simple Named–entity Recognizer to the part–of–speech tagger that was not finished at the time of the event, but the system continues ignoring the 'U' label. We also have evaluated which parts of the system contributed most to the improvement in performance.

Table 2 shows the accuracy results of the four combinations resulting from using (or not) domain–label features and hierarchical decomposition. These results have been calculated over the test set of SENSEVAL-2.

On the one hand, it becomes clear that enriching the feature set with domain labels systematically improves the results in all cases, and that this difference is specially noticeable in the case of nouns (over 3 points of improvement). On the other hand, the use of the hierarchies is unexpectedly useless in all cases. Although it is productive in some particular words (3 nouns, 12 verbs and 5 adjectives) the overall performance is significantly lower. A fact that can explain this situation is that the first–level classifiers do not succeed on classifying semantic–file labels with high precision (the average accuracy of first–level classifiers is only slightly over 71%) and that this important error is dramatically propagated to the second–level, not allowing the greedy sequential application of classifiers. A possible explanation of this fact is the way semantic classes are defined in WordNet. Consider for instance work#1 (activity) and work#2 (production), they seem quite close but a system trying to differentiate among semantic files needs to distinguish among these two senses. On the other extreme, such a classifier should collapse house#2 (legislature) with house#4 (family), which are quite different. Of course, joining both situations makes a pretty hard task.

nouns				
	without dom.		**with dom.**	
	fine	coarse	fine	coarse
not hier.	64.25	72.35	**67.90**	**75.60**
hier.	63.00	71.10	64.31	71.49
verbs				
	without dom.		**with dom.**	
	fine	coarse	fine	coarse
not hier.	51.61	61.63	**52.10**	**62.62**
hier.	50.28	60.80	51.11	61.96
adjectives				
	without dom.		**with dom.**	
	fine	coarse	fine	coarse
not hier.	66.17	66.17	**68.90**	**68.90**
hier.	65.35	65.35	68.21	68.21

Table 2: Fine/coarse–grained evaluation for different settings and part–of–speech

Regarding multiword preprocessing (not included in table 2), we have seen that is slightly useful in all cases. It improves the non–hierarchical scheme with domain information by almost 1 point in accuracy. By part–of–speech, the improvement is about 1 point for nouns, 0.1 for verbs and about 2 points for adjectives.

In conclusion, the best results obtained by our system on this test set correspond to the application of multiword preprocessing and domain–labels for all words, but no hierarchical decomposition at all, achieving a fine–grained accuracy of 61.51% and a coarse–grained accuracy of 69.00%. We know that it is not fair to consider these results for comparison, since the system is tuned over the test set. Our

aim is simply to fully inspect the TALP system to know which parts are useful for a real Word Sense Disambiguation system.

3 Work in progress

We think that the system presented in this paper still has a large room for improvement. Among all the research lines and developments that we are currently performing on the TALP system for WSD, we would like to mention the following:

- Tuning the preprocessing procedure with improved versions of the Named–entity Recognizer and Domain taggers.
- Studying in more detail the promising use of domain information in the feature set.
- Enriching the set of features with the most relevant features used by the SENSEVAL-2 systems, and using the Minipar[2] parser to obtain dependency and role information.
- Exploring more appropriate ways of making the hierarchical decomposition, not based on semantic files, and improve the sequential application of classifiers in order to reduce the cascade errors.
- Using unlabeled data to obtain larger sets of accurate training data, especially for those words/senses with few training examples.

4 Conclusions

This paper has presented the main characteristics and current performance of the TALP system within the framework of SENSEVAL-2 English lexical–sample task competition.

The system is mainly based on LazyBoosting (Escudero et al., 2000), which uses an improved version of the boosting algorithm AdaBoost.MH to perform the WSD classification problem.

We used a common set of features including local and topical context enriched with domain information. We obtained better performance separating multiword examples and also adding domain information.

Due to the small number of examples for training, we also tried to concentrate evidence reducing the fine-grained sense distinctions of WordNet. We perform a hierarchical procedure grouping those senses belonging to the same semantic file, preprocessing multiwords and ignoring 'U' label. After the competition, we have shown that the hierarchical decomposition fails to improve performance in this domain, while preprocessing of multiwords is quite useful. The improved system achieved a fine–grained accuracy of 61.51% and a coarse–grained accuracy of 69.00%.

[2]Available at http://www.cs.ualberta.ca/~ lindek .

5 Acknowledgements

We would like to thank Xavier Carreras, Lluís Padró, Victoria Arranz, and the anonymous referees for their helpful comments. This research has been partially funded by the European Commission (NAMIC project, IST-1999-12392) and the Spanish Research Department (HERMES project, TIC2000-0335-C03-02).

References

S. Abney, R. E. Schapire and Y. Singer. 1999. Boosting Applied to Tagging and PP–attachment. In *Proceedings of EMNLP–VLC'99.*

J. Carmona, S. Cervell, L. Màrquez, M. A. Martí, L. Padró, R. Placer, H. Rodríguez, M. Taulé and J. Turmo. 1998. An Environment for Morphosyntactic Processing of Unrestricted Spanish Text. In *Proceedings of LREC'98.*

X. Carreras and L. Màrquez. 2001. Boosting Trees for Clause Splitting. In *Proceedings of CoNLL'01.*

G. Escudero, L. Màrquez and G. Rigau. 2000. Boosting Applied to Word Sense Disambiguation. In *Proceedings of ECML'00.*

B. Magnini and G. Cavaglia. 2000. Integrating Subject Field Codes into WordNet. In *Proceedings of LREC'00.*

R. E. Schapire and Y. Singer. 1999. Improved Boosting Algorithms Using Confidence-rated Predictions. *Machine Learning*, 37(3):297–336.

R. E. Schapire and Y. Singer. 2000. BoosTexter: A Boosting-based System for Text Categorization. *Machine Learning*, 29(3/4):135–168.

D. Yarowsky. 1993. One Sense per Collocation. In *Proceedings of the DARPA Workshop on Human Language Technology.*

D. Yarowsky. 2000. Hierarchical Decision Lists for Word Sense Disambiguation. *Computer and the Humanities*, 34:179–186.

The UNED systems at SENSEVAL-2

David Fernández-Amorós, Julio Gonzalo, Felisa Verdejo
Depto. de Lenguajes y Sistemas Informáticos, UNED
{david,julio,felisa}@lsi.uned.es

Abstract

We have participated in the SENSEVAL-2 English tasks (all words and lexical sample) with an unsupervised system based on mutual information measured over a large corpus (277 million words) and some additional heuristics. A supervised extension of the system was also presented to the lexical sample task.

Our system scored first among unsupervised systems in both tasks: 56.9% recall in all words, 40.2% in lexical sample. This is slightly worse than the first sense heuristic for all words and 3.6% better for the lexical sample, a strong indication that unsupervised Word Sense Disambiguation remains being a strong challenge.

1 Introduction

We advocate researching unsupervised techniques for Word Sense Disambiguation (WSD). Supervised techniques offer better results in general but the setbacks, such as the problem of developing reliable training data, are very considerable. Also there's probably more to WSD than blind machine learning (a typical approach, although such systems produce interesting baselines).

Within the unsupervised paradigm, we are interested in performing in-depth measures of the disambiguation potential of different sources of information. We have previously investigated the informational value of semantic distance measures in (Fernández-Amorós et al.,). For SENSEVAL-2, we have turned to investigate pure coocurrence information as a source of disambiguation evidence. In essence, our system computes a matrix of mutual information for a fixed vocabulary and applies it to weight coocurrence counting between sense and context characteristic vectors.

In the next section we describe the process of constructing the relevance matrix. In section 3 we present the particular heuristics used for the competing systems. In section 4 we show the results by system and heuristic and some baselines for comparison. Finally in the last sections we draw some conclusions about the exercise.

2 The Relevance matrix

2.1 Corpus processing

Before building our systems we have developed a resource we've called the *relevance matrix*. The raw data used to build the matrix comes from the Project Gutenberg (PG) [1].

At the time of the creation of the matrix the PG consisted of more than 3000 books of diverse genres. We have adapted these books for our purpose : First, language identification was used to filter books written in English; Then we stripped off the disclaimers. The result is a collection of around 1.3Gb of plain text.

Finally we tokenize, lemmatize, strip punctuation and stop words and detect numbers and proper nouns.

2.2 Coocurrence matrix

We have built a vocabulary of the 20000 most frequent words (or labels, as we have changed all the proper nouns detected to the label PROPER_NOUN and all numbers detected to NUMBER) in the text and a symmetric coocurrence matrix between these words within a context of 61 words (we thought a broad context of radius 30 would be appropriate since we are trying to capture vague semantic relations).

2.3 Relevance matrix

In a second step, we have built another symmetric matrix, which we have called *relevance*

[1] `http://promo.net/pg`

matrix, using a mutual information measure between the words (or labels), so that for two words a and b, the entry for them would be $\frac{P(a \cap b)}{P(b)P(a)}$, where $P(a)$ is the probability of finding the word a in a random context of a given size. $P(a \cap b)$ is the probability of finding both a and b in a random context of the fixed size. We've introduced a threshold of 2 below which we set the entry to zero for practical purposes. We think that this is a valuable resource that could be of interest for many other applications other than WSD. Also, it can only grow in quality since at the time of making this report the data in the PG has almost doubled in size.

3 Cascade of heuristics

We have developed a very simple language in order to systematize the experiments. This language allows the construction of WSD systems composed of different heuristics that are applied in cascade so that each word to be disambiguated is presented to the first heuristic, and if it fails to disambiguate, then the word is passed on to the second heuristic and so on. We can have several such systems running in parallel for efficiency reasons (the matrix has high memory requirements). Next we show the heuristics we have considered to build the systems

- **Monosemous expressions.**

  Monosemous expressions are simply unambiguous words in the case of the all words English task. In the case of the lexical sample English task, however, the annotations include multiword expressions. We have implemented a multiword term detector that considers the multiword terms from WordNet's index.sense file and detects them in the test file using a multilevel backtracking algorithm that takes account of the inflected and base forms of the components of a particular multiword in order to maximize multiword detection. We tested this algorithm against the PG and found millions of these multiword terms.

  We restricted ourselves to the multiwords already present in the training file since there are, apparently, multiword expressions that where overlooked during manual tagging (for instance the WordNet expression 'the_good_old_days' is not hand-tagged as such in the test files)

- **Statistical filter**

  WordNet comes with a file, cntlist, literally 'file listing number of times each tagged sense occurs in a semantic concordance' so we use this to compute the relative probability of a sense given a word (approximate in the case of collections other than SemCor). Using this information, we eliminated the senses that had a probability under 10% and if only one sense remains we choose it. Otherwise we go on to the next heuristic. In other words, we didn't apply complex techniques with words which are highly skewed in meaning [2].

- **Relevance filter**

  This heuristic makes use of the relevance matrix. In order to assign a score to a sense, we count the coocurrences of words in the context of the word to be disambiguated with the words in the definition of the senses (the WordNet gloss tokenized, lemmatized and stripped out of stop words and punctuation signs) weighting each coocurrence by the entry in the relevance matrix for the word to be disambiguated and the word whose coocurrences are being counted, i.e., if s is a sense of the word α whose definition is S and C is the context in which α is to be disambiguated, then the score for s would be:

  $$\sum_{w \in C} R_{w\alpha} \text{freq}(w, C) \text{freq}(w, S) \text{idf}(w, \alpha)$$

  Where $\text{idf}(w, \alpha) = \log \frac{N}{d_w}$, with N being the number of senses for word α and d_w the number of sense glosses in which w appears. $\text{freq}(w, C)$ is the frequency of word w in the context C and $\text{freq}(w, S)$ is the frequency of w in the sense gloss S.

  The idea is to prime the occurrences of words that are relevant to the word being

[2] Some people may argue that this is a supervised approach. In our opinion, the cntlist information does not make a system supervised per se, because a) It is standard information provided as part of the dictionary and b) We don't use the examples to feed or train any procedure.

disambiguated and give low credit (possibly none) to the words that are incidentally used in the context.

Also, in the all words task (where POS tags from the TreeBank are provided) we have considered only the context words that have a POS tag compatible with that of the word being disambiguated. By compatible we mean nouns and nouns, nouns and verbs, nouns and adjectives, verbs and verbs, verbs and adverbs and vice versa. Roughly speaking, words that can have an intra-phrase relation.

We also filtered out senses with low values in the cntlist file, and in any case we only considered at most the first six senses of a word.

- **Enriching sense characteristic vectors**

  The relevance filter provided very good results in our experiments with SemCor and SENSEVAL-1 data as far as precision is concerned, but the problem is that there is little overlapping between the definitions of the senses and the contexts in terms of coocurrence (after removing stop words and computing idf) which means that the previous heuristic didn't disambiguate many words.

  To overcome this problem, we enrich the senses characteristic vectors adding for each word in the vector the words related to it via the relevance matrix weights. This corresponds to the algebraic notion of multiplying the matrix and the characteristic vector. In other words, if R is the relevance matrix and v our characteristic vector we would finally use $Rv + v$.

  This should increase the number of words disambiguated provided we eliminate the idf factor (which would be zero in most cases because now the sense characteristics vectors are not as sparse as before). When we also discard senses with low relative frequency in SemCor we call this heuristic *mixed filter*.

- **back off strategies**

  For those cases that couldn't be covered by other heuristics we employed the first sense heuristic. In the case of the supervised system for the English lexical sample task we thought of using the most frequent sense but didn't implement it due to lack of time.

4 Systems and Results

- **UNED–AW–U2**

  We won't delve into UNED-AW-U system as it is very similar to this one. This is an (arguably) unsupervised system for the English all words task. The heuristics we used and the results obtained for each of them are shown in Table 1.

Heuristic	Att.	Score	Prec	Rec
Monosemous exp	514	45500	88.5%	18.4%
Statistical filter	350	27200	77.7%	11.0%
Mixed filter	1256	50000	39.8%	20.2%
Enriched Senses	77	4300	55.8%	3.1%
First sense	249	13600	54.6%	5.5%
Total	2446	140600	57.5%	56.9%

Table 1: Unsupervised heuristics for English all words task

If the individual heuristics are used as standalone WSD systems we would obtain the results in Table 2.

System	Att.	Score	Prec	Recall
First sense	2405	146900	61.1%	59.4%
UNED–AW–U2	2446	140600	57.5%	56.9%
Mixed filter	2120	122600	57.8%	49.6%
Enriched senses	2122	108100	50.9%	43.7%
Random	2417	89191.2	36.9%	36.0%
Statistical filter	864	72700	84.1%	29.4%

Table 2: UNED–AW–U2 vs baselines

In the lexical sample task, we weren't able to multiply by the relevance matrix due to time constraints, so in order to increase the coverage for the relevance filter heuristic we expanded the definitions of the senses with those of the first 5 levels of hyponyms. Also, we selected the radius of the context to be considered depending on the POS of the word being disambiguated. For nouns and verbs we used 25 words radius neighbourhood and for adjectives 5 words at each side.

- **UNED–LS–U** This is essentially the same system as UNED–AW–U2, in this case applied to the lexical sample task. The results are displayed in Table 3.

Heuristic	Att.	Score	Prec	Recall
Relevance filt	3039	113617	37.3%	26.2%
First sense	1285	60000	46.7%	13.9%
Total	4324	173617	40.2%	40.2%

Table 3: Unsupervised heuristics for English lexical sample task

- **UNED–LS–T**

  This is a supervised variant of the previous systems. We have added the training examples to the definitions of the senses giving the same weight to the definition and to all the examples as a whole (i.e. definitions are considered more interesting than examples)

Heuristic	Att.	Score	Prec	Recall
Relevance filt	4116	206150	50.1%	47.6%
First sense	208	9300	44.7%	2.1%
Total	4324	215450	49.8%	49.8%

Table 4: Supervised heuristics for English lexical sample task

5 Discussion and conclusions

We've put a lot of effort into making the relevance matrix but its performance in the WSD task is striking. The matrix is interesting and its application in the relevance filter heuristic is slightly better than simple coocurrence counting, which proves that it doesn't discard relevant words. The problem seems to lie in the fact that irrelevant words (with respect to the word to be disambiguated) rarely occur both in the context of the word and in the definition of the senses (if they appeared in the definition they wouldn't be so irrelevant) so the direct impact of the information in the matrix is very weak. Likewise, relevant (via the matrix) words with respect to the word to be disambiguated occur often both in the context and in the definitions so the final result is very similar to simple coocurrence counting.

This problem only showed up in the lexical sample task systems. In the all words systems we were to enrich the sense definitions to make a more advantageous use of the matrix.

We were very confident that the relevance filter would yield good results as we have already evaluated it against the SENSEVAL-1 and SemCor data. We felt however that we could improve the coverage of the heuristic enriching the definitions multiplying by the matrix. A similar approach was used by Yarowsky (Yarowsky, 1992) and Schütze (Schütze and Pedersen, 1995) and it worked for them. This wasn't the case for us; still, we think the resource is well worth researching other ways of using it.

As for the overall scores, the unsupervised lexical sample obtained the highest recall of the unsupervised systems, which proves that carefully implementing simple techniques still pays off. In the all words task the UNED-WS-U2 had also the highest recall among the unsupervised systems (as characterized in the SENSEVAL-2 web descriptions), and the fourth overall. We'll train it with the examples in Semcor 1.6 and see how much we can gain.

6 Conclusions

Our system scored first among unsupervised systems in both tasks: 56.9% recall in all words, 40.2% in lexical sample. This is slightly worse than the first sense heuristic for all words and 3.6% better for the lexical sample, a strong indication that unsupervised Word Sense Disambiguation remains being a strong challenge.

References

D. Fernández-Amorós, J. Gonzalo, and F. Verdejo. The role of conceptual relations in word sense disambiguation. In *Applications of Natural Language to Information Systems (NLDB)'01, Madrid.*

H. Schütze and J. Pedersen. 1995. Information retrieval based on word senses. In *Fourth Annual Symposium on Document Analysis and Information Retrieval, Las Vegas NV*, pages 161–175.

D. Yarowsky. 1992. Using statistical models of roget's categories trained on large corpora. In *COLING'92, Nantes*, pages 454–460.

Semantic Tagging Using WordNet Examples

Sherwood Haynes
Illinois Institute of Technology
Department of Computer Science
Chicago, Illinois, 60616 USA
skhii@mindspring.com

Abstract

This paper describes IIT1, IIT2, and IIT3, three versions of a semantic tagging system basing its sense discriminations on WordNet examples. The system uses WordNet relations aggressively, both in identifying examples of words with similar lexical constraints and matching those examples to the context.

1 Introduction

The ability of natural language understanding systems to determine the meaning of words in context has long been suggested as a necessary precursor to a deep understanding of the context (Ide and Véronis, 1998; Wilks, 1988). Competitions such as SENSEVAL (Kilgarriff and Palmer, 2000) and SENSEVAL-2 (SENSEVAL-2, 2001) model the determination of word meaning as a choice of one or more items from a fixed sense inventory, comparing a gold standard based on human judgment to the performance of computational word sense disambiguation systems.

Statistically based systems that train on tagged data have regularly performed best on these tasks (Kilgarriff and Rosenzweig, 2000). The difficulty with these supervised systems is their insatiable need for reliable annotated data, frequently called the "data acquisition bottleneck."

The systems described here avoid the data acquisition bottleneck by using only a sense repository, or more specifically the examples and relationships contained in the sense repository.

WordNet version 1.7 (Miller 1990; Fellbaum 1998; WordNet, 2001) was chosen as the sense repository for the English Lexical Sample task (where systems disambiguate a single word or collocation in context) and the English All Word task (where systems disambiguate all content words) of the SENSEVAL-2 competition. WordNet defines a word sense (or synset) as a collection of words that can express the sense, a definition of the sense (called a gloss), zero or more examples of the use of the word sense, and a set of tuples that define relations between synsets or synset words.

2 General Approach

This paper describes three systems that were entered in SENSEVAL-2 competition, IIT1, IIT2, and IIT3. IIT1 and IIT2 were entered in both the English All Word task and the English Lexical Sample task. IIT3 was entered in the English All Word task only. All three systems use the same unsupervised approach to determine the sense of a target word:

1. for each syntactically plausible sense, find the set of WordNet examples that appear in that synset or a related synset.
2. for each example, compare the example to the context, scoring the quality of the match.
3. choose the sense whose synset is responsible for the inclusion of the highest scoring example.

Hereafter, **target words** identify the words to be disambiguated (so identified by the SENSEVAL-2 task). The **context** identifies the text surrounding and including a target word.

2.1 Collecting Examples of a Sense

The systems first collect a set of example sentences and phrases from WordNet for each synset matching a target word (or its canonical or collocational form). The set includes examples from the synset itself as well as those of related synsets. Table 1 lists the relations available in WordNet 1.7. The application of direct relations includes only the examples of the related synset (or synsets of related words). The transitive closure of relations additionally

WordNet Relation	Relation Type	Relation Operands	Application of Relation
Antonym		Word	Direct
Hypernym	Parent	Synset	Transitive Closure
Hyponym	Child	Synset	Direct
Entailment		Synset	Transitive Closure
Similarity	Set	Word	Transitive Closure
Member	Child	Synset	Direct
Stuff	Child	Synset	Direct
Part	Child	Synset	Direct
Has Member	Parent	Synset	Transitive Closure
Has Stuff	Parent	Synset	Transitive Closure
Has Part	Parent	Synset	Transitive Closure
Holonym	Parent	Synset	Transitive Closure
Meronym	Child	Synset	Direct
PPL		Word	Transitive Closure
See Also		Word	Direct
Pertains		Word	Transitive Closure
Attribute		Synset	Transitive Closure
Verb Group	Set	Synset	Not Used

Table 1
Use of WordNet Relations

includes examples from repeated application of the relation. That is, for the hypernym relation, examples from all ancestor synsets are included.

Table 2 lists the examples identified for the synset for *faithful - steadfast in affection or allegiance.* WordNet 1.7 displays the synset as:

faithful (vs. unfaithful)
=> firm, loyal, truehearted, fast(postnominal)
=> true
Also See-> constant#3; true#1; trustworthy#1, trusty#1

This *faithful* synset contributes 3 examples, the *see also* relation contributes examples for *constant*, *true,* and *trustworthy*, the *similarity* relation contributes the examples from the *firm* synset and the *antonym* relation contributes the *unfaithful* example.

2.2 Comparing Examples to the Context

Each example is compared to the context. Consider the first example in Table 2, *a man constant in adherence to his ideals*. Since each example contains a word being defined, the systems consider that this word matches the target word, so *constant* is assumed to match *faithful.* Call this word the **example anchor.**

The remaining words of the example are compared to the words surrounding the target word. The comparison begins with the word to

Synset Words	Example
constant	a man constant in adherence to his ideals
	a constant lover
	constant as the northern star
faithful	years of faithful service
	faithful employees
	we do not doubt that England has a faithful patriot in the Lord Chancellor
firm, loyal, truehearted, fast	"the true-hearted soldier...of Tippecanoe" - Campaign song for William Henry Harrison;
	a firm ally
	loyal supporters
	fast friends
true	true believers bonded together against all who disagreed with them
	the story is true
	"it is undesirable to believe a proposition when there is no ground whatever for supposing it true" - B. Russell;
	the true meaning of the statement
trustworthy	a trustworthy report
	an experienced and trustworthy traveling companion
unfaithful	an unfaithful lover

Table 2
Examples Relate to Synset *faithful – steadfast in affection or allegiance*

the left of the example anchor followed by the word immediately to the right of the anchor, the second word to the left of the anchor, the second word to the right of the anchor, and so on. So the order of comparison of the example words is *man, in, a, adherence, to, his, ideals.*

Each example word is compared to the unmatched context words in a similar sequence. So, for example, the example word *man* would first be compared to the word immediately to the left of the context word followed by the word to its left, and so on, until a match is found.

Word matches also use the WordNet relations as described in Table 1. Under parent relations, two words match if they have a common ancestor. Other transitive closure relations generate a match if either word appears in the other's transitive closure. The words also match if there is a direct relation between the words.

2.3 Scoring the Match

Once the words of an example have been matched to the context, the result is scored. The score for all systems is computed as:

Characteristic	Description
Distance	Magnitude of the difference in the word position of the matching example and context words relative to the position of the example and context anchors
Direction Change	1 if the example words adjacent to a word match context words both occurring before or after its matching context word, 0 otherwise.
Lexical Proximity	0 for exact matches; 1 for matches based on non-parent relation matches; sum of the distances to the closest common ancestor for matches under parent relations
Maximum and Minimum Lexical Generalization	0,0 for exact matches; 1,0 for matches based on non-parent relation matches; maximum and minimum distance to the closest common ancestor for matches under parent relations
Alignment Skew	Ratio of the matching phrase length to the example length.
Match Failure	1 for example words with no matching context word, 0 otherwise

Table 3
Scoring Penalty Characteristics

$$score = \frac{1}{1 + \sum_i \sum_j s(w_i, c_j, d_i)}$$

The scoring function s generates a non-negative value for each example word w_i, penalty characteristic c_j (Table 3), distance d_i of w_i from the example anchor. In IIT1, d_i is not considered, so a penalty calculation is independent of the word position in the example. In IIT2, d_i reduces penalties for w_i further away from the example anchor.

If an example anchor alignment with the context word is the only open-class match for an example, the example receives a zero score.

Haynes (2001) describes these calculations in more detail.

A sense of a target word receives the maximum score of the examples related to that sense. The systems suggest the sense(s) with the highest score, with multiple senses in the response in the event of ties. (If a tie occurs because the same example was included for two senses, the other senses are eliminated, the common example is dropped from the example set of the remaining senses, and the sense scores are recomputed.) If no sense receives a score greater than zero, the first sense is chosen.

IIT1 and IIT2 match a context word independent of other sense assignment decisions. The IIT3 system (English All Word task only) uses the IIT1 scoring algorithm for target words, but limits the senses of preceding context words to the sense tags already assigned.

System	Course Grained Precision/Recall	Fine Grained Precision/Recall
IIT1 Lexical Sample	34.1% / 33.6%	24.3% / 23.9%
IIT2 Lexical Sample	34.6% / 34.1%	24.7% / 24.4%
Baseline Lesk	33.1% / 33.1%	22.6% / 22.6%
Best Non-Corpus	36.7% / 36.7%	29.3% / 29.3%

Table 4
SENSEVAL-2 English Lexical Sample Results

System	Course Grained Precision/Recall	Fine Grained Precision/Recall
IIT1 All Word	29.4% / 29.1% *	28.7% / 28.3% *
IIT2 All Word	33.5% / 33.2% *	32.8% / 32.5% *
IIT3 All Word	30.1% / 29.7% *	29.4% / 29.1% *
Best Non-Corpus	46.0% / 46.0%	45.1% / 45.1%

Table 5
SENSEVAL-2 English All Word Results

3 Results

Table 4 and Table 5 show the results for IIT1, IIT2 and IIT3 as well as that of the Lesk Baseline (English Lexical Sample task) and the best non-corpus based system, the CRL DIMAP system. The SENSEVAL-2 (2001) website presents the complete competition results as well as the CRL DIMAP and baseline system descriptions.

The IIT1 and IIT2 performed better than the comparable baseline system but not as well as the best system in its class. The IIT3 approach improves on the performance of IIT1 by using its prior annotations in tagging subsequent words.

Due to time constraints, the English All Word submissions only processed the first 12% of the corpus. The recall values marked * consider only those instances attempted.

4 Discussion

Many of the examples in WordNet were the result of lexicographers expanding synset information to clarify sense distinctions for the annotators of the Semcor corpus (Fellbaum, 1998). This makes a compelling argument for the use of these WordNet examples to assist in a computational disambiguating process.

The examples for rare word senses could be used to provide corpus-based statistical methods with additional evidence. Such an approach should help address the knowledge acquisition bottleneck.

The implementation and results presented here do not seem to justify this optimism. There are several reasons, though, why the method should not be dismissed without further investigation:

- The example sets were empty for a number of the candidate word senses. When this occurred, the system constructed a pseudo example by appending the WordNet gloss to the target word. This was sufficient for most collocation senses and some non-collocation senses such as *call* as in *calling a square dance* (where the gloss includes *square* and *dance*, one of which is highly likely to occur in any use of the sense). Others such as *day* as in *sidereal day* or *turn off* (gloss *cause to feel intense dislike or distaste*) competed at a disadvantage.
- The pattern matching and scoring methods were never tuned against any corpus data. This allowed the algorithm to have few competitors in the class of untrained systems, but scoring methods relied on intuition-founded heuristics. Such tuning should improve precision and recall.
- The approach was developed to be used in tandem with statistical approaches. Further research is required before its additive value can be fully assessed. IIT3 would have done better to be based on IIT2 and an approach maximizing the scores for a sentence should do even better.
- The best-matching example was chosen regardless of how bad a match was involved. The system also defaulted to the first sense encountered when all examples had a zero score. Using threshold score values may well provide substantial precision improvements (at the expense of recall).
- Semantic annotation of the WordNet examples should improve the results.

In addition, the following programming errors affected the precision and recall results:

- The generated answers for many adjective senses (those with similarity relations) were incorrectly formatted and were therefore always scored as incorrect. For example, in the IIT1 entry for the English Lexical Sample, 7.1% of all annotations were incorrectly formatted. Scoring only the answers that were correctly formatted raises the course-grained precision for IIT1 to 36.7% and fine-grained precision to 26.1%, competitive with the course-grained performance of the best non-corpus system.
- No annotations were generated for target words preceded by the word *to*. This results in recall ≠ precision as seen in Table 4 and Table 5.
- In a few rare cases, the system identified the incorrect example word as the example anchor. One such occurrence was the synset *art, fine art* and the example *a fine collection of art.* The system considered it an example of the *fine art* collocation and chose *fine* as the anchor.

5 Conclusion

The approach presented here does not appear to be sufficient for a stand-alone word sense disambiguation solution. Whether this method can be combined with other methods to improve their results requires further investigation.

References

Christiane Fellbaum, ed. (1998) *WordNet: An Electronic Lexical Database.* The MIT Press, Cambridge, Massachusetts

Sherwood Haynes (2001) *http://skhii.home.mindspring.com*

Nancy Ide and Jean Véronis (1998) *Introduction to the Special Issue on Word Sense Disambiguation: The State of the Art.* Computational Linguistics, 24/1, pp. 1 – 40.

Adam Kilgarriff and Martha Palmer (2000) *Introduction to the Special Issue on SENSEVAL.* Computers and the Humanities 34/1, pp. 1 – 13.

Adam Kilgarriff and Joseph Rosenzweig (2000) *Framework and Results for English SENSEVAL.* Computers and the Humanities 34/1, pp. 15 – 48.

George Miller, ed. (1990) *WordNet: An On-line Lexical Database.* International Journal of Lexicography, 3/4

SENSEVAL-2 (2001) *http://www.sle.sharp.co.uk/sensev al2*

Yorick Wilks (1988) *Forward.* In "Lexical Ambiguity Resolution: Perspectives from Psycholinguistics, Neuropsychology, and Artificial Intelligence," Small, S., Cottrell, G., & Tanenhaus, M. ed., Morgan Kaufmann Publishers, Inc., San Mateo, California, pp. iii – ix.

WordNet (2001) version 1.7, available at *http://www.cogsci.princeton.edu/~wn/*

Classifier optimization and combination in the English all words task.

Véronique Hoste and **Anne Kool** and **Walter Daelemans**
CNTS - Language Technology Group
University of Antwerp
Universiteitsplein 1, 2610 Wilrijk
hoste@uia.ua.ac.be, kool@uia.ua.ac.be, daelem@uia.ua.ac.be

Proceedings of SENSEVAL-2, Preiss & Yarowsky, eds., p. 83-86, 2001

Abstract

We report on the use of machine learning techniques for word sense disambiguation in the English all words task of SENSEVAL2. The task was to automatically assign the appropriate sense to a possibly ambiguous word form given its context. A "word expert" approach was adopted, leading to a set of classifiers, each specialized in one single word form-POS combination. Experts consist of multiple classifiers trained on Semcor using two types of learning techniques, viz. memory-based learning and rule-induction. Through optimization by cross-validation of the individual classifiers and the voting scheme for combining them, the best possible word expert was determined. Results show that especially memory-based learning in a word-expert approach is a feasible method for unrestricted word-sense disambiguation, even with limited training data.

1 Introduction

We report on the use of machine learning, especially memory-based learning and classifier combination, for word sense disambiguation (WSD) in the English all words task of SENSEVAL2. WSD can be described as the problem of assigning the appropriate sense to a given word in a given context. Machine learning techniques show state-of-the-art accuracy on WSD, e.g. memory-based learning (Ng and Lee, 1996; Veenstra et al., 2000), decision lists (Yarowsky, 2000), and combination methods (Escudero et al., 2000).

Results of the first SENSEVAL exercise for English (Killgarriff and Rosenzweig, 2000), in which only a restricted set of words had to be disambiguated, showed that supervised learning systems outperform unsupervised ones, even when little corpus training material was available. In our submission to SENSEVAL2, we investigated whether the supervised learning approach can be scaled to the all-words task. As a back-off for word-tag pairs for which no or not enough training data was available, we used the most frequent sense in the WordNet1.7 sense lexicon (Fellbaum, 1998) as default classifier in the disambiguation process. Sense disambiguation was mainly performed by a memory-based learning classifier. Also the use of rule induction was explored. Furthermore, the outputs of these different classifiers were combined in order to study the usefulness of different voting strategies. Results show that all classifiers outperform the WordNet baseline and that memory-based learning compares favorably to rule induction and different voting strategies.

In the remainder of this paper, we first outline the sense-disambiguation architecture used in the experiments, and discuss the word expert approach and the optimization procedure. Then we report on the generalization accuracy achieved for the SENSEVAL2 test data.

2 Experimental Setup

2.1 Preprocessing

In the experiments, the Semcor corpus included in WordNet1.6 was used as training corpus. In the corpus, every word is linked to its appropriate sense in the lexicon. Texts that were used to create the semantic concordances were extracted from the Brown Corpus and then linked to senses in the WordNet lexicon. The training corpus consists of 409,990 wordforms, of which 190,481 are sense-tagged. For each word form in the corpus, a lemma and a part of speech is given.

The test data in the English all words task consist of three articles on different topics, with at total of 2,473 words to be sense-tagged. For

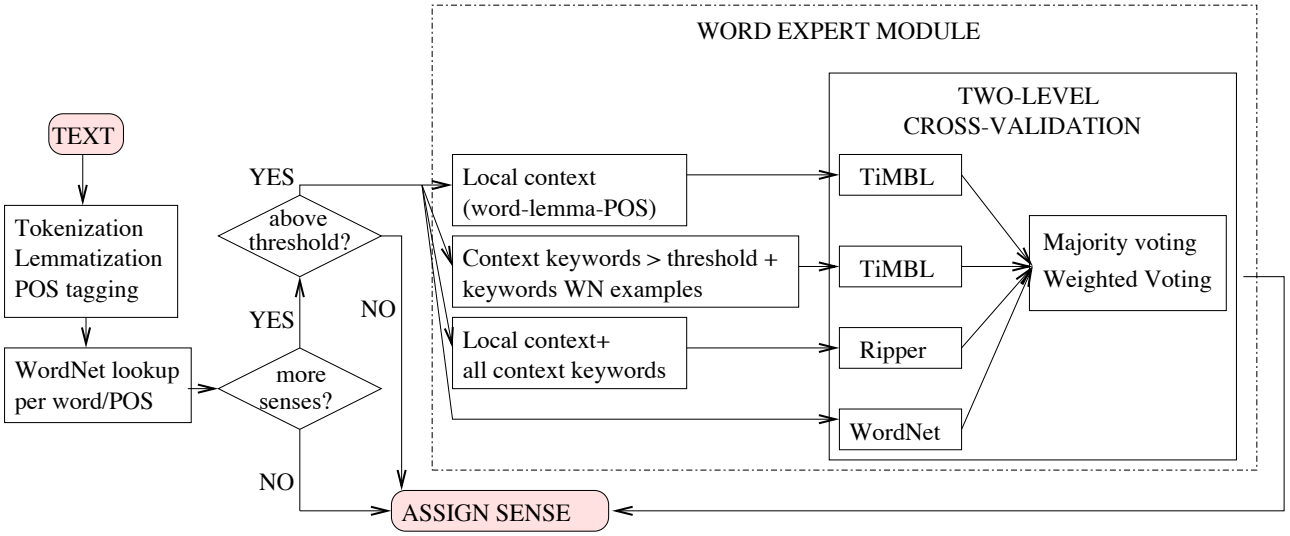

Figure 1: Disambiguation process.

both the training and the test corpus, only the word forms were used and tokenization, lemmatization and POS-tagging were done with our own software. For the part of speech tagging, the memory-based tagger MBT (Daelemans et al., 1996), trained on the Wall Street Journal corpus[1], was used. On the basis of word and POS information, lemmatization was done[2].

2.2 Word experts

After the preprocessing stage, WordNet1.7 was used to guide the sense disambiguation process. For every combination of a word form and a POS, WordNet was consulted to determine whether this combination had one or more possible senses. In case of only one possible sense (about 20% of the test words), the appropriate WordNet sense was assigned. In case of more possible senses, a threshold of 11 occurrences in the Semcor training data was determined. For all words below this threshold, the most frequent sense according to WordNet was assigned as sense-tag. For the other words, which represent more than 60% of the word forms to be sense-tagged, word experts were built for each word form-POS combination, leading to 568 word experts for the SENSEVAL2 test data.

These word experts consist of different trained subcomponents (see Figure 1) which make use of different knowledge.

The first subcomponent is trained using TiMBL, a package containing several memory-based learning algorithms and metrics (Daelemans et al., 2000). It takes as input a vector representing the local context of the focus word in a window of three word forms to the left and three to the right. For the focus word, also the lemma and POS are provided. For the context word forms, POS information is given. E.g., the following is a training instance:
`many JJ times NNS , , yet yet RB on IN each JJ occasion NN yet%4:02:02::.`
During training, those instances are stored in memory and during sense-tagging, the instance most similar to that of the ambiguous word and its context is selected and the associated class is returned as sense-tag.

A second subcomponent of each word expert trained with TiMBL is trained with information about possible disambiguating content keywords in a context of three sentences. The method used to extract these keywords for each sense is based on the work of (Ng and Lee, 1996). They determine the probability of a sense s of a focus word f given keyword k by dividing $N_{s,kloc}$ (the number of occurrences of a possible local context keyword k with a particular focus word-POS combination w with a particular sense s) by N_{kloc} (the number of occurrences of a possible local context keyword $kloc$ with a particular focus word-POS combi-

[1] ACL Data Collection Initiative CD-Rom 1, September 1991

[2] With a memory-based lemmatizer trained by Antal van den Bosch, see `http://ilk.kub.nl/`

nation w ignoring its sense). In addition, we also took into account the frequency of a possible keyword in the complete training corpus N_{kcorp}.

$p(s|k) = \frac{N_{s,kloc}}{N_{kloc}} \times (\frac{1}{N_{kcorp}})$

A word is a keyword for a given sense if (i) the word occurs more than M_1 times in that sense s, where M_1 is a predefined minimum number of times and if (ii) $p(s|k) \geq M_2$ for that sense s, where M_2 is some predefined minimum probability. Due to time restrictions M_1 was not optimized by cross-validation, but arbitrarily set to 3 and M_2 to 0.001.

In addition to the keyword information extracted from the local context of the focus word, possible disambiguating content words were also extracted from the examples that accompany the different sense definitions for a given focus word in WordNet. For each combination of a word form, POS and sense, all content words were extracted and added to the input vector of the memory-based learner. Both the contextual keywords and the example keywords were represented as binary features, with a value of 1 when the keyword was present in the example and 0 if not[3].

The third subcomponent of each word expert was trained with Ripper (Cohen, 1995), a rule learning algorithm, allowing both single-valued and set-valued attributes. In our disambiguation task, the ripper input vector contained local context feature values (as the first TiMBL), and a set-valued feature with all content words in a context of three sentences.

3 Optimization and Voting

In order to improve the predictions of the different single learning algorithms, algorithm parameter optimization was performed where possible. Furthermore, the possible gain in accuracy of different voting strategies was explored.

3.1 Optimization

For the first TiMBL memory-based learner, backward sequential selection (BSS) (Aha and Bankert, 1994) was performed for each word form-POS combination. BSS starts from the complete feature set and generates in each iteration new subsets by discarding a feature. The feature string with the best performance is retained. Furthermore, the use of different feature weighting possibilities was explored, viz. gain ratio weighting, information gain weighting, chi-squared weighting and shared variance weighting. For each feature weighting possibility, the k value, representing the number of nearest neighbours used for extrapolation, was varied between 1 and 19. Leave-one-out was used as testing method: testing was done on each instance of the training file, while the remainder of the training file functioned as training material.

Due to the size of the feature vectors for the second memory-based learner, which takes content words from the surrounding sentences and from the example sentences in the WordNet definitions as input, no feature selection was performed. For the same reasons, 10-fold cross-validation was used as testing method: the training data was split into 10 different parts and in each iteration, one part served as test set, while the remainder was used to train the classifier. The k value was varied (1-19), different weighting techniques (gain ratio weighting, chi-squared weighting and log likelihood weighting) and different distance metrics (number of mismatches, number of matches, number of matches minus number of mismatches) were explored.

For Ripper, the default parameter settings were used, due to time constraints and the slowness of the cross-validation process. 10-fold-cross-validation was used as testing method.

3.2 Voting

On the output of these three (optimized) classifiers and the default WordNet1.7. most frequent sense, both majority voting and weighted voting was performed. In case of majority voting, each sense-tagger is given one vote and the tag with most votes is selected. In weighted voting, more weight is given to the taggers with a higher overall accuracy. In case of ties when voting over the output of 4 classifiers, the first decision (TiMBL) was taken as output class. Voting was also performed on the output of the three learning classifiers without taking into ac-

[3]Since no length limitations were taken into account when building these vectors, they could grow very large. Therefore, a version of TiMBL was used that is optimized for sparse binary features, and allows a positional representation of the active keywords rather than a binary one, written by Jakub Zavrel.

Classifier	no. WE
Default (WordNet1.7)	16
TiMBL (context)	**155**
TiMBL (keywords)	**185**
Ripper	16
Majority Voting	33
Weighted Voting	58
Majority Voting (no WordNet)	53
Weighted Voting (no WordNet)	52
	568

Table 1: Best performing word experts on the Semcor train set

count the WordNet class. Table 1 shows the best performing classifiers per word form-POS combination of the Semcor train set: both optimized memory-based learners outperform the other classifiers.

4 Results

Table 2 shows the accuracy of our disambiguation system on the English all words test set. Since all 2,473 word forms were covered, no distinction is made between precision and recall. An accuracy of 63.61% and 64.54% were obtained according to the fine-grained and coarse-grained SENSEVAL2 scoring, respectively. Just as in the first SENSEVAL task for English (Killgarriff and Rosenzweig, 2000), top performance was for the nouns. All 86 "unknown" word forms, for which the test set annotators decided that no WordNet1.7 sense-tag was applicable, were obviously incorrectly classified.

	key	fine %	coarse %
noun (%1)	1,067	74.51	75.45
verb (%2)	554	47.83	49.64
adj. (%3-%5)	465	62.58	63.44
adv. (%2)	301	73.42	73.42
unkn.	86	0.00	0.00
total	**2,473**	**63.61**	**64.54**

Table 2: Results on the SENSEVAL2 test data.

5 Conclusion

This paper reported on the architecture and the results of the CNTS-Antwerp automatic disambiguation system in the context of the SENSEVAL2 English all words task. Disambiguation per word form-POS pair is performed through the application of word experts trained on local context information and cross-validated on the limited available training data. Among these word experts, optimized memory-based learning proves to be more accurate than default Ripper rule-induction and various voting strategies.

Acknowledgements

We like to thank Antal van den Bosch for taking care of the lemmatization and Erik Tjong Kim Sang for programming support.

References

D.W. Aha and R.L. Bankert. 1994. Feature selection for case-based classification of cloud types: An empirical comparison. In *Proceedings of the 1994 AAAI Workshop on Case-Based Reasoning*, pages 106–112. AAAI Press.

W.W. Cohen. 1995. Fast effective rule induction. In *Proc. 12th International Conference on Machine Learning*, pages 115–123. Morgan Kaufmann.

W. Daelemans, J. Zavrel, P. Berck, and S. Gillis. 1996. Mbt: A memory-based part of speech tagger-generator. In E. Ejerhed and I. Dagan, editors, *Fourth Workshop on Very Large Corpora*, pages 14–27.

W. Daelemans, J. Zavrel, K. van der Sloot, and A. van den Bosch. 2000. Timbl: Tilburg memory based learner, version 3.0, reference guide.

G. Escudero, L. Marquez, and G. Rigau. 2000. Boosting applied to word sense disambiguation. In *European Conference on Machine Learning*, pages 129–141.

C. Fellbaum. 1998. *WordNet : An Electronic Lexical Database*. MIT Press.

A. Killgarriff and J. Rosenzweig. 2000. English senseval: Report and results. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation*, pages 1239 1243.

H.T. Ng and H.B. Lee. 1996. Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In Arivind Joshi and Martha Palmer, editors, *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, pages 40 47, San Francisco. Morgan Kaufmann Publishers.

J. Veenstra, A. Van den Bosch, S. Buchholz, W. Daelemans, and J. Zavrel. 2000. Memory-based word sense disambiguation. *Computers and the Humanities*, 34(1/2):171–177.

D. Yarowsky. 2000. Hierarchical decision lists for word sense disambiguation. *Computers and the Humanities*, 34(1/2):179–186.

Combining Heterogeneous Classifiers for Word-Sense Disambiguation

H. Tolga Ilhan, Sepandar D. Kamvar, Dan Klein, Christopher D. Manning and **Kristina Toutanova**
Computer Science Department
Stanford University
Stanford, CA 94305-9040, USA

Abstract

The Stanford-CS224N system is an ensemble of simple classifiers. The first-tier systems are heterogeneous, consisting primarily of naive-Bayes variants, but also including vector space, memory-based, and other classifier types. These simple classifiers are combined by a second-tier classifier, which variously uses majority voting, weighted voting, or a maximum entropy model. Results from SENSEVAL-2 lexical sample tasks indicate that, while the individual classifiers perform at a level comparable to middle-scoring team's systems, the combination achieves high performance. In this paper, we discuss both our system and lessons learned from its behavior.

1 Introduction

The problem of supervised word sense disambiguation (WSD) has been approached using many different classification algorithms, including naive Bayes, decision trees, decision lists, and memory-based learners. While it is unquestionable that certain algorithms are better suited to the WSD problem than others (for a comparison, see Mooney (1996)), it seems to be the case that, given similar features as input, various algorithms do not behave dramatically differently. This was seen in the SENSEVAL-2 results where a large fraction of the systems had scores clustered in a fairly narrow region.

We began building our system with 23 supervised WSD systems, each submitted by a student taking the natural language processing course (CS224N) at Stanford University. Students were free to implement whatever WSD method they chose. While most implemented variants of naive Bayes, some implemented a range of other methods, including n-gram models, vector space models, and even memory-based learners. Although none of these systems alone would have produced more than middle-level performance on the SENSEVAL-2 task, we decided to investigate how they would behave in combination.

This paper is based on work supported in part by the National Science Foundation under Grants IIS-0085896 and IIS-9982226, by an NSF Graduate Fellowship, and by the Research Collaboration between NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation and CSLI, Stanford University.

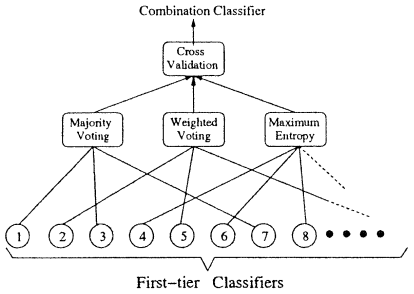

Figure 1: Organization of the system.

In section 2, we discuss the first-tier classifiers in greater depth and describe our methods of combination. Section 3 discusses performance, analyzing what benefit was found from combination, and when. We also discuss aspects of the component systems which substantially influenced overall performance.

2 The System

Figure 1 shows the high-level organization of our system. First, each of the 23 classifiers is run with 5-fold cross-validation on the training data. Classifiers are ranked, for each word, based on their held-out accuracy. In any given run of the system, for some k, the top k classifiers are kept, while lower-ranking classifiers are discarded. These remaining classifiers are combined by one of three methods.

- *Majority voting:* The sense output by the most classifiers is chosen. Ties are broken in favor of the highest-ranked classifier.

- *Weighted voting:* Each classifier is assigned a voting weight (see below) and adds that weight to the sense it outputs. The sense receiving the greatest total weight is chosen.
- *Maximum entropy:* A maximum entropy classifier is trained (see below) and run on the (classifier, vote) outputs from the first tier.

We consider k in the range $\{5, 7, 9, 11, 13, 15\}$, and so, once the ranking of the first-tier classifiers is set, there are 18 possible second-tier classifiers.

We train and test each (k, method) pair on the training data, again with 5-fold cross-validation. The classifier type and k-value which perform best on the held-out data are chosen. Once the (k, method) pair is chosen, all first-tier classifiers, as well as the parameters for the second-tier combinator, are retrained on the entire training corpus. Each target word is considered an entirely separate task, and different first- and second-tier choices can be, and are, made for each word. Table 1 shows what second-tier choices were made for each word.

2.1 Combination Methods

Our second-tier classifier takes training instances of the form $\bar{s} = (s, s_1, \ldots, s_k)$ where s is the correct sense and each s_i is the sense chosen by classifier i. We initially planned to combine students' classifiers using only a maximum entropy model. Such a model has a set of features $f_x(\bar{s})$ where each feature f_x is true over a subset of vectors $\bar{s}$. A conditional maximum entropy model with such features assigns, for any given choices s_i, a distribution over the possible senses s. This distribution is of the form:

$$P(s|s_1, \ldots, s_k) = \frac{\exp \sum_x \lambda_x f_x(s, s_1, \ldots, s_k)}{\sum_t \exp \sum_x \lambda_x f_x(t, s_1, \ldots, s_k)}$$

The intent was to design the features to recognize and exploit "sense expertise" in the individual classifiers. For example, one classifier might be trustworthy when reporting a certain sense but less so for other senses. However, there was nowhere near enough data to accurately estimate parameters for such models.[1]

In fact, we noticed that, for certain words, simple majority voting performed better than the maximum entropy model. It also turned out that the most complex features we could get value from were features of the form:

$$f_i(s, s_1, \ldots, s_k) = 1 \iff s = s_i$$

However, with only these features, the maximum entropy approach reduces to a weighted vote; the s which maximizes the posterior probability $P(s|s_1, \ldots, s_k)$ also maximizes the vote:

$$v(s) = \sum_i \lambda_i \delta(s_i = s)$$

The indicators δ are true for exactly one sense, and correspond to the simple f_i defined above.[2] The sense with the highest vote value of $v(s)$ will be the sense with the highest posterior probability $P(s|s_1, \ldots s_k)$ and will be chosen.

All three of our combination schemes can be seen as ways of estimating the weights λ_i. For majority voting, we skip any attempt at statistical estimation and simply assign each λ_i to be $1/k$. For the maximum entropy classifier, we estimate the weights by maximizing the likelihood of a held-out set, using the standard IIS algorithm (Berger et al., 1996).

In weighted voting, we do something in between. We treat the δ functions as probabilities, treat $v(s)$ as a mixture model, and do a single round of EM to update the λ_i starting from uniform weights. As we move from majority voting to weighted voting to maximum entropy, the estimation becomes more sophisticated, but also more prone to overfitting. Since solving overfitting is hard, while choosing between classifiers based on held-out data is relatively easy, this spectrum gives us a way to gracefully handle the range of sparsities in the training corpora for different words.

2.2 Individual Classifiers

While our first-tier classifiers implemented a variety of classification algorithms, the differences in their individual accuracies did not primarily stem from the algorithm chosen. Rather, implementation details led to the largest differences. Naive-Bayes classifiers which chose sensible window sizes, or dynamically chose between window sizes tended to outperform those which chose poor sizes. Generally, the optimal windows were either of size one (which

[1] The number of features was not large, only one for each (classifier, chosen sense, correct sense) triple. However, most senses are rarely chosen and rarely correct, and so most features had zero or singleton support.

[2] If the nth classifier e_n returns s as the sense, then $\delta(s_n = s)$ is 1, otherwise it is zero.

detected syntactic or collocational cues) or of very large size (which detected more topical cues). Programs with hard-wired window sizes of, say, 5, performed poorly. Ironically, such middle-size windows were commonly chosen by students, but never useful; either extreme was a better design.

Another implementation choice dramatically affecting performance, also for naive-Bayes, was the amount and type of smoothing. Heavy smoothing and smoothing which backed off conditional distributions to the relevant marginal distributions gave good results, while insufficient smoothing or backing off to uniform marginals gave substantially degraded results.[3]

There is one significant way in which our first-tier classifiers were likely different from other teams' systems. In the original class project, students were guaranteed that the ambiguous word would only appear in a single orthographic form. Since this was not true of the SENSEVAL-2 data, we mapped the ambiguous words (but not their context words) down to a citation form. We suspect that this lost quite a bit of information, since there is considerable correlation between form and sense, especially for verbs, but we made no attempt to re-engineer the student systems, and have not thoroughly investigated how big a difference this stemming made.

3 Results and Discussion

Table 1 shows the results per word, and table 2 shows results by part-of-speech. A wide range of models are chosen, and the chosen model usually beats the best single classifier for that word, on average by 1.9%. The improvement over the globally best single classifier is even greater.

Notably, if we use the test data as an oracle to chose the best combination method, rather than relying on held-out data, accuracy jumps by an average of 3.6%. This gap is dramatically larger than the gap between the top scoring systems for this SENSEVAL-2 task. While the knowledge of actual best performance is obviously not available, one might suspect that a more sophisticated or better-tuned method of choosing a final combination model might lead to significant improvement.

[3]In particular, there is a defective behavior with naive Bayes where, when one smoothes far too little, the chosen sense is the one which has occurred with the most words in the context window. For skewed-prior data like the SENSEVAL-2 sets, this is invariably the common sense, regardless of what the context words are.

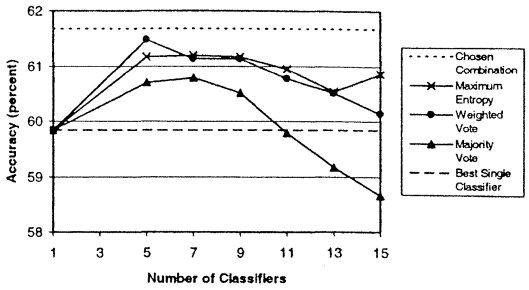

Figure 2: The accuracy of the various combination methods as the number of component systems changes. The *best single classifier* is chosen per word from held-out data and averaged. *Chosen combination* is also selected per word and averaged.

Figure 2 shows how the three combination methods' average scores varied with the number of component classifiers used. A critical aspect of our system is that the first-tier classifiers are very diverse, not only in implementation but also in performance. Initially, accuracy increases as added classifiers bring value to the ensemble. However, as lower-quality classifiers are added in, the better classifiers are steadily drowned out. The weighted vote and maximum entropy combinations are less affected by low-quality classifiers than the majority vote, being able to suppress them with low weights. Still, majority vote was a good method to have around for words where weights could not be usefully set by the other methods.

When combining heterogeneous classifiers, one would like to know when and how the combination will outperform the individuals. One factor is how complementary the mistakes of the individual classifiers are. We can measure this complementarity by averaging, over all pairs of classifiers, the fraction of errors that pair has in common. This gives average pairwise error independence. Another factor is the difficulty of the word being disambiguated. A high most-frequent sense baseline means that there is little room for improvement by combining classifiers. Figure 3 shows, for the overall top 7 first-tier classifiers, the absolute gain between their average accuracy and the accuracy of their majority. The x-axis is the difference between the pairwise independence and the baseline accuracy. The pattern is loose, but clear. The gain increases with complementarity and decreases with the baseline.

	Single		Combination			Oracle		Chosen	
word	base	sngl	vot7	wei7	me7	best	any	used	model
art-n	41.8	58.2	53.1	54.1	52.0	58.2	74.5	58.2	wei5
authority-n	33.7	70.7	70.7	70.7	68.5	76.1	92.4	72.8	wei5
bar-n	39.7	72.2	61.6	64.9	70.2	71.5	86.8	65.6	me9
begin-v	58.6	81.4	82.1	82.1	86.1	86.1	95.0	84.3	me15
blind-a	83.6	76.4	87.3	87.3	81.8	87.3	94.5	87.3	wei7
bum-n	75.6	55.6	75.6	75.6	71.1	75.6	91.1	64.4	me15
call-v	25.8	25.8	31.8	30.3	24.2	33.3	65.2	25.8	me5
carry-v	22.7	24.2	37.9	36.4	33.3	37.9	72.7	21.2	me15
chair-n	79.7	82.6	81.2	81.2	82.6	82.6	84.1	82.6	me5
channel-n	27.4	60.3	58.9	60.3	63.0	67.1	86.3	60.3	wei7
child-n	54.7	79.7	54.7	54.7	78.1	78.1	89.1	75.0	me15
church-n	53.1	73.4	75.0	75.0	75.0	76.6	90.6	75.0	me5
circuit-n	27.1	78.8	64.7	64.7	72.9	78.8	89.4	78.8	me5
collaborate-v	90.0	90.0	90.0	90.0	90.0	90.0	90.0	90.0	wei15
colorless-a	65.7	62.9	62.9	65.7	65.7	68.6	85.7	62.9	vot7
cool-a	46.2	53.8	55.8	55.8	48.1	59.6	84.6	48.1	me5
day-n	59.3	62.1	68.3	69.0	64.8	69.0	84.8	67.6	me5
detention-n	65.6	84.4	84.4	84.4	84.4	84.4	90.6	84.4	wei5
develop-v	29.0	29.0	34.8	34.8	34.8	42.0	69.6	33.3	vot13
draw-v	9.8	24.4	31.7	24.4	24.4	31.7	43.9	24.4	me5
dress-v	42.4	49.2	47.5	49.2	42.4	49.2	72.9	49.2	wei9
drift-v	25.0	28.1	25.0	25.0	28.1	34.4	75.0	25.0	vot7
drive-v	28.6	26.2	38.1	38.1	31.0	45.2	69.0	45.2	wei15
dyke-n	89.3	92.9	92.9	92.9	92.9	92.9	96.4	92.9	vot5
face-v	83.9	67.7	83.9	83.9	86.0	86.0	88.2	83.9	wei15
facility-n	48.3	67.2	67.2	69.0	63.8	74.1	91.4	65.5	wei15
faithful-a	78.3	78.3	78.3	78.3	78.3	78.3	100	78.3	wei15
fatigue-n	76.7	90.7	90.7	90.7	93.0	93.0	93.0	90.7	wei7
feeling-n	56.9	49.0	56.9	56.9	60.8	60.8	88.2	56.9	wei9
find-v	14.7	29.4	30.9	30.9	23.5	30.9	55.9	29.4	vot13
fine-a	38.6	51.4	57.1	58.6	60.0	61.4	80.0	55.7	me5
fit-a	51.7	82.8	89.7	89.7	79.3	89.7	96.6	89.7	wei9
free-a	39.0	53.7	57.3	57.3	61.0	61.0	75.6	61.0	me9
graceful-a	75.9	79.3	79.3	79.3	79.3	79.3	89.7	79.3	vot9
green-a	78.7	83.0	83.0	83.0	85.1	85.1	92.6	84.0	me15
grip-n	54.9	74.5	66.7	66.7	56.9	70.6	84.3	66.7	me11
hearth-n	75.0	62.5	75.0	62.5	62.5	75.0	87.5	75.0	vot15
holiday-n	83.9	83.9	83.9	83.9	83.9	83.9	96.8	83.9	me15
keep-v	37.3	47.8	38.8	50.7	47.8	52.2	68.7	47.8	me5
lady-n	69.8	77.4	79.2	79.2	77.4	79.2	83.0	79.2	wei7
leave-v	31.8	40.9	42.4	45.5	37.9	45.5	75.8	43.9	vot15
live-v	50.7	62.7	58.2	61.2	62.7	67.2	79.1	58.2	me15
local-a	57.9	68.4	71.1	71.1	68.4	73.7	92.1	68.4	vot15
match-v	35.7	47.6	45.2	45.2	45.2	54.8	83.3	42.9	me15
material-n	42.0	46.4	53.6	53.6	50.7	60.9	88.4	58.0	wei11
mouth-n	45.0	50.0	55.0	55.0	55.0	58.3	90.0	51.7	vot9
nation-n	70.3	73.0	70.3	70.3	73.0	73.0	83.8	73.0	me15
natural-a	27.2	55.3	47.6	47.6	47.6	55.3	79.6	52.4	wei13
nature-n	45.7	45.7	45.7	45.7	56.5	58.7	84.8	45.7	vot5
oblique-a	69.0	75.9	75.9	79.3	75.9	79.3	93.1	79.3	wei9
play-v	19.7	37.9	39.4	40.9	37.9	45.5	68.2	40.9	wei7
post-n	31.6	67.1	57.0	60.8	65.8	68.4	79.7	64.6	me13
pull-v	21.7	25.0	28.3	25.0	30.0	35.0	71.7	33.3	me11
replace-v	53.3	53.3	53.3	53.3	53.3	55.6	88.9	53.3	vot7
restraint-n	31.1	64.4	71.1	73.3	68.9	73.3	84.4	66.7	wei11
see-v	31.9	37.7	43.5	43.5	39.1	43.5	60.9	40.6	vot15
sense-n	22.6	52.8	60.4	58.5	52.8	64.2	83.0	60.4	vot11
serve-v	29.4	54.9	60.8	62.7	58.8	66.7	76.5	56.9	vot15
simple-a	51.5	54.5	51.5	51.5	54.5	54.5	83.3	53.0	me5
solemn-a	96.0	96.0	96.0	96.0	96.0	96.0	96.0	96.0	wei15
spade-n	63.6	63.6	78.8	78.8	81.8	81.8	81.8	75.8	wei15
stress-n	46.2	48.7	35.9	41.0	51.3	51.3	89.7	51.3	me9
strike-v	16.7	22.2	37.0	29.6	33.3	38.9	66.7	35.2	wei15
train-v	30.2	54.0	54.0	54.0	52.4	60.3	84.1	55.6	wei11
treat-v	38.6	47.7	54.5	56.8	47.7	59.1	95.5	54.5	vot7
turn-v	14.9	23.9	34.3	28.4	31.3	34.3	58.2	31.3	wei11
use-v	65.8	64.5	65.8	65.8	65.8	68.4	81.6	65.8	me9
vital-a	92.1	92.1	92.1	92.1	92.1	92.1	92.1	92.1	wei15
wander-v	80.0	80.0	82.0	82.0	80.0	82.0	82.0	80.0	me15
wash-v	25.0	66.7	33.3	58.3	50.0	58.3	83.3	25.0	vot15
work-v	26.7	50.0	45.0	41.7	43.3	45.0	76.7	41.7	wei13
yew-n	78.6	78.6	78.6	78.6	78.6	78.6	82.1	78.6	me15

Table 1: Results by word. Single classifiers: *base* = most-frequent-sense baseline, *sngl* = best single first-tier classifier as chosen on held-out data for that word. Fixed combinations: *vot* = majority vote, *wei* = weighted vote, *me* = maximum entropy combination; all are shown for the top seven classifiers only. Oracle bounds: *best* = best combination system as measured on the test data, *any* = test cases where at least one first-tier classifier produced the correct answer. Actually chosen: *model* shows which model performed best according to held-out data, and *used* shows its performance, which were our results for the SENSEVAL-2 English lexical sample task.

	Single		Combination			Oracle		Chosen
	base	sngl	vot7	wei7	me7	best	any	used
noun	50.5	67.0	65.8	66.4	67.7	71.7	86.6	68.3
adjective	57.8	67.1	68.0	68.4	67.8	71.1	86.7	68.6
verb	40.2	49.8	52.8	53.0	52.1	56.8	76.9	52.3
average	47.5	59.8	60.8	61.1	61.2	65.4	82.6	61.7

Table 2: Results by part-of-speech, and overall.

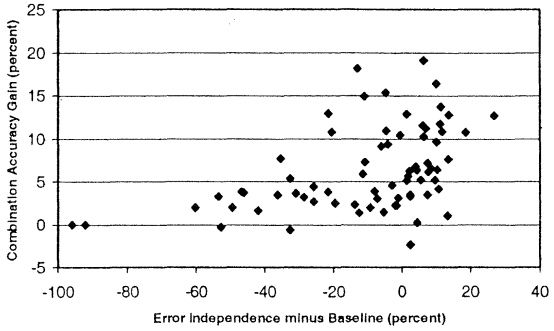

Figure 3: Gain in accuracy of majority vote over the average component performance as (pairwise independence − baseline accuracy) grows.

4 Conclusion

We have demonstrated that the combination of a number of heterogeneous classifiers can lead to a substantial performance increase over the individual classifiers. Our system is robust to both the wide range of accuracy of the first-tier classifiers and to sparsity of training data when building the second-tier classifier. The system's overall accuracy is high, despite the medium level of accuracy of the component systems.

5 Acknowledgments

We wish to thank the following people for contributing their classifiers to the Stanford-CS224N system: Zoe Abrams, Jenny Berglund, Dmitri Bobrovnikoff, Chris Callison-Burch, Marcos Chavira, Shipra Dingare, Elizabeth Douglas, Sarah Harris, Ido Milstein, Jyotirmoy Paul, Soumya Raychaudhuri, Paul Ruhlen, Magnus Sandberg, Adil Sherwani, Philip Shilane, Joshua Solomin, Patrick Sutphin, Yuliya Tarnikova, Ben Taskar, Kristina Toutanova, Christopher Unkel, and Vincent Vanhoucke.

References

A. L. Berger, S. A. Della Pietra, and V. J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22:39–71.

R. J. Mooney. 1996. Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. In *EMNLP 1*, pages 82–91.

The Språkdata-ML System as Used for SENSEVAL-2

Dimitrios KOKKINAKIS
Språkdata, Göteborg University
Box 200, SE-405 30
Göteborg, Sweden
Dimitrios.Kokkinakis @svenska.gu.se

Abstract

This paper describes the Språkdata-ML system as used in the SENSEVAL-2 exercise. The main focus of the paper is devoted to the process of feature extraction, preparation and organization of the test and training data.

Introduction

The methodology followed for sense disambiguation of the Swedish data by the Språkdata-ML system is supervised, based on Machine Learning (ML) techniques, particularly Memory Based Learning (MBL). The MBL implementation we used originates from the university of Tilburg in a system called TiMBL; details can be found in Daelemans *et al.* (1999). Thus, our main contribution in this task has been the effort to try and isolate a set of features that could maximize the performance of the MBL software. However, it is rather difficult to give the exact number of features and examples required for an adequate description of a word's sense or which algorithm performs best. We think that there is space for improvement of our system's performance by better modeling of the available resources (e.g. context, annotations), choice of parameters and algorithms, a claim that we have not explored to its full potential, further exploration is required. Intelligent example selection for supervised learning is an important issue in ML, an issue that we have not fully explored. In previous experiments for a similar problem for Swedish, the algorithm that performed best in TiMBL was a variant of the *k-nearest neighbor* (Mitchell, 1997) called IB1, an algorithm that we also used in the exercise; (Kokkinakis & Johansson Kokkinakis, 1999).

1 Data Preparation (Train)

To enhance the lexical disambiguation results using the available resources, we perform pre-processing in both the dictionary and the text to be sense-disambiguated. This is motivated by the fact that by making certain normalizations and simplifications in the resources we (hopefully) contribute to the production of qualitatively better results.

Initially, a text to be disambiguated is pre-processed by a tokeniser, a sentence boundary identifier, an idiom[1] and multiword identifier, a Name-Entity recogniser[2], a part-of-speech tagger, a lemmatiser and a semantic tagger[3]. Then, the input texts are transformed to the specified format that the MBL requires, which is feature-vectors of a specific length and content. The vectors we use consist of 102 features, the last two being the *id-number* and *class* or *sense* assigned to the vector. Since we do not know in advance which features will be useful for each particular word and sense, we chose to include features from a number of different information sources.

2 Vector Creation

The vectors consisted of: (i) selected information gathered from the dictionary entries (5 features); (ii) near-context (5 features); (iii) annotations applied on the training corpus (5

[1] The idioms originate from the Gothenburg Lexical Data Base/Semantic Database (GLDB/SDB) (http://spraakdata.gu.se/lb/gldb.html) and were used for the recognition and marking of idioms in the test/training corpus (over 4,000 idioms).

[2] See http://spraakdata.gu.se/svedk/ne.html for a demo.

[3] The semantic tagger originates from work by Kokkinakis *et al.* (2000) and uses the SIMPLE semantic classes for annotation (only nouns).

features); and (iv) information acquired from the lemmatised training corpus (85 features).

The corpus instances and dictionary were in XML format. An example of a corpus instance (1) for the first sense of the noun barn 'child' and a fragment of its dictionary description (2) are:

(1) **<instance id="barn.114"><answer instance="barn.114" senseid="barn_1_1" /> <context>**... försöken så att spädbarnen själva kunde styra de retningar som de utsattes för under försöket. Inom språkforskningen betyder det att **<head>**barnen**</head>** kan påverka hur olika talljud presenteras. När de får ... **</context> </instance>**

(2) <***lemma-entry*** id="barn_1" form= "barn" pos="n" inflection="~et =""><***lexeme*** id= "barn_1_1"><definition> människa som ej vuxit färdigt</definition> <definition-ext>till kropp och själ; under ngn åldersgräns som beror på samman-hanget</definition-ext> <synt-example>kvinnor och ~ släpptes fria </synt-example><synt-example>~ under 6 år kommer in gratis</synt-example><compound>spädbarn</compound>...<***cycle*** id=" barn_1_1_a"><trans>spec. om människa som ej nått pubertetsålder, straff-myndighetsålder etc.</trans><synt-example> ännu något år är hon ett ~</synt-example><compound> barnarbete </compound><compound>barnavårdsnämnd</compound></***cycle***>...</***lexeme***><***lexeme***>...<***cycle*** id=" barn_1_2_a"> <trans> äv. utvidgat, spec. om foster </trans><synt-example>hon är med ~ </synt-example><valency>med ~ </valency> </***cycle***>...</***lexeme***></***lemma-entry***>

2.1 Vector Creation (Dictionary)

The modeling of the vectors was performed in stages. The first stage of the processing uses the information from the dictionary. For every sense and sub-sense we extracted five representative nouns from the definition (and the definition extension) by applying part-of-speech tagging, lemmatization and exclusion of a number of *generic* nouns from a stop-list e.g. människa 'human' (a). If the number of nouns were less than five, we completed the list with compounds (if available).

Furthermore, the syntactic examples were used as training corpus and were added to the training instances (b). The valency information (if any) was also used in the same way (c). Consequently the amount of training material increased with 1,296 "new" disambiguated instances. A "dummy" *XXX* instance-number was given in these cases.

We did not put much effort on a more complex processing of the definitions since these are very short. The representations given below use the dictionary and corpus sample provided in (1) and (2).

(a) **<definition>**människa som ej vuxit färdigt**</definition><definition-ext>**till kropp och själ; under ngn åldersgräns som beror på sammanhanget **</definition-ext>**
become: barn_1_1: kropp, själ, åldersgräns

(b) **<synt-example>**kvinnor och ~ släpptes fria**</synt-example>**
become: **<instance id="barn.XXX"> <answer instance="barn.XXX" senseid=** "barn_1_1"**/> <context>** kvinnor och **<head>**barn**</head>** släpptes fria **</context></instance>**

(c) **<valency>**med ~**</valency>**
become: **<instance id="barn.XXX"> <answer instance="barn.XXX" senseid=** "barn_1_2_a"**/><context>** med **<head>**barn **</head> </context></instance>**

2.2 Vector Creation (Near Context)

The second stage involved the use of the near-context. Punctuation, auxiliary verbs and a number of other stop-words were removed and the surrounding tokens (±2) of each headword in the corpus were extracted (d). Only the lemma form of the headwords was used, and the context was not lemmatized:

(d) **<instance id="barn.114"><answer instance="barn.114" senseid=** "barn_1_1" **/><context>**... språkforskningen betyder ~~det att~~ **<head>**barn~~en~~ **</head>** ~~kan~~ påverka ~~hur~~ olika ...**</context> </instance>**
became: **<instance id="barn.114"> <answer instance="barn.114" senseid="** barn_1_1"**/><context>**språkforskningen betyder **<head>**barn**</head>** påverka olika **<context></instance>**

2.3 Vector Creation (Global Features)

During the third stage, the training corpus was processed by a name-entity recognizer (e.g. HUMAN, TIME), an idiom identifier (IDIOM) and a semantic tagger (e.g. BIO, ETHNOS, PHENOMENON). The annotations produced by these tools were gathered in the form of a list of labels, and the five most frequent in the respective set of instances for each sense and sub-sense were used in the vectors. For example, for the sense barn_1_1 the five most frequent annotations found in all training instances were: BIO, ORGANIZATION-AGENCY, LOCATION, SITU and OCCUPATION-AGENT.

2.4 Vector Creation (Global Context)

Often, near-context cannot distinguish between different senses. In such cases it is useful to look at a larger context and extract keywords representative for each sense. We made a frequency list of all noun and verb occurrences for all corpus instances for each sense. From the produced lists, 85 keywords per sense were extracted by eliminating high frequency (a word occurred in more than *X* percent of the cases with the sense) and low frequency words (a word occurred at least *Z* times in the list). For the sense barn_1_1 the 85 keywords included:

> ansikte, ansvar, apparatur, arm, avvikelse, barnmorska, barnomsorg, beredskap, betala, bild, detalj, dialog, djur, docka, erfarenhet, fel, föreställning, förslag, ...

After the collection and combination of the 95 features common to a sense (stages i, iii, iv in Section 2, e1), a complete case for a sense was produced (e2):

(e1) *Lemma_SENSE: 5 words from the dictionary information, 5 "semantic" labels, 85 representative words from the global context*

(e2) barn_1_1: kropp, själ, småbarn, spädbarn, åldersgräns, BIO, ORGANIZATION-AGENCY, LOCATION, SITU, OCCUPATION-AGENT, ansikte, ansvar, apparatur, arm, avvikelse, barnmorska, barnomsorg, ...

We assume then, that for each training instance the above list is "true" and we convert the training instances into vectors of 102 features, where the 95 positions of the features in each vector were substituted with '1' keeping intact the near context. Thus, the truncated training instance in (f) was re-formatted to (g):

(f) **<instance id="barn.114"><answer instance="barn.114" senseid= "barn_1_1" /><context>**språkforskningen betyder **<head>**barn**</head>** påverka olika **<context></instance>**

(g) språkforskningen, betyder,**<head>**barn **</head>**, påverka, olika, 1, 1, 1, 1, 1, 1, 1,..., barn.114, barn_1_1.

3 Data Preparation (Test)

The test material consisted of 1,525 corpus instances in the same format as the previous training example, but without any designation of the correct *senseid*. The material was processed in a similar manner as the training one. The major difference lies in the fact that at the vector-creation stage we used the feature-vectors representative for a sense, example (e) previously, and we compared them with the features produced for each test instance. A feature at a specific position then was assigned '1' if the feature in the test occurred in the representative feature vector or '0' otherwise. For instance, the test instance in (h) was transformed, after processing, to a 102-feature-vector.

(h) **<instance id="barn.114"><answer instance="barn.114" senseid= "??????" /><context>**I jungfrukammaren innanför köket bodde en kokerska och en husa. [Ett hus fyllt av minnen] Huset är fyllt av minnen. I fotoalbumen kan vi se farmor omgiven av sina små vitklädda **<head>**barn**</head>** och pappa i sjömanskostym lutad mot en björk. I farfars svarta, snidade skrivbord ... **</context> </instance>**

The class of the representative sense-vector that produced more '1's for the test instance was chosen as the class of that instance. In (i) there are four '1's which means that the specific test instance had four common features with the representative vector for sense barn_1_2_a, and less than four for all the other representative vectors for the rest of the senses for barn. Thus, the class for the test instance is assigned that sense (which may be altered by the MBL software during the nearest-neighbor

calculation). Thus, the test instance in (h) was transformed to the format illustrated in (i). The four '1's denote that there were four features in common with the representative vector for barn_1_2_a, the rest of the representative sense-vectors for barn (e.g. barn_1_1_a, barn_1_1_b etc.) had less common features than four, and so barn_1_2_a was chosen:

(i) små, vitklädda, <head>barn</head>, pappa, i, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, barn.114, barn_1_2_a

The training and test feature vectors were then fed to the TiMBL software, where the IB1 algorithm (nearest neighbor search) was used.

4 Results

Table 1 shows the evaluation of the test material. Since answers were provided for the whole material, precision and recall obtain the same value. Coarse-grain evaluation was not used, however coarse-grained is considered the least interesting of the three measures.

	INSTANCES	FINE	MIXED
ADJECTIVES	191	48,2%	54,4%
NOUNS	616	71,3%	74,9%
VERBS	718	57,8%	66,1%
MOST FREQ. BASELINE	45,3%		
WHOLE SAMPLE	**1,525**	**62,0%**	**68,2%**

Table 1. Official results for the Språkdata-ML system

Conclusion

The existence of sense ambiguity (polysemy and homonymy) is one of the major problems affecting the usefulness of basic corpus exploration tools. In this respect, we regard sense disambiguation as a very important process and component when it is seen in the context of a wider and deeper text-processing architecture. In this paper we have described a simple feature-vector extraction approach to sense disambiguation that was utilized in a MBL software. We do not believe that we have fully exploited the capabilities of either the software or the way we can model the available resources. These issues will be investigated in the future, as well as the evaluation of the sense-tagger on an even larger scale.

References

Daelemans W., Zavrel J., van der Sloot K. and van den Bosch A. (1999). *TiMBL: Tilburg Memory Based Learner, version 2.0, Reference Guide*. ILK Technical Report 99-01, Paper available from: http://ilk.kub.nl/~ilk/papers/ilk9901.ps.gz.

Kokkinakis D. and Johansson Kokkinakis S. (1999). Sense Tagging at the Cycle-Level Using GLDB. *Nordiska Studier i Lexikografi*, vol. 27:146-167. Gellerstam M., Jóhannesson K., Ralph B. and Rogström L. (eds). Nordiska Föreningen för Lexikografi & Meijerbergs Institut för Svensk Etymologisk Forskning.

Kokkinakis D., Toporowska Gronostaj M. and Warmenius K. (2000). Annotating, Disambiguating & Automatically Extending the Coverage of the Swedish SIMPLE Lexicon. *Proceedings of the 2nd Languages Resources and Evaluation Conference (LREC)*, vol. III:1397-1404. Athens, Hellas.

Mitchell T. M. (1997). *Machine Learning*. McGraw-Hill Series on Computer Science.

ATR-SLT System for SENSEVAL-2 Japanese Translation Task

Tadashi Kumano, Hideki Kashioka and **Hideki Tanaka**
ATR Spoken Language Translation Research Laboratories
2-2-2 Hikaridai Seika-cho Soraku-gun Kyoto 619–0288 JAPAN
{tadashi.kumano, hideki.kashioka, hideki.tanaka}@atr.co.jp

Abstract

We propose a translation selection system based on the vector space model.

When each translation candidate of a word is given as a pair of expressions containing the word and its translation, selecting the translation of the word can be considered equivalent to selecting the expression having the most similar context among candidate expressions. The proposed method expresses the context information in "context vectors" constructed from content words co-occurring with the target word. Context vectors represent detailed information composed of lexical attributes(word forms, semantic codes, etc.) and syntactic relations (syntactic dependency, etc.) of the co-occurring words.

We tested the proposed method with the SENSEVAL-2 Japanese translation task. Precision/recall was 45.8% to the gold standard in the experiment with the evaluation set.

1 Introduction

The SENSEVAL-2 Japanese translation task defines a sense of a Japanese word as an English translation. The same Japanese word in different contexts may have different English translations; therefore, translation ambiguity arises.

Translation Memory (henceforth TM) defining word senses were given to the task participants. Each target word has translation pairs of Japanese and English expressions as word sense candidates[1]. The target word is marked in the Japanese expression, but the corresponding part is unspecified in the English expression. Hence, selecting the most appropriate translation of the target Japanese word in the evaluation expression can be considered to be equivalent to selecting the expression with the most similar context in the TM. This is equivalent to the word sense disambiguation problem in a single language.

[1]Each target word has 21.6 pairs on average.

Generally, word sense disambiguation uses context information, such as the frequency of words that co-occur with the target word. The context information is learned from the correctly-annotated training corpora. However, no training corpus was given for the task and the given TM had shorter contexts because the TM expressions were rather incomplete. Therefore, instead of learning the co-occurring words with the target word from the training corpora, we extract detailed information from the TM expressions as context information. We utilize the information of co-occurring words with the target word (context words) as shown below.

- lexical attributes (word form, part-of-speech, semantic codes on thesaurus, etc.)
- syntactic relations to the target word (dependency relation, etc.)

We employed the vector space model, which is used for text retrieval (Salton and McGill, 1983) to calculate the similarity between the context word information of evaluation expressions and those of the TM. The detailed context information are expressed as "context vectors." We use cosine values between context vectors as a measure of similarity.

In this paper, we will explain first how to construct "context vectors," and then show the accuracy of the selection experiment to the correct data (gold standard).

2 Translation Selection Using Context Vectors

2.1 Context Vectors

2.1.1 Concept

We will explain how to construct a context vector from an expression e_1 with the target word "間 (*aida*; interval)", as an illustration.

Figure 1 shows the expression, which contains the content words "夫婦 (*fuufu*; married couple)", "子供 (*kodomo*; child)", and "産まれ

Table 1: Context Vectors Construction

	Type of syntactic relationship to the target word											
	modifying target word in case relation:				modified by target word in case relation:				target word	...	following words	all context words
	WO	*NO*	*NI*	...	*WO*	*NO*	*NI*	...				
(e_1) "夫婦 の 間 に 子供 が 産まれる (a baby is born to the couple)"	ϕ	*fuufu*	ϕ	ϕ	ϕ	ϕ	*umareru*	ϕ	*aida*	...	*kodomo* *umareru*	*fuufu* *kodomo* *umareru*
(e_2) "仕事 の 間 を ぬって 見舞い に 行く (to visit in hospital at the interval during one's work)"	ϕ	*shigoto*	ϕ	ϕ	*nutte*	ϕ	ϕ	ϕ	*aida*	...	*nutte* *mimai* *iku*	*shigoto* *nutte* *mimai* *iku*

fuufu-no aida-ni kodomo-ga umareru (for e_1); *shigoto-no aida-wo nutte mimai-ni iku* (for e_2)

$= \mathbf{c}_{e_1}$, $= \mathbf{c}_{e_2}$

$\lambda_{modifying\_TW} : \lambda_{modified\_by\_TW} : \lambda_{target} : \cdots : \lambda_{follow} : \lambda_{all}$

The ratio of vector components for each word attribute

る (*umareru*; be born)", and shows that the phrases containing these content words have some syntactic dependencies.

We then prepare a table that enumerates all possible syntactic relations between target word and context words, as in Table 1. For each expression, we then insert corresponding words to the column for each syntactic relation. For example, the row for e_1 of Table 1 can be obtained by the enumeration of expression e_1. If a syntactic relation is applicable to several words, such as the relation "following words" in Table 1, all of them are enumerated in the same column. If no content word comes under the syntactic relation, it is assigned empty (ϕ).

Each row of the table is designated a "context vector" $\mathbf{c}_e$ of a corresponding expression e.

2.1.2 Calculation of Context Vectors

In the preceding section, the table was explained as if it had context words in its elements, but "word attribute vectors" of context words are assigned to them practically. Hence, context vectors are the conjunctions of "word attribute vectors." Each word attribute vector $\mathbf{a}_w$ of a word w expresses lexical attributes of w, such as POS or semantic code. Word attribute vectors have a fixed dimension number, and each element has a non-negative value. The procedure for constructing word attribute vectors will be described below in Section 2.1.3.

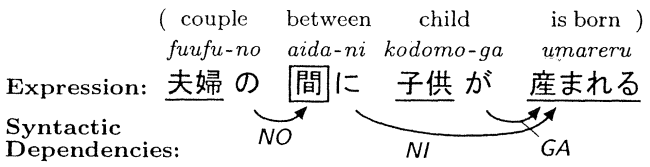

Figure 1: Syntactic Dependencies in Expression e_1

When several context words fall under the same syntactic relation like *kodomo* and *umareru* as we can see in the "following words" relation in Table 1, the word vectors assigned to the relation is calculated by selecting the maximum value for every vector component among values of all words in that relation. The calculation named vecmax is defined as follows:

$$\underset{i=1\ldots m}{\text{vecmax}}\ \mathbf{a}_i = (b_1, b_2, \ldots, b_n),$$

where

$$\begin{cases} \mathbf{a}_i \text{ is a } n\text{-dimensional vector,} \\ a_{ij} \text{ is a } j\text{-th element of vector } \mathbf{a}_i, \text{ and} \\ b_j = \max\limits_{i=1\ldots m} a_{ij}. \end{cases}$$

When joining word attribute vectors into a context vector, each word attribute vector is given a weight in order to get a certain ratio of vector components for each syntactic relation. This is necessary to specify the degree of the contribution to the context vectors according to the type of syntactic relation. For example, assuming that the ratio of the vector components is specified using λ_{syn_rel} (*syn_rel* denotes a specific syntactic relation type) as shown in Table 1, the context vector $\mathbf{c}_{e_1}$ of the expression e_1 will be calculated as follows:

$$\mathbf{c}_{e_1} = \cdots \oplus \lambda_{modifying\_TW} \cdot \frac{\mathbf{a}_{fuufu}}{|\mathbf{a}_{fuufu}|} \oplus \cdots \oplus \lambda_{modified\_by\_TW} \cdot \frac{\mathbf{a}_{umareru}}{|\mathbf{a}_{umareru}|} \oplus \cdots$$

Table 2: Constructing Word Attribute Vectors

	Type of syntactic attribute											
	Emergent Form				Pronunciation				POS			
	夫婦	子供	産まれる	···	*fu-u-fu*	*ko-do-mo*	*u-ma-re-ru*	···	noun	verb	···	
$\mathbf{a}_{fuufu} = ($	η_{e_form}	0	0	0	η_{e_pron}	0	0	0	η_{pos}	0	0	)
$\mathbf{a}_{kodomo} = ($	0	η_{e_form}	0	0	0	η_{e_pron}	0	0	η_{pos}	0	0	)
$\mathbf{a}_{umareru} = ($	0	0	η_{e_form}	0	0	0	η_{e_pron}	0	0	η_{pos}	0	)

	Type of syntactic attribute													
	Semantic Code													
	N86	N85	N74	N72	N5	N4	N3	N2	N1	P26	P17	P16	P1	···
	0	0	$\frac{\eta_{sem}}{\sqrt{7}}$	$\frac{\eta_{sem}}{\sqrt{7}}$	$\frac{\eta_{sem}}{\sqrt{7}}$	$\frac{\eta_{sem}}{\sqrt{7}}$	$\frac{\eta_{sem}}{\sqrt{7}}$	$\frac{\eta_{sem}}{\sqrt{7}}$	$\frac{\eta_{sem}}{\sqrt{7}}$	0	0	0	0	0)
	$\frac{\eta_{sem}}{\sqrt{9}}$	$\frac{\eta_{sem}}{\sqrt{9}}$	$\frac{\eta_{sem}}{\sqrt{9}}$	$\frac{\eta_{sem}}{\sqrt{9}}$	$\frac{\eta_{sem}}{\sqrt{9}}$	$\frac{\eta_{sem}}{\sqrt{9}}$	$\frac{\eta_{sem}}{\sqrt{9}}$	$\frac{\eta_{sem}}{\sqrt{9}}$	$\frac{\eta_{sem}}{\sqrt{9}}$	0	0	0	0	0)
	0	0	0	0	0	0	0	0	0	$\frac{\eta_{sem}}{\sqrt{4}}$	$\frac{\eta_{sem}}{\sqrt{4}}$	$\frac{\eta_{sem}}{\sqrt{4}}$	$\frac{\eta_{sem}}{\sqrt{4}}$	0)

$$\oplus\ \lambda_{target} \cdot \frac{\mathbf{a}_{aida}}{|\mathbf{a}_{aida}|} \oplus \cdots$$

$$\oplus\ \lambda_{follow} \cdot \frac{\underset{i\in\{kodomo,umareru\}}{\text{vecmax}} \mathbf{a}_i}{\left|\underset{i\in\{kodomo,umareru\}}{\text{vecmax}} \mathbf{a}_i\right|}$$

$$\oplus\ \lambda_{all} \cdot \frac{\underset{i\in\{fuufu,kodomo,umareru\}}{\text{vecmax}} \mathbf{a}_i}{\left|\underset{i\in\{fuufu,kodomo,umareru\}}{\text{vecmax}} \mathbf{a}_i\right|}.$$

2.1.3 Word Attribute Vectors

For lexical attributes, we prepare another table similar to that for context words described in the previous section. Table 2 shows that the table enumerates attributes for all words appearing for each lexical attribute. For each word, values are assigned to the column corresponding to the lexical attribute. The value zero is assigned to the column when the lexical attribute is not applicable to the word. In Table 2, the lexical attributes of each context word in expression e_1 are expressed in each row. The row is called "word lexical attributes" $\mathbf{a}_w$ of the corresponding word w.

We employ the semantic codes of a Japanese thesaurus as the semantic attributes. A semantic code may have superordinates because a thesaurus represents semantic relations on the hierarchical tree structure. For example, the word *fuufu* has semantic codes on seven levels, from "Noun 74" on the leaf node to "Noun 1" on the top, in the thesaurus "Nihongo Goi Taikei (Ikehara et al., 1997)" that we used. We treat all semantic codes as semantic attributes of word attribute vectors, and assign values to the corresponding elements equally.

Each lexical attribute of a word attribute vector should be assigned a value, the ratio of component vectors for each word lexical attribute being the specific value η_{word_attr} ($word\_attr$ denotes a specific word attribute type) in Table 2. Semantic attributes may have multiple components to be assigned values, each component should be normalized by the number of the components (See Table 2).

2.2 Translation Selection

To select an appropriate translation for an evaluation expression containing a target Japanese word, we need to compare the context vector of the evaluation expression with the context vectors of all candidate Japanese expressions in the TM. We then choose the candidate whose cosine value to the context vector of the evaluation expression is the maximum.

Each expression should have a unique context vector in order to compare context vectors. But context words, like target words, have ambiguity, and they have several candidates for semantic codes in the thesaurus. It seems unacceptable that the method requires disambiguation of context words before disambiguation of the target word. Therefore, we decided not to disambiguate context words before constructing the context vector. Instead, we construct "context vector candidates" from all combinations of the context word candidates. All combinations of the context vector candidates are used for calculating similarity, and the combination that has the maximum value is selected as the pair of the evaluation and the TM expressions. We can resolve ambiguity of context words when selecting the translation of the target word.

3 Description of Participating System

3.1 Resources, etc.

Our system used the following resources in addition to the given TM and evaluation set.

Table 3: Employed Parameters

word attribute type	ratio
Emergent Word Form	1
Pronunciation	1
Standard Form	4
(standard) Pronunciation	4
Part-Of-Speech	0
Conjugated Form	1
Semantic Code	12

syntactic relation type	ratio
modifying target word (case relation: specific)	3
(case relation: non-specific)	1
modified by target word (case relation: specific)	3
(case relation: non-specific)	1
target word	2
the phrase containing target word	2
preceding target word	1
following target word	1
all content words	2

Japanese Morphological Analyzer:
JUMAN (Kurohashi and Nagao, 1998)

Japanese Syntactic Analyzer:
KNP (Kurohashi, 1998)

Thesaurus:
Nihongo Goi Taikei (Ikehara et al., 1997)

3.2 Parameters

The following parameters have significant effects on the accuracy of our method.

1. The η_{word_attr} ratio of vector components specified for each word attribute when making word attribute vectors (Section 2.1.3)
2. The λ_{syn_rel} ratio of the vector components specified for each syntactic relation when joining word attribute vectors into context vectors (Section 2.1.2)

However, we did not optimize the parameters in our participating system, because of the task specification that no training corpus was given and the time limitations in the course of system development. Parameters were given manually by considering the parameter functions. All of the lexical and syntactic attributes and parameters that represent the ratio between attributes, which our participating system employed, are shown in Table 3.

4 Evaluation

Our participating system marked both the precision and the recall at 45.8% of the correct data (the gold standard) in the evaluation corpus selection. However, our participating system had some serious bugs in the vector normalization process. After correcting the bugs, we made another selection experiment using the same parameters described in Section 3.2. The accuracy of the corrected system was 49.3% (nouns: 50.0%, predicates: 48.5%).

5 Summary

We proposed a translation selection method for the SENSEVAL-2 Japanese translation task. The proposed method calculates the similarity between an evaluation expression containing the target word and Japanese expressions containing the same word in the TM. For calculating similarity, "context vectors" are constructed. Context vectors represent lexical attributes of context words and syntactic relations between context words and the target word. The system employed the proposed method with an accuracy of 49.3% after bug elimination.

Future plans are as follows.

1. To optimize parameters using the gold standard. We would like to use the optimized parameters to study the relation between context information type and accuracy on translation selection. In addition, we will examine whether employed lexical and syntactic attributes are appropriate for the task.
2. To apply the machine learning method to the task, preparing the training corpora. We will make use of the detailed context information proposed, the lexical and syntactic attributes, at machine learning.

References

S. Ikehara, M. Miyazaki, S. Shirai, A. Yokoo, H. Nakaiwa, K. Ogura, Y. Oyama, and Y. Hayashi, editors. 1997. *Nihongo Goi Taikei*, volume 1–5. Iwanami Shoten. (in Japanese).

S. Kurohashi and M. Nagao, 1998. *Japanese Morphological Analysis System JUMAN version 3.61.* Kyoto University. (in Japanese).

S. Kurohashi, 1998. *Japanese Syntactic Analysis System KNP version 2.0 b6 user's manual.* Kyoto University. (in Japanese).

G. Salton and M. J. McGill. 1983. *Introduction to Modern Information Retrieval.* McGraw-Hill.

Sense and Deduction: The Power of Peewees Applied to the SENSEVAL-2 Swedish Lexical Sample Task

Torbjörn Lager† and **Natalia Zinovjeva**‡
†Department of Linguistics, Uppsala University
‡Hapax Information Systems AB, Stockholm

Abstract

This paper describes our use of Prolog Word Experts (PWEs) in the SENSEVAL-2 competition. We explain how we specify our PWEs as sequences of transformation rules and how they can be trained on sense tagged corpus data. We give a semantics of PWEs by translating them into first order predicate logic, and we describe how PWEs can be compiled into Prolog procedures. We finally present our results for the Swedish lexical sample task: 63% (fine-grained score) for our best PWE, and a second place in the ranking.

1 Introduction

Word experts are small expert system-like modules for processing a particular target word based on neighboring words. Typically, a word expert uses rules that test the identity and relative position of words in the context in order to infer the role of the target word in the passage (Berleant, 1995). In this paper, we describe the development of various kinds of word experts in a logic programming framework, dealing with word sense disambiguation in the context of the SENSEVAL-2 competition.

In a logic programming framework, the task of engineering a word (sense) expert can be specified as follows. Given a suitable representation of a text, we want to define a predicate `sense/2` such that `sense(P,S)` is true iff the word at position `P` in the text has the sense `S`. In the remainder of the paper, we will refer to this kind of word expert as a Prolog Word Expert (or PWE for short – "Peewee" to its friends). This is to distinguish it from other kinds of word experts, and to emphasize the fact that it is 'programmed in logic'.

2 The Anatomy of a Peewee

2.1 Peewee Specifications

In the present paper, a word expert's knowledge will be expressed, not as Prolog clauses defining `sense/2` directly, but as a sequence of transformation rules. For example, here is how we specify a word expert which is able to disambiguate occurrences of *interest*:[1]

```
word_expert sense :=
  sense:add 6 <- word:interest@[0] o
  sense:6>1 <- word:in@[1] o
  sense:1>5 <- word:'%'@[-1] o
  end.
```

The first rule works as a default rule, which simply assigns the most frequent sense to the word *interest* (6 in this case). If no other rules apply, this is the tag that the word will eventually get. The other rules dictate when – based on the context – a word should have its tag changed. The second rule is to be read "replace the tag for sense 6 with the tag for sense 1, if the next word is *in*". The third rule says "replace the tag for sense 1 with the tag for sense 5, if the previous 'word' is '%'." The `o`-symbol is a composition operator, and `(R o Rs)` basically means that the output of applying the rule `R` forms the input to the application of the rules `Rs`. Thus, rules are strictly order-dependent. Note, for example, that the third rule is applicable only if the second rule is.

Needless to say, the above rules are not at all sufficient for the task of disambiguating all uses of *interest*. But the number of rules can be increased, and typically a word expert will

[1]This word was of course not used the Swedish task, but is used here for expository reasons. The sense tags are numbers: 1="readiness to give attention", 5="a company share", 6="money paid for the use of money", etc.

have access to anything between just a handful of rules and several hundred ones.[2]

2.2 Peewee Logic

Interestingly, a sequence of transformation rules can be translated into a set of axioms, expressed in first-order predicate logic, defining relationships between positions in a text, word forms, and senses (Lager, 2000; Lager & Nivre, 2001). For example, the meaning of the rules from the previous section can be spelled out as follows:

$$\forall p[w(p,interest) \rightarrow S_1(p,6)]$$

$$\forall p_0,p_1[S_1(p_0,6) \wedge p_1{=}p_0{+}1 \wedge w(p_1,in) \rightarrow S_2(p_0,1)]$$
$$\forall p_0,p_1,x[S_1(p_0,x) \wedge p_1{=}p_0{+}1 \wedge \neg w(p_1,in) \rightarrow S_2(p_0,x)]$$

$$\forall p_0,p_1[S_2(p_0,1) \wedge p_1{=}p_0{-}1 \wedge w(p_1,\%) \rightarrow S_3(p_0,5)]$$
$$\forall p_0,p_1,x[S_2(p_0,x) \wedge p_1{=}p_0{-}1 \wedge \neg w(p_1,\%) \rightarrow S_3(p_0,x)]$$

$$\forall x,p[S_3(p,x) \rightarrow S(p,x)]$$

The idea is that for each rule in the sequence a new predicate S_i is introduced, where the subscript indicates where in the sequence the rule belongs. Semantically, S_i relates a position to a sense, and the formulas define this predicate in terms of the predicate S_{i-1} plus a number of other predicates. Each S_i corresponding to a replacement rule is defined by two sentences – one stating the conditions under which a sense tag is replaced with another sense tag, the other one stating the conditions under which the old sense tag is kept.

Given a suitable logical representation of a text, such as

$$w(1,Sue)\ \ w(2,developed)\ \ w(3,an)\ \ w(4,interest)$$
$$w(5,in)\ \ w(6,computers)\ \ w(7,and)\ \ w(8,bought)$$
$$w(9,an)\ \ w(10,11.5)\ \ w(11,\%)\ \ w(12,interest)$$
$$w(13,in)\ \ w(14,Microsoft)$$

and given a suitable constructive proof method, the exact identity of the sense of an occurrence of the word *interest* – say the word at position 12 – will follow as a logical consequence of the theory formed by taking the union of the previous two sets of formulas. For example, the formula $\exists x[S(12,x)]$ is a theorem, for which we can construct (only) the example $x \rightarrow 5$, and we have thus formally proved that this particular occurrence of *interest* means "a share in a company".[3]

What we have here is something that we like to think of as *word sense disambiguation as deduction*, in analogy to the ideas of *parsing as deduction* due to Pereira and Warren (1983).

2.3 The Peewee Compiler

Since the above formulas have already logic programming form, it is straightforward to translate them into Prolog. For example, the second and the third formulas can be translated as follows:[4]

```
s2(P0,1) :- s1(P0,6), P1 is P0+1, w(P1,in).
s2(P0,X) :- s1(P0,X), P1 is P0+1, \+ w(P1,in).
```

To write Prolog procedures such as these by hand for many rules would be tedious and prone to errors. Fortunately, since the formalism for transformation rules is compositional, it was straightforward to write a compiler[5] that generates word expert procedures from word expert specifications automatically.

2.4 Peewee Training

There is an obvious choice of learning method for training Prolog Word Experts, namely Transformation-Based Learning (Brill, 1995). Of course, the fact that transformation rules can be learned from tagged corpora was a major reason for using them in the first place. The μ-TBL system – described in detail in (Lager, 1999) – uses the search and database capabilities of the Prolog programming language to implement a generalized form of transformation-based learning. Through its support of a compositional rule/template formalism and 'pluggable' algorithms, the μ-TBL system can easily be tailored to different learning tasks.[6]

Rules that can be learned in Transformation-Based Learning are instances of rule templates. For example, the second of the rules in our example PWE specification is an instance of the following template:

```
sense:A>B <- word:C@[1].
```

[2]A demo of a more potent PWE is available at: `http://www.ling.gu.se/~lager/Home/pwe_ui.html`

[3]The theory can be used in other ways too. *Searching* for a word token with a particular sense (say 5) becomes a matter of constructively proving $\exists p[S(p,5)]$.

[4]There are equivalent but more efficient ways to represent these clauses in Prolog (cf. Lager, 2000).

[5]Download the compiler from the PWE homepage at: `http://www.ling.gu.se/~lager/pwe.html`

[6]The μ-TBL system is available from: `http://www.ling.gu.se/~lager/mutbl.html`

The template is to be read "replace the tag for sense A with the tag for sense B if the word immediately to the right is C", where A, B and C are variables. Learning is a matter of repeatedly instantiating rule templates in training data, scoring rules on the basis of counts of positive and negative evidence of them, selecting the highest scoring rule on the basis of this ranking, and applying it to the training data.

3 Peewees at SENSEVAL-2

The lexical sample task for Swedish in SENSEVAL-2 involved 40 lemmas: 20 nouns, 15 verbs and 5 adjectives. Together they represented 145 senses and 304 sub-senses. 8,718 annotated instances were provided as training material and 1,525 unannotated instances were provided for testing. Furthermore, a lexicon – the GLDB (Gothenburg Lexical Database) – complete with morphological information, definitions, language examples, etc. was available.

Our team explored three approaches. For each lemma, we trained:

- PWE-smpl: a simple PWE capable of arriving at a single sense for each instance of that lemma in the testing material.
- PWE-disj: a committee of PWEs (i.e. a set of PWEs) capable of arriving at (possibly) multiple senses for each instance of that lemma, by collecting the individual results into a set.
- PWE-vote: a committee of PWEs capable of arriving at a single sense for each instance of that lemma, by applying a simple voting procedure.

As it turned out, the second of these approaches produced a rather unimpressive result, and we will therefore spend very little time discussing it. Indeed, had we been able to run the scoring software ourselves (which we were not), we would have left them outside the competition altogether.

3.1 The Simple Peewees

For the training of our simplest form of sense disambiguation expert, the following set of seven templates was used:

```
sense:A>B <- word:C@[-1].
sense:A>B <- word:C@[-1,-2].
sense:A>B <- word:C@[1].
sense:A>B <- word:C@[1,2].
sense:A>B <- word:C@[1] & word:D@[2].
sense:A>B <- word:C@[-1] & word:D@[-2].
sense:A>B <- word:C@[-1] & word:D@[1].
```

The idea was to exploit a fact noted by many researchers in the field: that the sense of an occurrence of a word can fairly successfully be determined from just looking at the two previous words and the two following words (cf. Ide & Véronis, 1998). The choice of the above set of templates is based on a fairly thorough trail-and-error process and works well for most words that we have tried.

3.2 The Peewee Committees

The idea here was to train five different PWEs for each lemma, and then to use a simple voting mechanism to arrive at a final decision. The PWEs were different only in that they used different sets of templates during the training. Templates looking forwards only, templates looking backwards only, and templates looking both forwards and backwards. Furthermore, one member in each committee was trained for using a bag-of-words approach to disambiguation, based on templates of the following form:

```
sense:A>B <- inBag:W@[0].
sense:A>B <- inBag:W1@[0] & inBag:W2@[0].
```

Finally, one PWE in each committee had access to a list of words extracted from the language examples provided by the GLDB.

3.3 The Procedure

In this section we describe the actions that we took in order to submit our entry in the competition.

- In a preparatory step, the XML formatted training data was parsed and subsequently converted into the format required by the μ-TBL system.
- The training was performed, and resulted in one PWE specification per lemma. Training took between 5 seconds and a couple of minutes per lemma, depending on the amount of training data available for the lemma in question.
- The PWE specifications were compiled into a set of PWE procedures, by means of the PWE compiler.

- Simple procedures were written to print the results to a file in the prescribed format, and the PWEs were then run on the test data. This took only a couple of seconds for the whole test corpus.

3.4 Results

In the following table we show the results of our entry in the competition, copied from the SENSEVAL-2 homepage.[7]

System	Evaluation	Accuracy (%).
PWE-smpl	Fine	61.1
	Mixed	66.8
PWE-vote	Fine	63.0
	Mixed	68.6

Five groups and altogether eight systems participated in the Swedish lexical sample task. In terms of ranking, our PWE-vote came in second, after Yarowski's JHU system, and before the Göteborg team's best entry. However, we hasten to add that the step from Yarowski's (nearly 70%, fine grained evaluation) to our results is a very significant 7%, and that the step down to Göteborg's result is very small and probably statistically insignificant. Our simple Peewees shared the fourth place with Resnik et al.'s UMD-SST.

As can be seen from the table, the PWE committees did slightly better than a single simple PWE. It is however dubious whether the small difference was really worth the trouble. It is quite possible that training a single PWE on the *combination* of corpus data and the examples from the GLDB would have lead to a result almost as good, and with less work.

4 Conclusion

It seems we can conclude that an approach to word sense disambiguation based on Transformation-Based Learning is competitive with approaches based on Memory-Based Learning as used by the Göteborg team, and support vector machine (SVM) learning, used by the University of Maryland team. This is good news for those aiming at building NLP systems in which transformation rules play a major role.

As we have seen, there is meaning in the life of Peewees, and sound mathematical meaning at that! Also, given the link between first order logic and a logic programming language such as Prolog, the implementation follows very directly from the specification. The existence of a compiler from Peewee specifications into Prolog procedures makes Peewees very convenient to work with in a Prolog environment.

[7]Note that the coarse-grained evaluation was not applicable to the Swedish task. Also, it should be noted that our results in the first round of evaluation were slightly worse than the results reported here. However, this was due to a spelling error which could be corrected by the conference organizers and thus did not involve any resubmission of test results.

5 Acknowledgements

We thank the SENSEVAL-2 organizers for making all this possible, and in particular Jerker Järborg and Dimitrios Kokkinakis in Göteborg for their work on preparing for the Swedish lexical sample task.

References

Berleant, D. (1995) Engineering "Word Experts" for Word Disambiguation. *Natural Language Engineering*, 1(4).

Brill, E. (1995) Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part of Speech Tagging. *Computational Linguistics* 21.

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

Lager, T. (1999) The μ-TBL System: Logic Programming Tools for Transformation-Based Learning. In *Proceedings of CoNLL'99*, Bergen, Norway.

Lager, T. (2000) A Logic Programming Approach to Word Expert Engineering. In *Proceedings of ACIDCA 2000: Workshop on Corpora and Natural Language Processing*, Monastir, Tunisia, March 22-24 2000.

Lager, T. and Nivre, J. (2001) Part of Speech Tagging from a Logical Point of View. In de Groote, P., Morrill, G., Retor, C. (eds.) *Logical Aspects of Computational Linguistics.* Springer-Verlag, LNAI. VOL. 2099.

Pereira, F. and Warren, D. H. D. (1983) Parsing as Deduction, In *Proceedings of the 21th Meeting of the ACL.*

Primitive-Based Word Sense Disambiguation For SENSEVAL-2

Lim Beng Tat, Prof Zaharin Yusoff, Dr Tang Enya Kong and **Dr Guo Cheng Ming**
Unit Terjemahan Melalui Komputer,
Universiti Sains Malaysia,
11800, Pulau Pinang.
{btlim, zarin, enyakong, cmguo}@cs.usm.my

Abstract

This paper describes a descriptive-semantic-primitive-based method for word sense disambiguation (WSD) with a machine-tractable dictionary and conceptual distance data among primitives. This approach is using unsupervised learning algorithm and focuses only on the immediately surrounding words and basis morphological form to disambiguate a word sense. This approach also agrees with past observations that human only requires a small window of a few words to perform WSD. (Choueka & Lusignan, 1985). In additional, this paper also describes our experience in doing the English all-word task in SENSEVAL-2. Then, we will discuss the results in the SENSEVAL-2 evaluation.

Apart from the description of current system, possibilities for future work are explored

1 Primitive-Based Word Sense Disambiguation

This system consists of three important components: machine-tractable dictionary, conceptual distance data and sense tagger that uses a simple summation algorithm.

1.1 Machine-Tractable Dictionary

The first one is Machine-Tractable Dictionary (MTD) such as WordNet and LDOCE (Longman Dictionary of Contemporary English) especially LDOCE has been used extensively in NLP research and provide a broad set of senses for sense tagging. MTD contains word senses and their definitions are defined in term of descriptive and tagged primitives (words attached with sense number). Primitives are a set of words derived from dictionary (Guo, 1989b) and it is used to define the definition of a word sense. (For further information about primitives, please refer to Wilks.Y (1977)). For example, father#1 has a definition defined by using four primitives that are '*title1*','*respect2*', '*priest3*' and '*church4*' (refer figure 1).

For SENSEVAL-2 competition, the pre-release WordNet1.7 was used for this purpose. After WordNet1.7 was downloaded, the entries including their definition, sense number and sense id in WordNet was extracted and written into a temporary file. Primitives (not tagged) were derived from the words used in the word senses' definition. Then, the first 7 words of the definition text of the WordNet dictionary were disambiguated using the information from an existing MTD (LDOCE) and the derived primitives (Guo, 1998a). The existing MTD (LDOCE) contained word senses and the words in their definition are already tagged.

Thus, a new MTD (the pre-release WordNet 1.7) was ready for the usage of tagging process.

1.2 Conceptual Distance Data

Conceptual distance data is showing the relatedness between two tagged primitives.

Basically, the conceptual distance data is calculated by using content terms in the definition to determine the relatedness measure between two primitives layer by layer. The definitions of the primitives are getting from the existing MTD (LDOCE). It is important to note that a tagged primitive is also a word sense. For example, the first and second layer of referential definition for word sense '*forecast2*' is:

forecast2 [def] predict1 in2 advance3
predict1 [def] make1 a2 prediction3 about4; tell1 in2 advance3
1-referential layer: forecast2 predict1 advance3
2-referential layer: predict1 make1 prediction3 tell1
(note:'advance3' is omitted because it has been counted in the first layer)

Formula used to compute the relatedness percentage:

% for first layer of the first target word sense and first layer of the second target word sense:

if q<(n1+n2)/2 then p1 = (q/((n1+n2)/2)*70%

if q>(n1+n2)/2 then p1 = 70%

% for other layers:

x1 = q1 / ((n3+n4)/2)

x2 = q2 / ((n1+n4)/2)

x3 = q3 / ((n2+n3)/2)

p2=((x1+x2+x3)/3)*30%

The total value = p1+p2

n1 = no. of the element in the first layer of first target word sense

n2 = no. of the element in the first layer of second target word sense

n3 = no. of the element in the second layer of first target word sense

n4 = no. of the element in the second layer of second target word sense

q, q1, q2, q3 = no. of common content terms for each comparison

p1, p2 = final value of the relatedness measure

1.3 Sense Tagger

The third one is sense tagger. Sense tagger will get the input from MTD and its conceptual distance data among primitives to do the word sense disambiguation. Currently, the tagger consists of three processes:

- Preprocess process.
- Dictionary look-up process
- Numerical calculation algorithm

In the preprocess process, test data, which is downloaded for the usage of SENSEVAL-2, is going through several processes before tagging process takes place. The first process is separating the given text into sentences using full stops as separator. After that, the words in the sentences that do not require tagging will be removed, leaving only the heads (words to be sense-tagged) behind. Then, each word in the sentences will be stemmed, leaving only morphological root. The list of heads is then cut into chunks of three successive heads to be tagged in seconds.

In dictionary look-up module, word senses with their definition for each of the words in a chunk is extracted from MTD.

After that, sense tagger will use numerical calculation algorithm to choose the suitable word sense for the words in the sentence. This algorithm is to compute the path value among the definition of the word senses in a sentence. This is done first, by summing up the semantic data from conceptual distance data when comparison among primitives in the definitions for the word sense pairs in the sentence. After that, the result of summation has to be multiplied with the distance value between the two words in the sentence. This distance value basically depends on total words in a sentence. For example, sentence "Father marry couple", the distance value for '*father*' and '*marry*' is 2 whereas the distance value for '*father*' and '*couple*' is 1. This is because word '*father*' is closer to the word '*marry*' than '*couple*'. Then this computation continues for the other of word sense pairs.

Finally, this algorithm will compare the path values for the combination of word senses in a sentence and find the highest path value. Then this algorithm assigns the best combination of senses to each word in the sentence.

For example, with reference to Figure 1, assume that words such as '*father*', '*marry*' and '*couple*' have two senses only.

In the first step, definition of sense 1 from '*father*' will compare with definition of sense 1 from '*marry*'.

father 1	marry 1	value
title1	take1	x1
	person2	x2
	marriage3	x3
respect2	take1	x4
	person2	x5
	marriage3	x6
priest3	take1	x7
	person2	x8
	marriage3	x9
church4	take1	x10
	person2	x11
	marriage3	x12
Total		x1+x2+...x12=X
Total comparison		12

(Note: x1, x2, ...x12 are the values accessed from conceptual data.)

In the second step, definition of sense 1 from '*father*' will compare with definition of sense 1 from '*couple*'. The total comparison is 4*4=16 and total value extracted from conceptual distance data is Y. Then in the third step, definition of sense 1 from '*couple*' will compare with definition of sense 1 from '*marry*'. The total comparison is 4*3=12 and total value extracted from conceptual distance data is Z. The calculations for second step and third are as same as the step.

So, the path value for father1 marry1 couple1 = 2(X/12) +Y/16 + 2(Z/12).

Formula used to compute path value:

$$\text{Path value} = \sum_{i=1}^{n} (\text{distance})(s\_i \,/\, \text{total comparison})$$

where n = the total of words sense pairs , s = the total summation of values getting from conceptual distance data for *i*-th of word sense pairs.

This process will continue for other combination of word senses:

father1	marry1	couple2
father1	marry2	couple1
father1	marry2	couple2
father2	marry1	couple1
father2	marry1	couple2
father2	marry2	couple1
father2	marry2	couple2

The total combination of word sense for this example is 2*2*2=8. Finally, this algorithm will compare the path values for the combination of word senses in a sentence and find the most suitable combination of word senses. (Please refer to Figure 2)

2 Result

System	Number of primitives	Course Grained Precision/ Recall	Fine Grained Precision/ Recall
usm1	492	35.5% / 34.7%	34.5% / 33.8%
usm2	478	37.0% / 37.0%	36.0% / 36.0%
usm3	4000	34.4% / 34.4%	33.6% / 33.6%

Table 1: SENSEVAL-2 English All Word Results

(note:usm = Universiti Sains Malaysia)

With reference to the above table, usm1, usm2 and usm3 are three systems that are different in the number of primitives used in MTD as well as in the conceptual distance data and also MTD used. MTD used in usm1 is less comprehensive compare to MTD used in usm2 and usm3. More comprehensive is meaning that each of the entries is represented by a more complete set of primitives. MTD used in usm2 and usm3 is the same. Because of we are focusing more on speed of the system, overall of the results decreases when only the head words are considered.

3 Future extension of the system

In order to improve the existing algorithm, we need to avoid repeated calculation especially repeated comparison among the primitives. The concept of dynamic programming is needed to reduce the calculation. Basically, by using this method, result of the calculation is stored in memory so that the result can be accessed easily later when it is needed. As a result, although this method will increase the memory usage, it can also increase speed of the calculation significantly especially when a long sentence is processed. This is important because since the speed of the algorithm is increasing, it can be used in the real time application such information retrieval system especially in the Internet.

In additional, the accuracy of the system can be increased because more words in a sentence can be considered when a target word is tagged.

It is also important to note that this system not only can be used for English language, it can also be used in the other languages such Bahasa Malaysia, Chinese and Japanese.

Conclusion

In this paper, we have illustrated the overall architecture of our application of unsupervised learning technique to word sense disambiguation. Besides that, we have also presented that how our application in handling the given sentence and how we manage to complete the English all task given by SENSEVAL-2 competition. In additional, we illustrated the improvement over the algorithm we have presented in this paper. This is to make the algorithm becoming more efficient and practical to implement in real time application.

References

Guo, C-M (1989a) "Constructing a Machine-Tractable Dictionary from Longman Dictionary of Contemporary English.". Doctoral dissertation. New Mexico State University.

Guo, C-M (1989b) "Deriving a natural set of semantic primitives from Longman Dictionary of Contemporary English." Proceedings of the Second Irish Conference on Artificial Intelligence and Cognitive Science. 218-227

Wilks, Y. (1977) "Good and Bad Arguments About Semantic Primitives." In Communication and Cognition, Vol 10, No 3/4.

Y. Choueka and S.Lusigna. (1985). "Disambiguation by Short Contexts". Computer and the Humanities. 19:147-157.

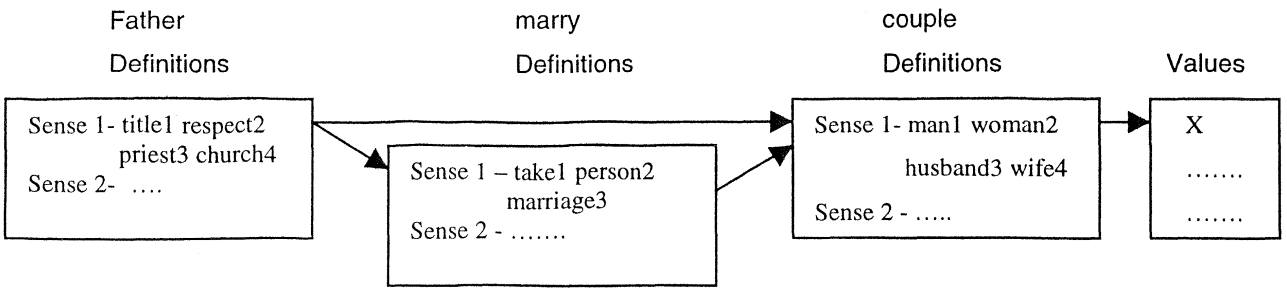

Figure 1: Example for sense-pair comparison among the words in a sentence.

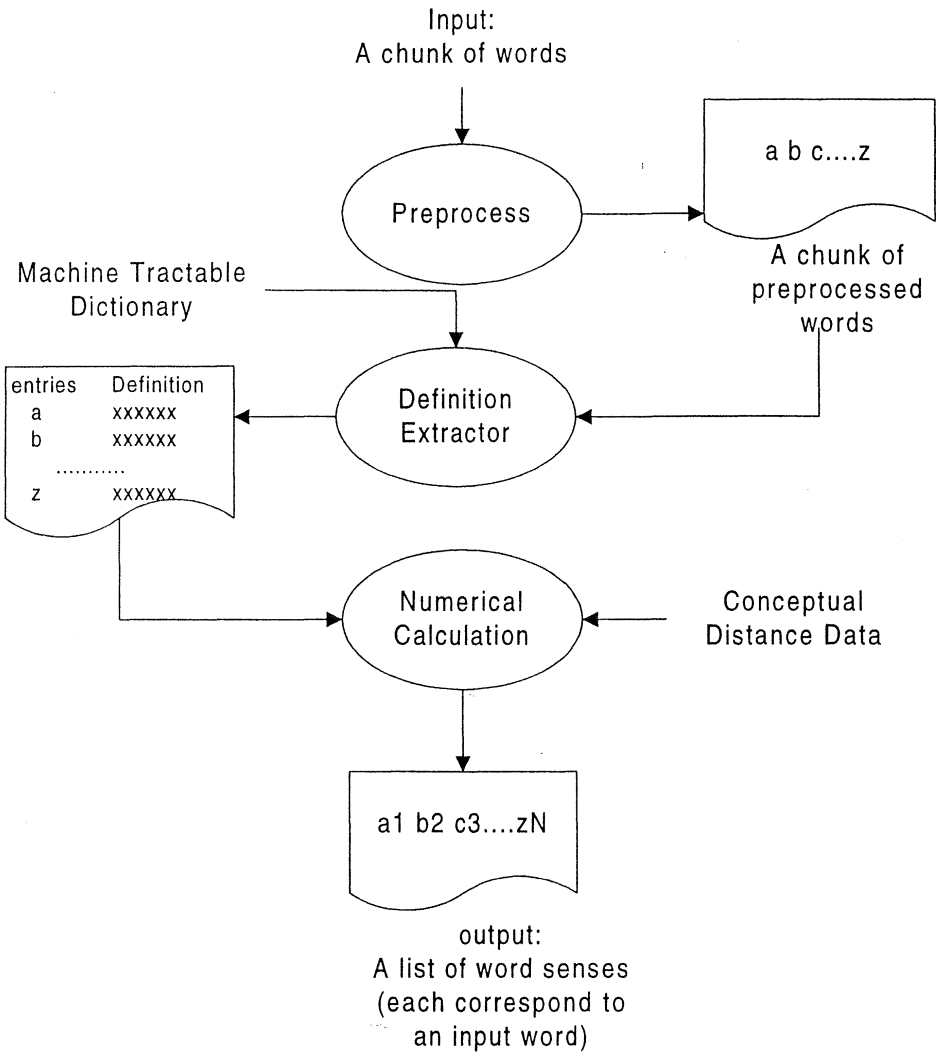

Figure 2: Sense Tagger

Use of Machine Readable Dictionaries for Word-Sense Disambiguation in SENSEVAL-2

Kenneth C. Litkowski
CL Research
9208 Gue Road
Damascus, MD 20872
ken@clres.com

Abstract

CL Research's word-sense disambiguation (WSD) system is part of the DIMAP dictionary software, designed to use any full dictionary as the basis for unsupervised disambiguation. Official SENSEVAL-2 results were generated using WordNet, and separately using the New Oxford Dictionary of English (NODE). The disambiguation functionality exploits whatever information is made available by the lexical database. Special routines examined multiword units and contextual clues (both collocations, definition and example content words, and subject matter analyses); syntactic constraints have not yet been employed. The official coarse-grained precision was 0.367 for the lexical sample task and 0.460 for the all-words task (these are actually recall, with actual precision of 0.390 and 0.506 for the two tasks). NODE definitions were automatically mapped into WordNet, with precision of 0.405 and 0.418 on 75% and 70% mapping for the lexical sample and all-words tasks, respectively, comparable to WordNet. Bug fixes and implementation of incomplete routines have increased the precision for the lexical sample to 0.429 (with many improvements still likely).

Introduction

CL Research's participation in SENSEVAL-2 was designed to (1) extend WSD techniques from SENSEVAL-1 (Litkowski, 2000), (2) generalize WSD mechanisms to rely on a full dictionary rather than a small set of entries where individual crafting might intrude, and (3) investigate WSD using one dictionary mapped into another (WordNet). Results indicate positive achievements for each of these goals. Time constraints precluded a complete assessment of the upper limits that can be achieved. In particular, although the general architecture from SENSEVAL-1 was retained, several specific WSD routines were not reimplemented. Incomplete testing, debugging, and implementation of new routines significantly affected the official results. Several of these problems are investigated more fully below.

CL Research's WSD functionality is implemented in DIMAP[1], designed primarily for creation and maintenance of lexicons for natural language processing. In particular, DIMAP is designed to make machine-readable dictionaries (MRDs) tractable and to create semantic networks (similar to WordNet (Fellbaum, 1998) and MindNet (Richardson, 1997)) automatically by analyzing and parsing definitions. Section 1 describes the dictionary preparation techniques for WordNet and NODE (The New Oxford Dictionary of English, 1998), as well as the mapping from NODE to WordNet. Section 2 describes the WSD techniques used in SENSEVAL-2. Section 3 describes the SENSEVAL-2 results and section 4 discusses these results..

1 Dictionary Preparation

DIMAP can disambiguate any text against WordNet or any other dictionary converted to DIMAP, with a special emphasis on corpus instances for specific lemmas. The dictionaries used for disambiguation operate in the background (as distinguished from the foreground development and maintenance of a dictionary), with rapid btree lookup to access and examine the characteristics of multiple senses of a word after a sentence has been parsed. DIMAP allows multiple senses for each entry, with fields for the definitions, usage notes, hypernyms, hyponyms,

[1] DIctionary MAintenance Programs, available from CL Research at http://www.clres.com.

arbitrary other semantic relations, and feature structures containing arbitrary information.

WordNet is already integrated in DIMAP in several ways, but for SENSEVAL-2, WordNet was entirely converted to alphabetic format for use as the disambiguation dictionary. In this conversion, all WordNet information (e.g., verb frames and glosses) and relations are retained. Glosses are analyzed into definition, examples, usage or subject labels, and usage notes (e.g., "used with 'of'"). Verb frames are used to build collocation patterns, typical subjects and objects, and grammatical characterizations (e.g., transitivity). WordNet file and sense numbers are converted into a unique identifier for each sense.

A separate "phrase" dictionary was constructed from all noun and verb multiword units (MWUs), using WordNet's sense index file. For nouns, an entry was created for the last word (i.e., the head), with the first word(s) acting as a "hynonymic" indicator; an entry was also created for the first word, with the following word(s) acting as a collocation pattern (e.g., "work of art" is a hyponym of *art* and a collocation pattern under *work*, written "~ **of art**"). For verbs, an entry was created for the first word, with a collocation pattern (e.g., "keep an eye on" is entered as a collocation pattern "~ **an eye on**" under *keep*). In disambiguation, this dictionary was examined first for a match, with the full phrase then used to identify the sense inventory rather than a single word.

NODE was prepared in a similar manner, with several additions. A conversion program transformed the MRD files into various fields in DIMAP, the notable difference being the much richer and more formal structure (e.g., lexical preferences, grammar fields, and subsensing). Conversion also considerably expanded the number of entries by making headwords of all variant forms (fully duplicating the other lexical information of the root form) and phrases run on to single lemma entries. E.g., "**(as) happy as a sandboy (or Larry or a clam**" under *happy* was converted into six headwords (based on the alternatives indicated by the parentheses), as well as a collocation pattern for a sense under *happy*, written "(as|?) ~ as (a sandboy | Larry | a clam)", with the tilde marking the target word.

NODE was then subjected to definition processing and parsing. Definition processing consists of further expansion of the print dictionary: (1) grabbing the definitions of cross-references and (2) assigning parts of speech to phrases based on analysis of their definitions. Definition parsing puts the definition into a sentence frame appropriate to the part of speech, making use of typical subjects, objects, and modificands. The sentence parse tree was then analyzed to extract various semantic relations, including the superordinate or hypernym, holonyms, meronyms, satellites, telic roles, and frame elements. After parsing was completed, a phrase dictionary was also created for NODE.[2]

The SENSEVAL tasks were run separately against the WordNet and NODE sense inventories, with the WordNet results submitted. To investigate the viability of mapping for WSD, subdictionaries were created for each of the lexical sample words and for each of the all-words texts. For the lexical sample words, the subdictionaries consisted of the main word and all entries identifiable from the phrase dictionary for that word. (For *bar*, in NODE, there were 13 entries where **bar** was the first word in an MWU and 50 entries where it was the head noun; for *begin*, there was only one entry.) For the all-words texts, a list was made of all the task words to be disambiguated (including some phrases) and a subdictionary constructed from this list. For both tasks, the creation of these subdictionaries was fully automatic; no hand manipulation was involved.

The NODE dictionaries were then mapped into the WordNet dictionaries (see Litkowski, 1999), using overlap among words and semantic relations. The 73 dictionaries for the lexical sample words gave rise to 1372 WordNet entries and 1722 NODE entries.[3] Only 491 entries were common (i.e., no mappings were available for the remaining 1231 NODE entries); 881 entries in WordNet were therefore inaccessible through NODE. For the entries in

[2]WordNet definitions were not parsed. In an experiment, the semantic relations identifiable through parsing were frequently inconsistent with those already given in WordNet, so it was decided not to confound the disambiguation.

[3]Entries included all parts of speech; disambiguation was required to identify the part of speech as well.

common, there was an average of 5.6 senses, of which only 64% were mappable into WordNet. The *a priori* probability of successful mapping into the appropriate WordNet sense is 0.064, the baseline for assessing WSD via another dictionary mapped into the WordNet sense-tagged keys.[4]

2 Disambiguation Techniques

The lexical sample and all-words texts were modified slightly. Satellite tags were removed and entity references were converted to an ASCII character. In the all-words texts, contraction and quotation mark discontinuities were undone. These changes made the texts more like normal text processing conditions.

The texts were next reduced to sentences. For the lexical sample, a sentence was assumed to consist of a single line. For the all-words texts, a sentence splitter identified the sentences, which were next submitted to the parser. The DIMAP parser produced a parse tree for each sentence, with constituent phrases when the sentence was not parsable with the grammar, allowing the WSD phase to continue.

The first step in the WSD used the part of speech of the tagged word to select the appropriate sense inventory. Nouns, verbs, and adjectives were looked up in the phrase dictionary; if the tagged word was part of an MWU, the word was changed to the MWU and the MWU's sense inventory was used instead.

The dictionary entry for the word was then accessed. Before evaluating the senses, the topic area of the context provided by the sentence was "established" (only for NODE). Subject labels for all senses of all content words in the context were tallied.

Each sense of the target was then evaluated. Senses in a different part of speech were dropped from consideration. The different pieces of information in the sense were assessed: collocation patterns, contextual clue words, contextual overlap with definitions and examples, and topical area matches. Points were given to each sense and the sense with the highest score was selected; in case of a tie, the first sense in the dictionary was selected.[5]

Collocation pattern testing (requiring an exact match with surrounding text) was given the largest number of points (10), sufficient in general to dominate sense selection. Contextual clue words (a particle or preposition) was given a small score (2 points). Each content word of the context added two points if present in the sense's definition or examples, so that considerable overlap could become quite significant. For topic testing, a sense having a subject label matching one of the context topic areas was awarded one point for each word in the context that had a similar subject label (e.g., if four words in the context had a medical subject label, four points would be awarded if the instant sense also had a medical label).

3 Results

As shown in Table 1, using WordNet as the disambiguation dictionary resulted in an overall precision (and recall) of 0.293 at the fine-grained level and 0.367 at the coarse-grained level. Since CL Research did not use the training data in any way, running the training data also provided another test of the system. The results are remarkably consistent, both overall and for each part of speech. Using NODE as the disambiguation dictionary and mapping its senses into WordNet senses achieved comparable levels of precision, although recall was somewhat lower, as indicated by the difference in the number of items on which the precision was calculated. Overall, about 75% of the senses were mapped into WordNet.

Table 2. All-Words

Run	Items	Fine	Coarse
WordNet	2473	0.451	0.460
NODE	1727	0.416	0.418

For the all-words task, the disambiguation results

[4]Note that a mapping from WordNet to NODE is likely to generate similar mismatch statistics.

[5]Several other functions were implemented only in stub form at the time of the test runs, to evaluate: type restrictions (e.g., transitivity), presence of accompanying grammatical constituents (e.g., infinitive phrase or complements), form restrictions (such as number and participial), grammatical role (e.g., as a modifier), and selectional restrictions (such as subject, object, modificand, and internal arguments).

Table 1. Lexical Sample Precision												
	Adjectives			Nouns			Verbs			Total		
Run	Items	Fine	Coarse	Items	Fine	Coarse	Items	Fine	Coarse	Items	Fine	Coarse
WordNet Test	768	0.354	0.354	1726	0.338	0.439	1834	0.225	0.305	4328	0.293	0.367
NODE Test	420	0.288	0.288	1403	0.402	0.539	1394	0.219	0.305	3217	0.308	0.405
WordNet Training	1533	0.365	0.365	3455	0.334	0.444	3623	0.219	0.299	8611	0.291	0.369
NODE Training	864	0.116	0.116	2848	0.366	0.483	2567	0.227	0.315	6249	0.276	0.365

were significantly higher than for the lexical sample, with a precision (and recall) of 0.460 for the WordNet coarse-grained level. For NODE, about 70% were mapped into WordNet (indicated by the reduced number of items), with precision on the mapped items only slightly less.[6]

4 Discussion

Because of the usual bugs and incomplete implementation, the official results do not adequately indicate the potential of our approach. The official results are actually recall rather than precision, since an answer was submitted when it shouldn't have been, as distinguished from cases where the parser picked the wrong part of speech or was unable to select a sense. The actual precision for the lexical sample task is 0.311 for the fine grain and 0.390 for the coarse grain, and for the all-words task, 0.496 and 0.506 for fine and coarse grains, respectively.

Minimal debugging and inability to implement several routines significantly affected the scores. Examining the reasons for failures in the test runs and making program fixes has thus far resulted in increasing precision (and recall) to 0.340 and 0.429 for the lexical sample. Further improvements are likely, although it is not clear whether the SENSEVAL-1 precision of 0.67 is achievable using only the information available in WordNet.

It is more likely that using NODE will achieve better results. Improvements in automatic mapping have now reached 90% mapping; it is also relatively easy to make manual adjustments to the maps to achieve even higher performance from the lexicographically-based lexical resource. Since the automatic mapping is inaccurate to an unknown degree (perhaps 25-30%), improving the maps will achieve better results using NODE via WordNet, rather than WordNet alone. Using NODE also provides a much richer set of data upon which to make improvements in WSD. Finally, since NODE is lexicographically-based and with an arguably better sense inventory, we are confident that our WSD would have scored much higher if the taggers had used this inventory.

[6]For both tasks, NODE senses were identified for all words, but could be mapped only for the percentages given.

Conclusion

Given the very preliminary implementation of the disambiguation routines and lack of adequate debugging, the results indicate that using MRDs (and even mapping from one into another) shows considerable potential for unsupervised and general word-sense disambiguation.

Acknowledgements

I wish to thank Oxford University Press for making NODE available (Patrick Hanks and Rob Scriven) and for many useful discussions (Glynnis Chantrell and Judy Pearsall).

References

Fellbaum, C. (1998). *WordNet: An electronic lexical database.* Cambridge, Massachusetts: MIT Press.

Litkowski, K. C. (1999, 21-22 June). Towards a Meaning-Full Comparison of Lexical Resources. Association for Computational Linguistics Special Interest Group on the Lexicon Workshop. College Park, MD.

Litkowski, K. C. (2000). SENSEVAL: The CL Research Experience. *Computers and the Humanities, 34*(1-2), 153-158.

The New Oxford Dictionary of English (J. Pearsall, Ed.). (1998). Oxford: Clarendon Press.

Richardson, S. D. (1997). Determining similarity and inferring relations in a lexical knowledge base [Diss], New York, NY: The City University of New York.

Using Domain Information for Word Sense Disambiguation

Bernardo Magnini, Carlo Strapparava, Giovanni Pezzulo and **Alfio Gliozzo**
ITC-irst, Istituto per la Ricerca Scientifica e Tecnologica, I-38050 Trento, ITALY
email: {magnini, strappa, pezzulo, gliozzo}@itc.it

Abstract

The major goal in ITC-irst's participation at SENSEVAL-2 was to test the role of domain information in word sense disambiguation. The underlying working hypothesis is that domain labels, such as MEDICINE, ARCHITECTURE and SPORT provide a natural way to establish semantic relations among word senses, which can be profitably used during the disambiguation process. For each task in which we participated (i.e. English all words, English 'lexical sample' and Italian 'lexical sample') a different mix of knowledge based and statistical techniques were implemented.

1 Introduction

Current investigation in Word Sense Disambiguation (WSD) at ITC-irst focuses on the role of *domain information*. The hypothesis is that domain labels (such as MEDICINE, ARCHITECTURE and SPORT) provide a natural and powerful way to establish semantic relations among word senses, which can be profitably used during the disambiguation process. In particular, domains constitute a fundamental feature of text coherence, such that word senses occurring in a coherent portion of text tend to maximize domain similarity. The importance of domain information in WSD has been remarked in several works, including (Gonzalo et al., 1998) and (Buitelaar and Sacaleanu, 2001). In (Magnini and Strapparava, 2000) we introduced "Word Domain Disambiguation" (WDD) as a variant of WSD where for each word in a text a *domain* label (among those allowed by the word) has to be chosen instead of a *sense* label. We also argued that WDD can be applied to disambiguation tasks that do not require fine grained sense distinctions, such as information retrieval and content-based user modeling. For SENSEVAL-2 the goal was to evaluate the role of domain information in WSD: no other syntactic or semantic information has been used (e.g. semantic relations in WORDNET) except domain labels. Three systems have been implemented, integrating knowledge-based and statistical techniques, for the three tasks we participated in, i.e. English 'all words', English 'lexical sample' and Italian 'lexical sample'. The main lexical resource for domains is "WordNet Domains", an extension of English Wordnet 1.6 (Fellbaum, 1998) developed at ITC-irst, where synsets have been annotated with domain information.

2 WordNet Domains

The basic lexical resource we used in SENSEVAL-2 is "WordNet Domains", an extension of WORDNET 1.6 where each synset has been annotated with at least one domain label, selected from a set of about two hundred labels hierarchically organized (see (Magnini and Cavaglià, 2000) for the annotation methodology and for the evaluation of the resource). The information from the domains that we added is complementary to what is already in WORDNET. First of all a domain may include synsets of different syntactic categories: for instance MEDICINE groups together senses from Nouns, such as `doctor#1` and `hospital#1`, and from Verbs such as `operate#7`. Second, a domain may include senses from different WORDNET sub-hierarchies (i.e. deriving from different "unique beginners" or from different "lexicographer files"). For example, SPORT contains senses such as `athlete#1`, deriving from `life_form#1`, `game_equipment#1` from `physical_object#1`, `sport#1` from `act#2`, and `playing_field#1` from `location#1`. Finally, domains may group senses of the same word into homogeneous clusters, with the side

effect of reducing word polysemy in WORDNET. Table 1 shows an example. The word "bank" has ten different senses in WORDNET 1.6: three of them (i.e. sense 1, 3 and 6) can be grouped under the ECONOMY domain, while sense 2 and 7 both belong to GEOGRAPHY and GEOLOGY, causing the reduction of the polysemy from 10 to 7 senses. For the purposes of SENSEVAL-2 we have considered 41 disjoint labels which allow a good level of abstraction without loosing relevant information (i.e. in the experiments we have used SPORT in place of VOLLEY or BASKETBALL, which are subsumed by SPORT).

Sense	*Synset & Gloss*	*Domains*	*Semcor occurr.*
#1	depository financial institution, bank, banking concern, banking company (a financial institution...)	ECONOMY	20
#2	bank (sloping land...)	GEOGRAPHY, GEOLOGY	14
#3	bank (a supply or stock held in reserve...)	ECONOMY	-
#4	bank, bank building (a building...)	ARCHITECTURE, ECONOMY	-
#5	bank (an arrangement of similar objects...)	FACTOTUM	1
#6	savings bank, coin bank, money box, bank (a container...)	ECONOMY	-
#7	bank (a long ridge or pile...)	GEOGRAPHY, GEOLOGY	2
#8	bank (the funds held by a gambling house...)	ECONOMY, PLAY	
#9	bank, cant, camber (a slope in the turn of a road...)	ARCHITECTURE	-
#10	bank (a flight maneuver...)	TRANSPORT	-

Table 1: WORDNET senses, domains and occurrences in Semcor for the word "bank"

Two mapping procedures have been implemented for SENSEVAL-2 in order to use domain information. For the English tasks a mapping from WORDNET 1.6 to the WORDNET 1.7 pre-release made available to participants; for the Italian task a mapping from WORDNET 1.6 to WORDNET 1.5, because the interlingual index of EuroWordNet (Vossen, 1998) is in that version. The mapping to WORDNET 1.7 is based on a set of heuristics (e.g. correspondences between synonyms, glosses and hypernyms) which discover corresponding synset pairs. Then, an inheritance algorithm is applied to WORDNET 1.7 in order to fill unassigned synsets with domain labels. As far as the Italian wordnet is concerned the same procedure used for the WORDNET 1.7 mapping has been applied to WORDNET 1.5, resulting in the annotation of the Interlingual Index. Then the equivalence links (we excluded eq_hyperonym and eq_hyphonym) from the ILI to the Italian synsets were used to bring the domain information to Italian words.

There was no time for a complete evaluation of the quality of the mapping procedures.

3 Algorithms

The starting point in the algorithm design was the previous work in word domain disambiguation reported in (Magnini and Strapparava, 2000). One drawback of that approach is that, for rather long texts, it does not consider domain variations. To overcome this problem we have introduced *contexts* within which domains are calculated. A second direction of work has been the acquisition of domain information from annotated texts (i.e. Semcor and the training data). The following sections presents details of the disambiguation procedures implemented for SENSEVAL-2.

3.1 Linguistic Processing

XML files made available by the task organizers have been processed with an XML parser. As for lemmatization and part-of-speech tagging the Tree Tagger, developed at the University of Stuttgart (Schmid, 1994) has been used, both for English and Italian. The WordNet morphological analyser has also been used in order to resolve ambiguities and lemmatization mistakes. After this process texts are represented as vectors of triples: word lemma, WORDNET part of speech and position in the text.

3.2 Scoring Domains for a Lemma

The basic procedure in domain driven disambiguation is a function that, given a lemma L,

associates a score to each domain defined for that lemma in Wordnet Domains. Such a score is the relative frequency of the domain in L, computed on the basis of the occurrences of the synsets of L in Semcor. Semcor occurrences for synsets with multiple domain annotations are repeated for each domain (e.g. if a synset has 2 occurrences and 2 labels it is counted as having 4 occurrences), while synsets with 0 occurrences are counted as 0.5. As an example, consider the lemma "bank" in Table 1. According to our scoring method, it has 57 total occurrences in Semcor. The GEOLOGY domain collects contributions from senses 2 and 7, for a total of 16 occurrences in Semcor, which corresponds to a frequency .28 (i.e. $fq[D_{Geology}](bank) = 0.28$).

3.3 Domain Vectors

The data structure that collects domain information is called a *Domain Vector* (DV). Intuitively a DV represents the domains that are relevant for a certain lemma (or word sense) in a certain context. We have considered three kinds of DV's: a DV for a lemma L within a context C (DV_L^C), for the case of test data; a DV for a synset S of a lemma L within a context C (DV_S^C), for the case of training data; and a DV for a synset S of a lemma L in WORDNET (DV_S), which is used when no training data are available.

DV for a lemma in context (DV_L^C). Given a set of domains $D_1 \ldots D_n$, a DV for a lemma L in a position K within a text represents the relevance of those domains for that lemma, i.e. each component $DV_L[i]$ gives the degree of relevance of the domain D_i for the lemma L. Given a context of $\pm C$ words before and after the lemma L in the position K, each component of the domain vector is defined with the following formula:

$$DV_L^C[i] = \sum_{k=-C}^{+C} Fq[D_i](L_k) * gauss$$

where *gauss* is the normal distribution centered on the position K. In the current algorithms C is set to 50 because our experiments with Semcor showed that the precision decreases below that thresold.

Intuitively, the above formula takes into account the contribution of the lemmas in the context C to the sense of the target lemma L. In addition a DV actually selects a set of relevant domains rather than just one domain.

DV for a synset in context (DV_S^C) In case a training corpus is available where lemmas are annotated with the correct sense, Domain Vectors are computed with the formula above. Instead of considering a lemma in a position K within a text, we have a sense for that lemma (i.e. a synset). DV_S^C represents a "typical" vector for a sense S of a lemma L.

DV for a synset without context (DV_S) When a training corpus is not available (as for the 'all words' task), a simpler way to build a DV for a certain synset is to compute it with respect to WordNet Domains. Given a synset S in WordNet Domains, the domain vector DV_S is a vector that has 1's in the position of its domain(s) and 0's otherwise. A more accurate DV could be obtained by considering contextual information such as the synset gloss.

3.4 Comparing Domain Vectors

To disambiguate a lemma L (i.e. the target lemma) in a text, first its DV_L^C is computed. The next step consists of comparing the DV of the target lemma L with the domain vectors for each sense of L derived either from the training set, when available, or from WordNet Domains, when training data are not available. The sense vector DV_S which maximizes the similarity is selected as the appropriate sense of L in that text. The similarity between two DV's is calculated with the standard scalar product: $DV_1 \cdot DV_2 = \sum_i DV_1[i] * DV_1[i]$.

4 Results and Discussion

Table 2 presents the results, in terms of precision and recall, obtained at the SENSEVAL-2 initiative for the three tasks in which we participated.

Task	*Precision*	*Recall*
English All Words (fine g.)	.748	.357
English All Words (coarse g.)	.748	.357
English Lexical Sample (fine g.)	.665	.249
English Lexical Sample (coarse g.)	.720	.269
Italian Lexical Sample (fine g.)	.375	.371

Table 2: Final results of ITC-irst systems at SENSEVAL-2

4.1 English 'All Words'

The 'all words' task seems to benefit from the domain approach. One reason for this is that texts are enough long to provide an accurate context (as mentioned in section 3.3, we used a window of 100 content words around the target word) within which domains are coherent. The rather low degree of recall reflects the fact that few words in a text carry relevant domain information. Most of the words actually behave such as a "factotum" (see (Magnini and Cavaglià, 2000) for a preliminary discussion on this problem) that can equally occur in almost every domain. Some words lie outside the domain approach and their senses could be captured with the integration of local (e.g. syntactic) information.

4.2 English 'Lexical Sample'

From the point of view of domain driven disambiguation, the 'lexical sample' task was inherently more difficult than the 'all words' task for two reasons. First the context provided for disambiguation was generally shorter than the 100 words we used to build a semantic vector. Second, the high number of "factotum" words to be disambiguated resulted in a recall even lower (i.e. about 0.24) than for the 'all words' task. The improvement of performance from the fine grained to the coarse grained evaluation seems to confirm that, at least to some degree, domain clustering corresponds to the sense grouping created by the task organizers.

4.3 Italian 'Lexical Sample'

The low results obtained for the Italian 'lexical sample' task may have several causes. First of all, the absence of a training set and the absence of any tagged text for Italian forced us to use a similarity function (see 3.4) trained to an English corpus. This was possible because we maintained the mappings between the English and the Italian wordnets. However, these multiple mappings (i.e. from WORDNET1.6 to WORDNET1.5 and then to the Italian synsets through the equivalence links) are another source of possible errors, especially concerning the domain information associated with Italian synsets.

5 Conclusions

We have described an approach to word sense disambiguation based on domain information. The underlying assumption is that domains constitute a fundamental feature of text coherence. As a consequence, word senses occurring in a coherent portion of text tend to maximize domain similarity. Three systems have been implemented, integrating knowledge-based and statistical techniques, for the three task we participated in. As for lexical resources, the systems make use of WordNet Domains, an extension of English Wordnet 1.6, where synsets have been annotated with domain information. The disambiguation algorithm is based on domain vectors that collect contextual information with respect to the target word. At this moment only domain information is used in our system. A promising research direction is the use of local information (e.g. syntax) to capture word behaviors that lie outside the domain approach.

References

P. Buitelaar and B. Sacaleanu. 2001. Ranking and selecting synsets by domain relevance. In *Proc. of NAACL Workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations*, Pittsburgh, PA, June.

C. Fellbaum. 1998. *WordNet. An Electronic Lexical Database*. The MIT Press.

J. Gonzalo, F. Verdejio, C. Peters, and N. Calzolari. 1998. Applying eurowordnet to cross-language text retrieval. *Computers and Humanities*, 32(2-3):185–207.

B. Magnini and G. Cavaglià. 2000. Integrating subject field codes into WordNet. In *Proceedings of LREC-2000, Second International Conference on Language Resources and Evaluation*, Athens, Greece, June.

B. Magnini and C. Strapparava. 2000. Experiments in word domain disambiguation for parallel texts. In *Proc. of SIGLEX Workshop on Word Senses and Multi-linguality*, Hong-Kong, October.

H. Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing.*

P. Vossen. 1998. Special issue on eurowordnet. *Computers and Humanities*, 32.

Decision Lists for English and Basque

David Martinez
IXA NLP Group
University of the Basque Country
649 pk. 20.080
Donostia. Spain.
jibmaird@si.ehu.es

Eneko Agirre
IXA NLP Group
University of the Basque Country
649 pk. 20.080
Donostia. Spain.
eneko@si.ehu.es

Abstract

In this paper we describe the systems we developed for the English (lexical and all-words) and Basque tasks. They were all supervised systems based on Yarowsky's Decision Lists. We used Semcor for training in the English all-words task. We defined different feature sets for each language. For Basque, in order to extract all the information from the text, we defined features that have not been used before in the literature, using a morphological analyzer. We also implemented systems that selected automatically good features and were able to obtain a prefixed precision (85%) at the cost of coverage. The systems that used all the features were identified as BCU-ehu-dlist-all and the systems that selected some features as BCU-ehu-dlist-best.

1 Introduction

Our group took part in three tasks in Senseval-2001: all-words and lexical sample for English, and lexical sample for Basque. We applied the same algorithm in all tasks, but using different feature sets. The method we used was based on Yarowsky's Decision Lists (Yarowsky, 1994).

We have to mention that different motivations were pursued when working for English or for Basque. In the last years, our work for English has focused on studying the contribution of different kinds of features to WSD (Agirre and Martinez, 2000; Agirre and Martinez, 2001a) and on analyzing different knowledge types on a common setting (Agirre and Martinez, 2001b).

The English tasks gave us the chance to compare the performance of our method with state-of-the-art systems. Unfortunately, due to time constrains, we could not train the system with syntactic and semantic features, as was our goal. The systems we used for the English tasks were trained with topical and local features already mentioned in the literature (Yarowsky, 1994).

In the English lexical sample, we presented two systems: one trained with all the features (BCU-ehu-dlist-all) and another that selected automatically good features and was able to obtain a prefixed precision (85%) at the cost of coverage (BCU-ehu-dlist-best). In the all-words exercise, we presented only one system, which used all the features and was trained using Semcor (previously we mapped automatically WN 1.6 senses with the WN 1.7 Pre-release). The features and the systems will be described in Section 3.

The Basque task presented interesting challenges for us. Previous work in WSD was performed using MRDs, but this was our first approach to the problem using Decision Lists. The Basque language has some particularities that make the selection of features a difficult task. First of all, Basque is an agglutinative language, and some syntactic information is given by inflectional suffixes. Therefore, it is necessary a powerful morphologic analysis of the text in order to identify the lemma and the different parts of each word. Also, phenomena like noun ellipsis have to be taken into account.

We had the tools to perform a deep morphological analysis of the text (Urkia, 1997) and we were able to define a richer feature set than the one used for English. The complex structure of the analysis allowed constructing different feature types, which should be studied in detail. However, our approach was to define many features and integrate them together in the system, expecting that the Decision List algorithm would be powerful enough to choose the best ones in each case. We will explain the feature set in Section 4. We also presented two systems for Basque: one using all the features

and another selecting good features (those above a threshold of 85% precision).

Following this introduction, we will briefly explain the Decision List algorithm in Section 2. Section 3 will be devoted to the English lexical task and Section 4 to the English all-words task. The Basque lexical task will be described in Section 5. The results obtained by our systems will be discussed in Section 6. Finally, we will resume our conclusions in Section 7.

2 Decision Lists

Decision lists as defined in (Yarowsky, 1994) are simple means to resolve ambiguity problems. With the addition of some hierarchical structure, it was one of the most successful systems on the Senseval-1 exercise. 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.

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

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.

3 English lexical-sample task

In the English lexical sample task, we presented two systems: BCU-ehu-dlist-all and BCU-ehu-dlist-best.

3.1. BCU-dlist-ehu-all

We trained our Decision List algorithm using local and global features:

- Local features: bigrams and trigrams around the target word, consisting on lemmas or word forms or parts of speech. Also a bag of lemmas constructed using the content words in a ±4 word window around the target.
- Global features: a bag of lemmas with the content words included in the whole context provided for the target word.

We did not use the tags P and U. There was no special treatment for multiword detection.

3.2. BCU-dlist-ehu-best

In this case, instead of using the whole set of features, only the best features for each word were chosen. Features that had a precision above a threshold of 85% were automatically selected running the system on the training data, using 10 fold cross validation. With this second system we wanted to guarantee high precision for each word, at the cost of coverage.

4 English all-words task (BCU-dlist-ehu-all)

In the all-words exercise, we presented only one system, which used all the features and was trained using Semcor. A mapping between the senses in Semcor (tagged with WordNet 1.6 senses) and WordNet 1.7 was performed automatically for nouns and verbs; only words with those parts-of-speech were treated (we could not finish the mapping for adjectives on time).

5 Basque lexical-sample task (BCU-dlist-ehu-all and BCU-dlist-ehu-best)

As mentioned before, Basque is an agglutinative language, and syntactic information is given by inflectional suffixes. The morphological analysis of the text is a necessary previous step in order to select informative features. In Basque, the determiner, the number and the declension case are appended to the last element of the phrase. In order to include this information in our representation, we have to use more rich features than those defined for English. When defining our feature set for Basque, we tried to introduce the same knowledge that is represented by features that work well for English.

We will describe our feature set with an example: for the phrase "**elizaren arduradunei**" (which means "to the directors of the church") we get the following analysis:

eliza	**|ren**	**|arduradun**	**|ei**
church	**|of the**	**|director**	**|to the +plural**

The order of the words is inverse in English. We extract the following information for each word:

elizaren:
- Lemma: eliza (church)
- PoS: noun
- Declension Case: genitive (of)
- Number: singular
- Determiner mark: yes

arduradunei:
- Lemma: arduradun (director)
- PoS: noun
- Declension Case: dative (to)
- Number: plural
- Determiner mark: yes

We will assume that *eliza* (*church*) is the target word. Words and lemmas are shown in lowercase and the other information in uppercase. As local features we defined different types of unigrams, bigrams, trigrams and a window of ±4 words. The unigrams were constructed combining word forms, lemmas, case, number, and determiner mark. We defined 4 kinds of unigrams:

```
Uni_wf0 elizaren
Uni_wf1 eliza  SING+DET
Uni_wf2 eliza  GENITIVE
Uni_wf3 eliza  SING+DET      GENITIVE
```

As for English, we defined bigrams based on word forms, lemmas and parts-of-speech. But in order to simulate the bigrams and trigrams used for English, we defined different kinds of features. For word forms, we distinguished two cases: using the text string (Big_wf0), or using the tags from the analysis (Big_wf1). The word form bigrams for the collocation "elizaren arduradunei" are shown below. In the case of the feature type "Big_wf1", the information is split in three features:

```
Big_wf0 elizaren arduradunei

Big_wf1 eliza GENITIVE
Big_wf1GENITIVE arduradun_PLUR+DET
Big_wf1 arduradun_PLUR+DET DATIVE
```

Similarly, depending on the use of the declension case, we defined three kinds of bigrams based on lemmas:

```
Big_lem0 eliza arduradun

Big_lem1 eliza GENITIVE
Big_lem1 GENITIVE arduradun
Big_lem1 arduradun DATIVE

Big_lem2 eliza_GENITIVE
Big_lem2 arduradun_DATIVE
```

The bigrams constructed using Part-of-speech are illustrated below. We included the declension case:

```
Big_pos_-1 NOUN GENITIVE
Big_pos_-1 GENITIVE NOUN
Big_pos_-1 NOUN DATIVE
```

Trigrams are built similarly, by combining the information from three consecutive words. We also used as local features all the content words in a window of ±4 words around the target. Finally, as global features we took all the content lemmas appearing in the context, which was constituted by the target sentence and the two previous and posterior sentences.

One difficult case to model in Basque is the ellipsis. For example, the word "elizakoa" means "the one from the church". We were able to extract this information from our analyzer and we represented it in the features, using a mark as the elliptic word.

We implemented two systems; in the first one, we integrated all the features in the Decision List algorithm, expecting that the most informative ones would be chosen. The performance of the different features was not studied separately. Our second system for Basque applied feature selection in a similar way as for English.

This was our first approach to represent the Basque sentences in a feature set suitable for the Decision List algorithm. We detected some reasons that could have lowered the performance of the system:

- In Basque the word order is free. The performance of bigrams and trigrams, which have to be in fixed positions, could be affected for this fact.
- When we introduce the number, declension case, and determiner; the relation between some words that are close in the text could be lost. We have tried to overcome this by defining many features, but we did not

analyze them by hand, and some could introduce noise. Deeper study of the features should be done in order to know the real performance of the method. Uncommon cases, like ellipsis, should be further examined.

- Another source of noise was the morphological analyzer, which in some cases produced very ambiguous analysis, or errors.

6 Results and Discussion

In the English lexical task, BCU-ehu-dlist-all scored 57.3% in precision and 98% in coverage. It beat easily the different baselines and with a simple implementation, was close in precision to more elaborate and complex systems. With BCU-ehu-dlist-best we were able to obtain a precision of 82.9% for 28% coverage. The threshold of 85% precision proved to be too high for some words, and too low for others. Besides, the chosen features had low coverage and could be applied only in a few cases.

In the English all-words task, we obtained almost the same precision as in the lexical task: 57.2%. The coverage was limited to nouns and verbs with training examples in Semcor, and reached 51% of the target words. Clearly, more training data was required to compete in recall with the best systems.

For Basque, with BCU-ehu-dlist-all we obtained 73.2% precision for 100% coverage. The system improved in almost 9 points the precision of the most frequent sense (MFS) baseline, but was two points below the best system (JHU- John Hopkins University). We have to notice that the JHU system won the lexical sample task both for Basque and for English; and while the difference in recall with our system was only 2% for Basque, it reached 8% for English. We think that the reason for this is that our feature set for Basque is better, although our ML algorithm is worse.

Finally, with BCU-ehu-dlist-best the 85% threshold worked better than for English and we reached higher coverage. We were able to obtain 84.9% precision for 57% coverage. However, again, the threshold was too high for some words (for 4 words no feature was chosen), and too low for others (easy words like "enplegu" chose the whole feature set).

The positions of our systems in the different tasks are illustrated in Table 1:

Task	System	Position	
		Precision	Recall
Basque lexical	best	1st of 3	3rd of 3
Basque lexical	all	3rd of 3	2nd of 3
English lexical	best	1st of 20	20th of 20
English lexical	all	9th of 20	9th of 20
English all-words	all	7th of 21	14th of 21

Table 1: Classification of our systems (*version 1.5, published 28 Sep. 2001).* Fine-grained scoring. Only last versions of resubmitted systems (R) are included. Baselines are not incorporated. Only supervised systems are included in the lexical tasks.

7 Conclusions

In the English tasks we were able to compare a limited version of our system (with a reduced feature set) with state-of-the-art systems. We observed that with minimum work, we could obtain results above the average of the other systems. Our next goal is to test the system with semantic and syntactic features and compare the performance with other systems.

For Basque, we defined a preliminary set of features and achieved good performance. Our results were close to the best system and above the MFS baseline. In the future, we want to refine the feature set and explore other sources of information, as syntactic features.

Finally, more experiments on feature selection should be performed in order to take advantage of this technique.

8 References

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. and Martínez D. *Learning class-to-class selectional preferences.* Proceedings of the Workshop CoNLL. In conjunction with ACL'2001. Toulouse, France. 2001.

Agirre E. and Martinez D. *Knowledge sources for Word Sense Disambiguation.* Proceedings of the Fourth International Conference TSD 2001, Plzen (Pilsen), Czech Republic. 2001.

Urkia M., *Euskal Morfologiaren tratamendu informatikorantz*, Gasteiz, UPV/EHU, Ph.D. thesis. 1997.

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.

Disambiguating Noun and Verb Senses Using Automatically Acquired Selectional Preferences*

Diana McCarthy and **John Carroll**
Cognitive & Computing Sciences
University of Sussex
Brighton BN1 9QH, UK
{dianam,johnca}@cogs.susx.ac.uk

Judita Preiss
Computer Laboratory
University of Cambridge, JJ Thomson Avenue
Cambridge CB3 0FD, UK
Judita.Preiss@cl.cam.ac.uk

Abstract

Our system for the SENSEVAL-2 all words task uses automatically acquired selectional preferences to sense tag subject and object head nouns, along with the associated verbal predicates. The selectional preferences comprise probability distributions over WordNet nouns, and these distributions are conditioned on WordNet verb classes. The conditional distributions are used directly to disambiguate the head nouns. We use prior distributions and Bayes rule to compute the highest probability verb class, given a noun class. We also use anaphora resolution and the 'one sense per discourse' heuristic to cover nouns and verbs not occurring in these relationships in the target text. The selectional preferences are acquired without recourse to sense tagged data so our system is unsupervised.

1 Introduction

In the first SENSEVAL, we used automatically acquired selectional preferences to disambiguate head nouns occurring in specific grammatical relationships (Carroll and McCarthy, 2000). The selectional preference models provided co-occurrence behaviour between WordNet synsets[1] in the noun hyponym hierarchy and verbal predicates. Preference scores, based on mutual information, were attached to the classes in the models. These scores were conditioned on the verbal context and the grammatical relationship in which the nouns for training had occurred. The system performed comparably to the other system using selectional preferences alone.

The work here is an extension of this earlier work, this time applied to the English all words task. We use probability distributions rather than mutual information to quantify the preferences. The preference models are modifications of the Tree Cut Models (TCMs) originally proposed by Li and Abe (1995; 1998). A TCM is a set of classes cutting across the WordNet noun hypernym hierarchy which covers all the nouns of WordNet disjointly, i.e. the classes in the set are not hyponyms of one another. The set of classes is associated with a probability distribution. In our work, we acquire TCMs conditioned on a verb class, rather than a verb form. We then use Bayes rule to obtain probability estimates for verb classes conditioned on co-occurring noun classes.

Using selectional preferences alone for disambiguation enables us to investigate the situations when they are useful, as well as cases when they are not. However, this means we loose out in cases where preferences do not provide the necessary information and other complementary information would help. Another disadvantage of using selectional preferences alone for disambiguation is that the preferences only apply to the grammatical slots for which they have been acquired. In addition, selectional preferences only help disambiguation for slots where there is a strong enough tie between predicate and argument. In this work, we use subject and object relationships, since these appear to work better than other relationships (Resnik, 1997; McCarthy, 2001), and we use argument heads, rather than the entire argument phrase.

Our basic system is restricted to using only selectional information, and no other source of disambiguating information. However, we ex-

* This work was supported by UK EPSRC projects GR/L53175 'PSET: Practical Simplification of English Text' and GR/N36462/93 'Robust Accurate Statistical Parsing (RASP)'.

[1] We will hereafter refer to WordNet synsets as classes.

perimented with two methods of extending the coverage to include other grammatical contexts. The first of these methods is the 'one sense per discourse' heuristic (Gale et al., 1992). With this method a sense tag for a given word is applied to other occurrences of the same word within the discourse. The second method uses anaphora resolution to link pronouns to their antecedents. Using the anaphoric links we are able to use the preferences for a verb co-occurring with a pronoun with the antecedent of that pronoun.

2 System Description

There is a training phrase and a run-time disambiguation phase for our system. In the training phase a preprocessor and parser are used to obtain training data for selectional preference acquisition. At run-time the preprocessor and parser are used for identifying predicates and argument heads for application of the acquired selectional preferences for disambiguation. Anaphora resolution is used at run-time to make links between antecedents of nouns, where the antecedents or the predicates may be in subject or object relationships.

2.1 Preprocessor and Parser

The preprocessor consists of three modules applied in sequence: a tokeniser, a part-of-speech (PoS) tagger, and a lemmatiser. The tokeniser comprises a small set of manually-developed finite-state rules for identifying word and sentence boundaries. The tagger (Elworthy, 1994) uses a bigram HMM augmented with a statistical unknown word guesser. When applied to the training data for selectional preference acquisition it produces the single highest-ranked tag for each word; at run-time it returns multiple tags whose associated forward-backward probabilities are incorporated into parse probabilities. The lemmatiser (Minnen et al., 2001) reduces inflected verbs and nouns to their base forms.

The parser uses a 'shallow' unification-based grammar of English PoS tags, performs disambiguation using a context-sensitive probabilistic model (Carroll and Briscoe, 1996), and recovers from extra-grammaticality by returning partial parses. The output of the parser is a set of *grammatical relations* specifying the syntactic dependency between each head and its dependent(s), read off from the phrase structure tree that is returned from the disambiguation phase. For selectional preference acquisition we applied the analysis system to the 90 million words of the written portion of the British National Corpus (BNC); both in the acquisition phase and at run-time we extracted from the analyser output only subject-verb and verb-direct object dependencies[2]. Thus we did not use the SENSEVAL-2 Penn Treebank-style bracketings supplied for the test data.

2.2 Selectional Preferences

A TCM provides a probability distribution over the noun hyponym hierarchy of WordNet. We acquire TCMs conditioned on WordNet verb classes to represent the selectional preferences of the verbs in that verb class. The noun frequency data used for acquiring a TCM is that occurring with verbs from the target verb class. The verb members for training are taken from the class directly and all hyponym classes. However not all verbs in a verb class are used for training. We use verbs which have a frequency at or above 20 in the BNC, and belong to no more than 10 WordNet classes.

The noun data is used to populate the hypernym hierarchy with frequencies, where the frequency count for any noun is divided by the number of noun classes it is a member of. A hyperonym class includes the frequency credit attributed to all its hyponyms.

A portion of two TCMs is shown in figure 1. The TCMs are similar as they both contain direct objects occurring with the verb *seize*; the TCM for the class which includes *clutch* has a higher probability for the **entity** noun class compared to the class which also includes *assume* and *usurp*. This example includes only classes at WordNet roots, although it is quite possible for the TCM to use more specific noun classes. The method for determining the generalisation level uses the minimum description length principle and is a modification of that proposed by Li and Abe (1995; 1998). In our modification, all internal nodes of WordNet have their synonyms placed at newly created leaves. Doing this ensures that all nouns are

[2] In a previous evaluation of grammatical relation accuracy, the analyser returned subject-verb and verb-direct object dependencies with 84–88% recall and precision (Carroll et al., 1999).

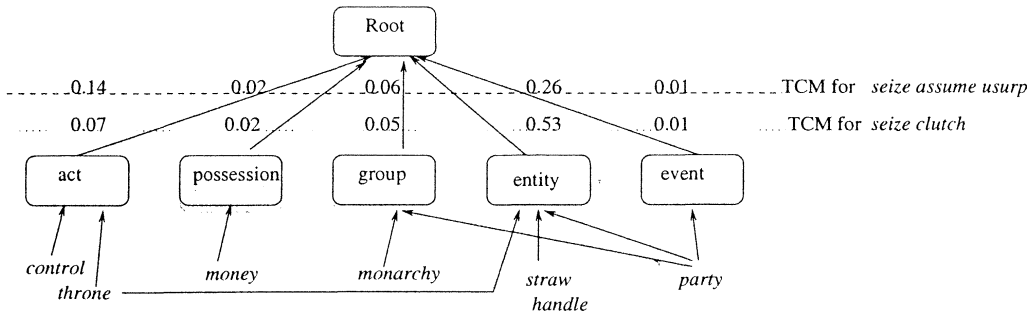

Figure 1: TCMs for the direct object slot of two verb classes which include the verb *seize*.

covered by the probability distribution specified by the TCM.

2.3 Disambiguation

The probability distributions enable us to get estimates for $p(noun\ class|verb\ class)$ for disambiguation. To disambiguate a noun occurring with a given verb, the noun class ($n1$) out of all those to which the noun belongs that gives the largest estimate for $p(n1|v1)$ is taken, where the verb class ($v1$) is the one for the co-occurring verb which maximises this estimate. The selectional preferences provide an estimate for $p(n1|v1)$. The probability estimate of the hyperonym noun class ($n2$) occurring above $n1$ on the TCM for $v1$ is multiplied by the ratio of the prior probability estimate for the hyponym divided by that for the hyperonym on the TCM, i.e. by $\frac{p(n1)}{p(n2)}$. These prior estimates are taken from populating the noun hypernym hierarchy with the prior frequency data.

To disambiguate a verb occurring with a given noun, the verbclass ($v2$) which gives the largest estimate for $p(v2|n3)$ is taken. The noun class ($n3$) for the co-occurring noun is taken as the one that maximises this estimate. Bayes rule is used to obtain this estimate:

$$p(v2|n3) = p(n3|v2)\ \frac{p(v2)}{p(n3)}$$

The TCMs for the candidate verb classes are used for the estimate of $p(n3|v2)$. The estimate for $p(n3)$ is taken from a frequency distribution stored over the entire noun hyponym hierarchy for the prior noun data for the target grammatical slot. The estimate $p(v2)$ is taken from a frequency distribution over the entire verb hyponym hierarchy for the given grammatical slot.

2.4 Increasing Coverage – OSPD and anaphora resolution

When applying the one sense per discourse (OSPD) heuristic, we simply used a tag for a noun, or verb to apply to all the other nouns (or verbs) in the discourse, provided that there was not more than one possible tagging provided by the selectional preferences for that discourse.

In order to increase coverage of the selectional preferences we used anaphoric links to allow preferences of verbs occurring with pronouns to apply to antecedents.

The anaphora resolution algorithm implemented is due to Kennedy and Boguraev (1996). The algorithm resolves third person pronouns, reciprocals and reflexives, and its cited accuracy is 75% when evaluated on various texts taken from the World Wide Web.

The algorithm places each discourse referent into a coreference class, where discourse referents in the same class are believed to refer to the same object. The classes have a salience value associated with them, and an antecedent for a pronoun is chosen from the class with the highest salience value. The salience value of a class is computed by assigning weights to the grammatical features of its discourse referents, and these grammatical features are obtained from the Briscoe and Carroll (1996) parser.

3 Evaluation

We entered three systems for the SENSEVAL-2 English all words task:

sussex-sel Selectional preferences were used alone. Preferences at the subject slot were applied first, if these were not applicable then the direct object slot was tried.

System (sussex-)	Precision (%)	Recall (%)	Attempted (%)
sel	59.8	14.0	23
sel-ospd	56.6	16.9	30
sel-ospd-ana	54.5	16.9	31

Table 1: English all words fine-grained results

Slot	Nouns (%)	Verbs (%)
subject	34	36
direct object	28	45
random baseline	24	25

Table 2: Analysis of sussex-sel precision for polysemous nouns and verbs

sussex-sel-ospd The selectional preferences were applied first, followed by the one sense per discourse heuristic. In the English all words task a discourse was demarcated by a unique text identifier.

sussex-sel-ospd-ana The selectional preferences were used, then the anaphoric links were applied to extend coverage, and finally the one sense per discourse was applied.

The results are shown in table 1. We only attempted disambiguation for head nouns and verbs in subject and direct object relationships, those tagged using anaphoric links to antecedents in these relationships and those tagged using the one sense per discourse heuristic. We do not include the coarse-grained results which are just slightly better than the fine-grained results, and this seems to be typical of other systems. We did not take advantage of the coarse grained classification as this was not available at the time of acquiring the selectional preferences.

From analysis of the fine-grained results of the selectional preference results for system sussex-sel, we see that nouns performed better than verbs because there were more monosemous nouns than verbs. However, if we remove the monosemous cases, and rely on the preferences, the verbs were disambiguated more accurately than the nouns, having only a 1% higher random baseline. Also, the direct object slot outperformed the subject slot. In future it would be better to use the preferences from this slot first.

4 Conclusions

Given that this method is unsupervised, we feel our results are promising. The one sense per discourse heuristic works well and increases coverage. However, we feel that anaphora resolution information has not reached its full potential. There is plenty of scope for combining evidence from several anaphoric links, especially once we have covered more grammatical relationships. We hope that precision can also be improved by combining or comparing several pieces of evidence for a single test item. We are currently acquiring preferences for adjective-noun relationships.

References

John Carroll and Ted Briscoe. 1996. Apportioning development effort in a probabilistic LR parsing system through evaluation. In *ACL/SIGDAT Conference on Empirical Methods in Natural Language Processing*, pages 92–100, University of Pennsylvania, PA.

John Carroll and Diana McCarthy. 2000. Word sense disambiguation using automatically acquired verbal preferences. *Computers and the Humanities. Senseval Special Issue*, 34(1–2):109–114.

John Carroll, Guido Minnen, and Ted Briscoe. 1999. Corpus annotation for parser evaluation. In *EACL-99 Workshop on Linguistically Interpreted Corpora*, pages 35–41, Bergen, Norway.

David Elworthy. 1994. Does Baum-Welch re-estimation help taggers? In *4th ACL Conference on Applied Natural Language Processing*, pages 53–58, Stuttgart, Germany.

William Gale, Kenneth Church, and David Yarowsky. 1992. A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26:415–439.

Chris Kennedy and Bran Boguraev. 1996. Anaphora for everyone: Pronominal anaphora resolution without a parser. In *16th International Conference of Computational Linguistics, COLING-96*, pages 113–118.

Hang Li and Naoki Abe. 1995. Generalizing case frames using a thesaurus and the MDL principle. In *International Conference on Recent Advances in Natural Language Processing*, pages 239–248, Bulgaria.

Hang Li and Naoki Abe. 1998. Generalizing case frames using a thesaurus and the MDL principle. *Computational Linguistics*, 24(2):217–244.

Diana McCarthy. 2001. *Lexical acquisition at the syntax-semantics interface: diathesis alternations, subcategorization frames and selectional preferences.* Ph.D. thesis, University of Sussex.

Guido Minnen, John Carroll, and Darren Pearce. 2001. Applied morphological processing of English. *Natural Language Engineering*, 7(3):207–223.

Philip Resnik. 1997. Selectional preference and sense disambiguation. In SIGLEX *Workshop on Tagging Text with Lexical Semantics: Why What and How?*, pages 52–57, Washington, DC.

Combination of contextual features for word sense disambiguation: LIU-WSD

Magnus MERKEL & Mikael ANDERSSON
Dept. of Computer and Information Science,
Linköping University
S581 83 Linköping, Sweden,
magme@ida.liu.se, miand@ida.liu.se

Abstract

This paper describes a system for word sense disambiguation that participated in the Swedish Lexical Sample task of SENSEVAL-2. The system LIU-WSD is based on letting different contextual features cast votes on preferred senses according to a ranking scheme.

Introduction

The addition of new languages to the SENSEVAL-2 workshop, among these languages also Swedish, presented an opportunity to learn more about WSD applied to Swedish by participation in the event. Previously, we had had no experience of building word sense disambiguation software, but the Swedish Lexical Sample task seemed like a suitable occasion for trying another field of NLP (in recent years our focus has been on word alignment and parallel corpora).

Due to time constraints our initial plans of implementing some kind of version of decision lists (Yarowsky, 2000; Pedersen 2001) were abandoned in the end and we decided to go for a slightly simpler approach based on a general algorithm and voting strategies for contextual features on different levels. The contextual features that were being considered were unigrams and bigrams, both in fixed and variable positions, together with possibilities to include parts-of-speech, lemmas and graph words (inflected words).

1 Data and pre-processing

The data and resources used in the LIU-WSD system were apart the following:

- Sample and training data, provided by the task organisers – the sample data were a great help in order to understand the format of the provided material.
- Part of the lexicon data provided for the Swedish lexical sample task. Here only the information on the number of senses and division of main and sub senses was used. Information contained in examples, definitions and for valency was left out.

As the system was a first attempt to word sense disambiguation for us, we set up a small corpus containing for five lexical items with around 60 contexts for each lexeme. This was used to test various approaches and algorithms as well as making sure of conversions to and from the format used in the task.

As we wanted to use morphosyntactic information and lemmas in the system, the Swedish Constraint Grammar package, SWECG-2 from Conexor was used (Karlsson et al., 1994). The training corpus was fed into the tagger, SWECG-2, which returned an XML file where the text was POS tagged and lemmatised. Below is a sample of how the sentence *Men påståendet är litet missvisande.* came back in XML format.

```
<instance id="barn.19">
<answer instance="barn.19"/>
<context>
<w id="w1" base="men" pos="CC">men</w>
<w id="w2" base="påstående"
   pos="N">påståendet</w>
<w id="w3" base="vara" pos="PRES">är</w>
<w id="w4" base="litet & litet" pos="ADV &
   A">litet</w>
<w id="w5" base="missvisa"
   pos="NDE">missvisande</w>
  <w id="w6" base="."
   pos="interpunction">.</w>...
```

2 Training

The actual training was a matter of building tables of information from the training corpus. The task was first to decide on a number of contextual features to be observed, then set

SENSES	frq	1	1.a	1.b	1.c	1.d	1.e	1.f	1.g	2	2.a	2.b	2.c	2.d	3	4
Distrib.	152	30	12	0	7	15	4	1	14	6	7	0	32	3	2	19
Rel.freq.		.19	.07	-	.04	.09	.02	-	.09	.03	.04	-	.21	.01	.01	.12
Stand.dev.		.03	.02	-	.01	.02	.01	-	.02	.01	.01	-	.03	.01	-	.02
egen	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Rel.freq.		1.0	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ratio		5	-	-	-	-	-	-	-	-	-	-	-	-	-	-
T-score		24.8	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 1. Extract from training data for the lexeme *kraft,* and the contextual feature [-1, lemma] when *egen* is observed in position -1. The upper part of the table shows the sense distribution in the training corpus and the lower part give data on *egen* when seen immediately in front of *kraft.*

thresholds and store the information as a database where each lexeme was represented with a number of relevant contextual features observed in the training data.

Basically, a table contains general information on the sense distribution in the training corpus, including frequencies, relative frequencies, standard deviation and a statistical measure, the Student T-measure. The idea behind the statistical measure is to use a test that can tell whether a certain observation of a contextual feature in relation to the choice of a word sense is statistically significant or not. We test whether the existence of a certain contextual feature F changes the distribution of senses for a certain sense S. There are many tests that can be used. We chose to use the Student T-test. The

$$p = \frac{s}{T}$$

probability p for a sense S is estimated as:
where s is the total number of contexts holding word W with sense S, and T is the total number of contexts where the word W occurs.

To arrive at a t-score we set N as the number of contexts for W where the feature F is holds, and n as the number of contexts among N that contain the sense S. We then calculate p':

$$p' = \frac{n}{N}$$

To be interesting p' must be greater than p. We can now test whether the distribution for F is the same as if when F is not observed, which is the actual t-test. We then test *H0* (p=p') vs. *H1* (p<p') and calculate t as

$$t = \frac{p' - p}{dev}$$

where *dev* is the standard deviation for p, testing on the 95% confidence level. An example how the t-score looks for a particular feature is illustrated in Table 1.

2.1 Feature patterns

Twelve general feature patterns were extracted from the training corpus. The patterns are defined by choosing options on three different levels: (i) unigram or bigram, (II) lemma (base form), graph word (inflected form) or parts-of-speech category (POS), and (iii) fixed position or position within an interval.[1]

1. Unigram, lemma, position -1
2. Unigram, lemma, position +1
3. Unigram, lemma, position -5 to -2
4. Unigram, lemma, position +2 to +5
5. Unigram, lemma, all positions
6. Unigram, POS, position -2 to -1
7. Unigram, POS, position +1 to +2
8. Bigram, lemma, position -2 to -1
9. Bigram, lemma, position +1 to +2
10. Bigram, lemma, all positions
11. Bigram, POS, position -2 to -1
12. Bigram, POS, position +1 to +2

Patterns 1-7 all concern unigrams, while the rest operate on bigrams. Pattern 11, for example, will extract all POS bigrams in position -2 to -1, i.e., the POS tags for the two words that immediately precede the head word (which is assumed to be in position 0).

For each lexeme, a table was built like the one in Table 1 for each of the twelve general feature

[1] Feature 3, 4, 5, 6, 7 and 10 are all be bags of word features. The others indicate fixed positions.

patterns. The frequency threshold used was 3 and function words were ignored except in the bigram patterns.

3 Procedure and algorithm

When a test instance is to be disambiguated, a pre-processor first matches the test context with the training data for the lexeme in question. This involves identifying exactly those contextual features from the test data that are applicable to the test instance. The filtered set of tables is then used as input to the main disambiguation algorithm.

The algorithm is based on a voting strategy combined with some heuristics. The voting strategy entails that all classes of feature patterns (at the most 12 classes) will cast votes for a particular sense. The class vote is determined by which sense is ranked highest within that class. The winner for each class is determined by the number of sense choices for the included features of that class. For example, for the instance 9 in the test corpus of the lexeme *barn* (Eng. *child*), the voting would result in choosing sense *barn_1_1* as it was ranked as number one in three classes, see below.

```
barn.9:
VOTES for senses:
Sense
   1_1:  Rank1: 3
   1_1.a:  Rank3: 1
   1_1.b:  Rank1: 1, Rank3: 1
   1_2:  Rank1: 1, Rank2: 1
   1_2.a:  Rank3: 1
sense selected: barn_1_1
```

During the training phase it was discovered that features that contained a relative frequency of 1.0 (i.e. all observations of the feature only occurred with a single sense) could be considered as a relatively "sure sense", we included this strategy in the algorithm.

The basic outline of the algorithm is as follows:

1. If there are no applicable data for the instance (i.e. no tables), pick the most common sense in the training corpus as the sense for this instance.
2. Otherwise, if there is a "sure sense", i.e. a contextual feature is found with the relative frequency 1.0, this sense is selected, if there is no conflict between several "sure senses".
3. Check the t-scores for all features in every class, and rank each sense as Rank1, Rank2, Rank3, etc. Each feature class will then be ranked and will cast votes accordingly. If there is a sense winner (i.e. most number of first places in the feature classes), this sense is selected.
4. If there are several senses that are tied, check how many votes they have for second places, and the best one of them is chosen as the sense.
5. If there's still a tie when considering second places, start clustering senses together into a main sense, for example, 1_1.a, 1_1_b, and 1_1_c are all considered as sense 1.1. Only those senses that were tied for first place are considered, but if these senses all share a single main sense, then that sense is chosen.
6. If there be several main senses, go back to original (sub-)senses, and select the sense with highest frequency in the training corpora.
7. If all of the above fails, simply resort to taking the most common sense in the training sample.

4 Results

The official results for the LIU-WSD system in the Swedish Lexical Sample task were the following:

Fine-grained scoring: 56.5 per cent precision, 56. 5 per cent recall.
Mid-grained scoring: 61.6 per cent precision, 61.6 per cent recall.

The coarse-grained scoring is not relevant as an evaluation criterion due to the fact that the tested lexemes were categorised as belonging to the same main sense, which means that all results received precision scores of 100 per cent.

5 Evaluation

As the results were slightly worse than we had hoped for, it is interesting to point out further details from the scores and on the strategies that were actually used by the system. Table 2 illustrates what part of the algorithm that was used for selecting a particular sense and how well each strategy worked. It is notable that

Table 2. Overview of sense selecting criteria for the LIU-WSD system in SENSEVAL-2

	TOTAL	CORRECT	ERROR	PRECISION
Sure senses:	875	619	256	70%
Voted Rank1	393	194	199	49%
Tied Rank2	115	28	87	24%
Common main sense	27	7	20	25%
Most frequent main sense:	38	7	31	18%
Sub sense	77	17	60	22%
Most frequent sense	0	0	0	
	1525	**872**	**653**	

more than 50 per cent of the selections were made by the heuristics to select a "sure sense" based on the relative frequency. This is also the most successful in terms of precision of all the strategies. The voting strategies performed far worse, 49% for selection based on senses that were ranked first, and only 24% when a tie for first place was found and the second positions were considered. The strategies to select a shared common main sense, a most frequent main sense or most frequent sub sense when the other criteria failed were clearly not very successful, as indicated by precision rates varying from 18 to 25 per cent.

It is also worth pointing out that there were always some significant features for each instance of the test corpus, which meant that step 1 of the algorithm never triggered. The same is true for the last step. If we break down the results into different parts-of-speech, we can see the following:

Table 3. Results for nouns, verbs and adjectives

NOUNS	
Precision:	74%
Better than baseline (MFS):	20/20
Average no. of senses:	8.05
VERBS	
Precision:	40.4%
Better than baseline (MFS):	12/15
Average no. of senses:	14.26
ADJECTIVES	
Precision:	40%
Better than baseline (MFS):	4/5
Average no. of senses:	14.4

As has been noted elsewhere, nouns are easier than verbs and adjectives when it comes to word sense disambiguation (cf. Yarowsky 2000). This is clearly the case here, and a contributing factor to this is that the number of senses for nouns is significantly smaller than for the other word classes. However, the system does in general perform better than the standard baseline (Most Frequent Sense of the training corpus). For nouns, all 20 test instances are better than the baseline. There is however work to be done to improve the performance for verbs and adjectives.

6 Discussion

Clearly this system can be improved further, especially when it comes to how the voting system should be set up. As the feature classes sometimes are overlapping, some features will contribute several times to the votes (but from different classes), therefore some kind of inductive Machine Learning algorithm to infer which combination of features is the best should be tested. Another possible improvement would be to include information from the examples in the lexicon and also to include the inflected form of the word that is to be disambiguated in the process.

Acknowledgements

Our thanks go to the organisers of SENSEVAL-2 and the people in Gothenburg who made sure that Swedish was a part of it all. Also we would like to thank Lars Ahrenberg giving valuable input on the use of the T-test used in the system.

References

Karlsson, F,. A. Voutilainen, J. Heikkilä & A. Antilla (1994). *A Language-Independent System for Parsing Unrestricted Text.* Mouton de Gruyter, Berlin and New York.

Pedersen T. (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.

Yarowsky D. (2000). Hierarchical Decision Lists for Word Sense Disambiguation. *Computers and the Humanities*, 34(2):179-186, 2000

Pattern Learning and Active Feature Selection for Word Sense Disambiguation

Rada F. MIHALCEA
Southern Methodist University
Dallas, Texas, 75275-0122
rada@seas.smu.edu

Dan I. MOLDOVAN
University of Texas at Dallas
Richardson, Texas, 75083-0688
moldovan@utdallas.edu

Abstract

We present here the main ideas of the algorithm employed in the *SMUls* and *SMUaw* systems. These systems have participated in the SENSEVAL-2 competition attaining the best performance for both English all words and English lexical sample tasks[1]. The algorithm has two main components (1) pattern learning from available sense tagged corpora (SemCor) and dictionary definitions (WordNet), and (2) instance based learning with active feature selection, when training data is available for a particular word.

1 Introduction

It is well known that WSD constitutes one of the hardest problems in Natural Language Processing, yet is a necessary step in a large range of applications including machine translation, knowledge acquisition, coreference, information retrieval and others. This motivates a continuously increasing number of researchers to develop WSD systems and devote time to finding solutions for this challenging problem.

The system presented here was initially designed for the semantic disambiguation of *all words* in open text. The SENSEVAL competitions created a good environment for supervised systems and this encouraged us to improve our system with the capability of incorporating larger training data sets when provided.

There are two important modules in this system. The first one uses pattern learning relying on large sense tagged corpora to tag all words in open text. The second module is triggered only for the words with large training data, as was the case with the words from the lexical sample tasks. It uses an instance based learning algorithm with active feature selection.

To our knowledge, both pattern learning and active feature selection are novel approaches in the WSD field, and they led to very good results during the SENSEVAL-2 evaluation exercise.

2 System description

The WSD algorithm used in this system has the capability of tagging words when no specific sense tagged corpora is available, automatically scaling up to larger training data[2] when provided.

Due to space constraints, we will not be able to give a detailed description of the system. However we try to gain space and replace one thousand words with a picture: Figure 1 shows an overview of the system architecture. It illustrates the two main components, namely pattern learning from available sense tagged corpora and dictionary definitions and instance based learning with active feature selection. The two modules are preceded by a preprocessing phase which includes compound concept identification, and followed by a default phase that assigns the most frequent sense as a last resort, when no other previous methods could be applied. The shaded areas in Figure 1 are specific for the case when larger training data sets are available.

During the preprocessing stage, SGML tags are eliminated, the text is tokenized, part of speech tags are assigned using Brill tagger (Brill, 1995), and Named Entities (NE) are identified with an *in-house* implementation of an NE recognizer. To identify collocations, we determine sequences of words that form compound concepts defined in WordNet.

In the second step, patterns[3] are learned from WordNet, SemCor and GenCor, which is a large

[1] This is in conformity with the original ranking, following the evaluation of systems answers submitted before deadline.

[2] I.e. in addition to the publicly available sense tagged corpora

[3] We alternatively call them *rules* as they basically specify the sense triggered by a given local context, using rules like "if the word before is X then sense is Y"

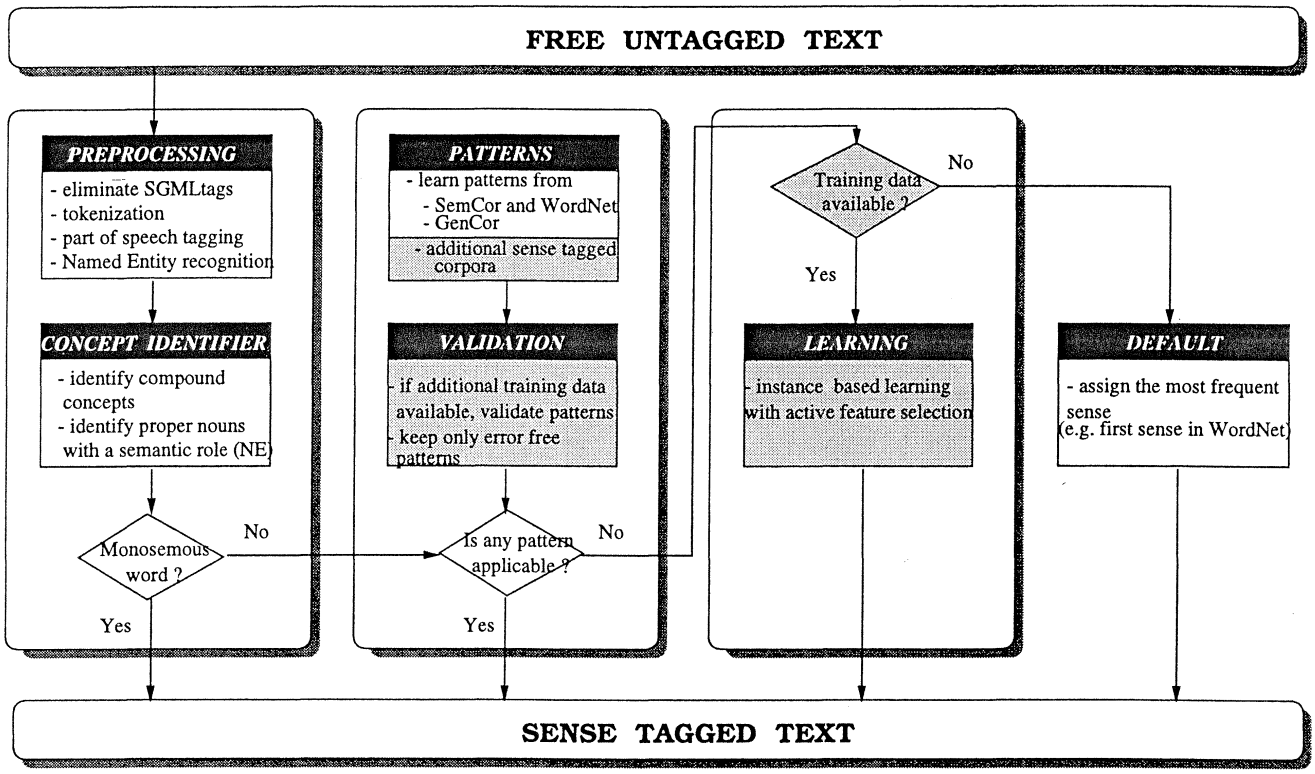

Figure 1: System architecture

sense tagged corpus automatically built via a set of heuristics. If additional training data is available, patterns are filtered through a validation process. Practically, the patterns are applied on the sense tagged data and we keep only those with no counter-examples found in the training sets.

The third step consists of a learning mechanism with active feature selection. This step is initiated only for those words with a sufficiently large number of examples, as was the case with the words in the SENSEVAL lexical sample tasks.

3 Pattern learning

This module is intended for solving the semantic ambiguity of *all* words in open text. To this end, we build disambiguation patterns using SemCor, WordNet and GenCor. Several processing steps were required to transform the first two resources into a useful corpus for our task. Moreover, these lexical resources coupled with a set of heuristics were used as seeds for generating a new sense tagged corpus called GenCor.

SemCor The SENSEVAL-2 English tasks have decided to use the WordNet 1.7 sense inventory, and therefore we had to deal with the task of mapping SemCor senses, which were assigned using an earlier version of WordNet, to the corresponding senses in WordNet 1.7. When a word sense from WordNet 1.6 is missing we assign a default sense of 0.[4]

WordNet The main idea in generating a sense tagged corpus out of WordNet is very simple. It is based on the underlying assumption that each example pertains to a word belonging to the current synset, thereby allowing us to assign the correct sense to at least one word in each example. For instance, the example given for *mother#4* is *"necessity is the mother of invention"*, and the word *mother* can be tagged with its appropriate sense.

GenCor is a generated sense tagged corpus, containing at the moment about 160,000 tagged words, which uses as seeds the sense tagged examples from SemCor and WordNet, as well as some of the principles for generating sense tagged corpora presented in (Mihalcea and Moldovan, 1999). Due to space limitations we cannot present here the methodology for creating this corpus. A thorough description is provided in (Mihalcea, 2001).

[4]SemCor 1.7a is available for download at http://www.seas.smu.edu/~rada/semcor

Once we have created this large corpus with examples of word meanings, we can start to learn patterns. A pattern basically consists of the local context for each semantically tagged word found in the corpus. The local context is formed by a window of N words to the left and M words to the right of each word considered. Additionally, a set of constraints is applied to filter out meaningless patterns.

Patterns are formed following the rules for regular expressions. Each word in the corpus is represented by its base form, its part of speech, its sense, if there is any provided, and its hypernym, again if the sense is known. Any of these word components can be unspecified, and therefore denoted with the symbol *. A count is also associated with every pattern, indicating the number of times it occurred in the corpus.

When trying to disambiguate a word, first we search for all available patterns that match the current context. In doing so, we use the current word as a pivot to perform matching. If there are several patterns available, then the decision of which pattern to apply is based on the pattern *strength*. The strength of a pattern is evaluated in terms of (1) number of specified components, (2) number of occurrences and (3) pattern length.

<the/DT modal/JJ/1 age/NN at/IN> is considered to be stronger than <modal/NN/1 age/NN>. Also, <clear/JJ/4 water/NN/1> is stronger than <clear/JJ water/NN/1>. Moreover, the inclusion of the hypernym among the word components gives us the means for generalization. For instance, <*/NN/*/room/1 door/NN/1> matches "kitchen door" as well as "bedroom door".

Another important step performed during the all words disambiguation task is sense propagation. The patterns do not guarantee a complete coverage of all words in input text, and therefore additional methods are required. We use a cache-like procedure which assigns to each ambiguous word the sense of its closest occurrence, if any can be found. The words still ambiguous at this point are assigned by default the first sense in WordNet.

Words with a significant number of semantic tagged examples constitute a special case in our system. There is a second module designed to handle the semantic disambiguation of these words. This module, described in the following section, exploits the benefits of having larger training data available for a particular word.

4 Learning with active feature selection

Learning mechanisms for disambiguating word senses have a long tradition in the WSD field. For our system, we have decided for an instance based algorithm with information gain feature weighting. The reasons for this decision are three fold: first, it has been advocated that forgetting exceptions is harmful for language learning applications (Daelemans et al., 1999), and instance based algorithms are known for their property of taking into consideration every single training example when making a classification decision; secondly, instance based learning algorithms have been successfully used in WSD applications (Veenstra et al., 2000); finally, this type of algorithms are efficient in terms of training and testing time. We have initially used the MLC++ implementation, and later on switched to Timbl (Daelemans et al., 2001).

Even more important than the choice of learning methodology is the selection of features employed during the learning process. There are several features recognized as good indicators of word sense, including the word itself (CW) and its part of speech (CP), surrounding words and their parts of speech (CF), collocations (COL), syntactic roles, keywords in contexts (SK). More recently, other possible features have been investigated: bigrams in context (B), named entities (NE), the semantic relation with the other words in context, etc.

Our intuition was that different sets of features have different effects depending on the ambiguous word considered. Feature weighting was clearly proven to be an advantageous approach for a large range of applications, including WSD. Still, weights are computed independently for each feature and therefore this strategy does not always guarantee to provide the best results.

For our system, we actively select features using a forward search algorithm. In this way, we practically generate *meta word experts*. Each word will have a different set of features that will eventually lead to the best disambiguation accuracy.

Using this approach, we combine the advantages of instance based learning mechanisms that have the nice property of *"not forgetting exceptions"*, with an optimized feature selection scheme. One could argue that decision trees have the capability of selecting relevant features, but

it has been shown (Almuallim and Dietterich, 1991) that irrelevant features significantly affect the performance of decision trees as well.

The algorithm for active feature selection is sketched in Figure 2. It is worth mentioning that in step 2, the training and testing corpora are extracted for each ambiguous word. This means that examples pertaining to the word *"dress down"* are separated from the examples for the single word *"dress"*.

1. *Generate pool of features PF = $\{F_i\}$. Initialize the set of selected features with the empty set SF=$\{\emptyset\}$.*
2. *Extract training and testing corpora for the given target ambiguous word.*
3. *For each feature F_i in the pool PF:*
   - 3.1. *Run a 10-fold cross validation on the training set; each example in the training set contains the features in SF* **and** *the feature F_i.*
   - 3.2. *Determine the feature F_i leading to the best accuracy.*
   - 3.3. *Remove F_i from PF and add it to SF.*
4. *Repeat step 3 until no improvements are obtained.*

Figure 2: Algorithm for active feature selection

The pool PF contains a large number of features, including those previously mentioned CW, CP, CF, COL, SK, B, NE, as well as other features like the noun before and after (NB, NA), head of the noun phrase, surrounding verbs, and others.

5 Results in SENSEVAL-2

The overall performance of the system in the English all words task was 69% for fine-grained scoring, respectively 69.8% for coarse-grained scoring (*SMUaw*). In the English lexical sample task, we obtained 63.8% for fine-grained scoring, respectively 71.2% for coarse-grained scoring (*SMUls*). These results ranked our system before deadline as the best performing for both tasks.

Discussion

There were several interesting cases encountered in the SENSEVAL-2 data, justifying our approach of using *active* feature selection. The influence of a feature greatly depends on the target word: a feature can increase the precision for a word, while making things worse for another word. For example, a word such as *free* does not benefit from the surrounding keywords (SK) feature, whereas *colourless* gains almost 7% in precision when this feature is used.

free.a[CW CP CF SK]	→	57.85%
free.a[CW CP CF]	→	63.57%
colourless.a[CW CP CF]	→	78.57%
colourless.a[CW CP CF SK]	→	85.71%

Another interesting example is constituted by the noun *chair*, which was disambiguated with high precision by simply using the current word (CW) feature. This is explained by the fact that the most frequent senses are **C***hair* meaning *person* and **c***hair* meaning *furniture*, and therefore the distinction between lower and upper case spellings makes the distinction among the different meanings of this word.

We have also tested the system on the SENSEVAL-1 data, and performed the disambiguation task in respect with Hector definitions, as required by the first disambiguation exercise. The overall result achieved on this data was higher than the one reported by the best performing system. Besides proving the validity of our approach, this fact also proved that our system is not tight in any ways to the sense inventory or data format employed.

6 Conclusion

Pattern learning and active feature selection are new approaches in the WSD field. They have been implemented in a system that participated in the SENSEVAL-2 competition, with an excellent performance in both *English all words* and *English lexical sample* tasks.

References

H. Almuallim and T.G. Dietterich. 1991. Learning with many irrelevant features. In *Proceedings of AAAI-91*, volume 2, pages 547–552, Anaheim, California.

E. Brill. 1995. Transformation-based error driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–566, December.

W. Daelemans, A. van den Bosch, and J. Zavrel. 1999. Forgetting exceptions is harmful in language learning. *Machine Learning*, 34(1-3):11–34.

W. Daelemans, J. Zavrel, K. van der Sloot, and A. van den Bosch. 2001. Timbl: Tilburg memory based learner, version 4.0, reference guide. Technical report, University of Antwerp.

R. Mihalcea and D.I. Moldovan. 1999. An automatic method for generating sense tagged corpora. In *Proceedings of AAAI-99*, Orlando, FL, July.

R. Mihalcea. 2001. GenCor: a large semantically tagged corpus. (in preparation).

J. Veenstra, A. van den Bosch, S. Buchholz, W. Daelemans, and J. Zavrel. 2000. Memory-based word sense disambiguation. *Computers and the Humanities*, 34:171–177.

The University of Alicante Word Sense Disambiguation System*

Andrés Montoyo and **Armando Suárez**
Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante
Alicante, Spain
{montoyo | armando}@dlsi.ua.es

Abstract

The WSD system presented at SENSEVAL-2 uses a knowledge-based method for noun disambiguation and a corpus-based method for verbs and adjectives. The methods are, respectively, Specification Marks and Maximum Entropy probability models. So, we can say that this is a hybrid system which joins an unsupervised method with a supervised method. The whole system has been used in lexical sample english task and lexical sample spanish task.

1 Introduction

In this paper a Word Sense Disambiguation system based on Specification Marks (SM) and Maximum Entropy probability models (ME) is presented. SM is an unsupervised knowledge-based method and has been applied to noun disambiguation. ME belongs to the statistical approach to WSD in NLP and uses a tagged corpus in order to learn a probability model that can be used to predict the correct sense of a word. SM does not need a previously tagged corpus, it uses the semantic information stored in WordNet.

The weakness of supervised corpus-based approaches rely on availability of corpora and their dependency of the data which were used in the training phase. Knowledge-based approaches use previously acquire linguistic knowledge. This knowledge is extracted from human lexicographers experience and can be in form of electronic dictionary or lexicon. While their success seems poorest than statistical methods, they don't need neither an existing corpus nor a training phase and they can be more domain independent.

* This paper has been partially supported by the Spanish Government (CICYT) project number TIC2000-0664-C02-02.

So, the University of Alicante system performs the WSD task combining unsupervised with supervised methods. The whole system has been used in lexical sample English task and lexical sample Spanish task.

2 Specification Marks Framework

The method we present here consists basically of the automatic sense-disambiguating of nouns that appear within the context of a sentence and whose different possible senses are quite related. Its context is the group of words that co-occur with it in the sentence and their relationship to the noun to be disambiguated. The disambiguation is resolved with the use of the WordNet lexical knowledge base.

The intuition underlying this approach is that the more similar two words are, the more informative the most specific concept that subsumes them both will be. In other words, their lowest upper bound in the taxonomy. (A "concept" here, corresponds to a Specification Mark (SM)). In other words, the more information two concepts share in common, the more similar they obviously are, and the information commonly shared by two concepts is indicated by the concept that subsumes them in the taxonomy.

The input for the WSD module will be the group of words $W = \{W_1, W_2, ..., W_n\}$. Each word wi is sought in WordNet, each one has an associated set $S_i = \{S_{i1}, S_{i2}, ..., S_{in}\}$ of possible senses. Furthermore, each sense has a set of concepts in the IS-A taxonomy (hypernymy/Hyponymy relations). First, the concept that is common to all the senses of all the words that form the context is sought. We call this concept the Initial Specification Mark (ISM), and if it does not immediately resolve the ambiguity of the word, we descend from one level

to another through WordNet's hierarchy, assigning new Specification Marks. The number of concepts that contain the subhierarchy will then be counted for each Specification Mark. The sense that corresponds to the Specification Mark with highest number of words will then be chosen as the sense disambiguation of the noun in question, within its given context.

At this point, we should like to point out that after having evaluated the method, we subsequently discovered that it could be improved with a set of heuristics, providing even better results in disambiguation. The set of heuristics are Heuristic of Hypernym, Heuristic of Definition, Heuristic of Common Specification Mark, Heuristic of Gloss Hypernym, Heuristic of Hyponym and Heuristic of Gloss Hyponym. Detailed explanation and evaluation of the method and heuristics can be found in (Montoyo and Palomar, 2000; Montoyo and Palomar, 2001), while its application to NLP tasks are addressed in (Montoyo et al., 2001).

3 Maximum Entropy Framework

Maximum Entropy(ME) modeling is a framework for integrating information from many heterogeneous information sources for classification. ME probability models were successfully applied to some NLP tasks such as POS tagging or sentence boundary detection (Ratnaparkhi, 1998).

The WSD system presented in this paper is based on conditional ME probability models (Saiz-Noeda et al., 2001). It implements a supervised learning method consisting of the building of word sense classifiers through training on a semantically tagged corpus. A classifier obtained by means of a ME technique consists of a set of parameters or coefficients estimated by means of an optimization procedure. Each coefficient is associated to one feature observed in training data. A feature is a function that gives a measure for some characteristic in a context associated to a class. The main purpose is to obtain the probability distribution that maximizes the entropy, that is, maximum ignorance is assumed and nothing apart of training data is considered. As advantages of ME framework, knowledge-poor features applying and accuracy can be mentioned; ME framework allows a virtually unrestricted ability to represent problem-specific knowledge in the form of features (Ratnaparkhi, 1998).

Let us assume a set of contexts X and a set of classes C. The function $cl : X \rightarrow C$ that performs the classification in a conditional probability model p chooses the class with the highest conditional probability: $cl(x) = \arg\max_c p(c|x)$, where x is a context and c a class. The features have the form of (1), where $cp(x)$ is some observable characteristic[1]. The conditional probability $p(c|x)$ is defined as (2) where α_i are the parameters or weights of each feature, and $Z(x)$ is a constant to ensure that the sum of probabilities for each possible class in this context is equal to 1.

$$f_{c'}(x,c) = \begin{cases} 1 & \text{if } c' = c \text{ and } cp(x) = true \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$p(c|x) = \frac{1}{Z(x)} \prod_{i=1}^{K} \alpha_i^{f_i(x,c)} \quad (2)$$

4 The system at Senseval-2

The Spanish and English lexical sample tasks at the SENSEVAL-2 workshop had been performed by our system in three phases. The first one is a naive multi-word detection; the second one, the disambiguation of nouns by means of the SM method, and the third one, the disambiguation of verbs and adjectives by means of the ME method.

In a previous step, training and test data had been tagged with Tree-Tagger(Schmid, 1994) for English files and Conexor's FDG Parser (Tapanainen and Järvinen,) for Spanish files in order to get the part-of-speech information and identify nouns, verbs and adjectives.

Multi-words detection

The multi-word detection has been performed by combining the words around the target word in each sample and consulting WordNet for English (examining the training data, we conclude that this is not necessary for Spanish data). If a multi-word is found in WordNet a multi-word instance is assigned and no further single word

[1]The ME approach is not limited to binary funtions, but the optimization procedure(*Generalized Iterative Scaling*) used for the estimation of the parameters needs this kind of features.

disambiguation will be done. This kind of instances has been disambiguated with the first sense of WordNet (even if it is a polysemous one).

Nouns with Specification Marks

The second phase consist of noun classification, and has been performed by the SM method described previously.

Verbs and adjectives with Maximum Entropy

The third and final phase, the verbs and adjectives disambiguation, has been performed by the ME method. The SENSEVAL-2 training data has been used in order to obtain the classification functions to be applied on the test data. The set of features defined for ME training is described below and it is based on features selection made in (Ng and Lee, 1996) and (Escudero et al., 2000).

The set of features corresponds to words around the word to classify and POS labels at positions related to the target word in each sentence: w_0, w_{-1}, w_{-2}, w_{-3}, w_{+1}, w_{+2}, w_{+3}, (w_{-2}, w_{-1}), (w_{-1}, w_{+1}), (w_{+1}, w_{+2}), (w_{-3}, w_{-2}, w_{-1}), (w_{-2}, w_{-1}, w_{+1}), (w_{-1}, w_{+1}, w_{+2}), (w_{+1}, w_{+2}, w_{+3}), p_{-3}, p_{-2}, p_{-1}, p_{+1}, p_{+2}, p_{+3}. Each w_i is the lemma of the word at position i in the context (in collocations, at least one of the words must be a content word). Each p_i is the POS label at position i.

Other set of features consists of a surrounding nouns selection. This selection is doing by means of frequency information of nouns co-occurring with a sense. Nouns co-occurring with a class in a $K\%$ of examples of that class in the corpus or more are selected to build a feature for each possible class[2].

5 Senseval-2 results analysis

Analyzing the first evaluation results of the English lexical sample task (fine-grained scoring) reported by SENSEVAL-2 committee (*precision* = 0.421 and *recall* = 0.411) , some conclusions can be extracted from them.

The nouns disambiguation obtains the worst results (see table 1). We can mostly assure that the reason is the kind of method used: knowledge-based for nouns and corpus-based for verbs and adjectives.

POS	precision	recall
Nouns	0.299	0.292
Verbs	0.486	0.480
Adjectives	0.709	0.635

Table 1: Results of the English Lexical Sample Task (Fine-grained)

The results of the Spanish lexical sample task (fine-grained scoring) reported by SENSEVAL-2 committee are *precision* = 0.514 and *recall* = 0.503. Nevertheless, the nouns results rise to 56% of precision (table 2). It seems that the set of nouns selected for this task is easier to Specification Marks than English ones, maybe related to lexical resources used and the language itself. However, the recall of nouns is too low because a implementation error causes that the accented words had not been recognize (*corazón*, *operación* and *órgano*).

POS	precision	recall
Nouns	0.566	0.435
Verbs	0.511	0.511
Adjectives	0.687	0.687

Table 2: Results of the Spanish Lexical Sample Task (Fine-grained)

The preprocessing of the train and test data are relevant. Some errors of the POS-tagger had been detected and they affect some answer instances. Multi-words are a not resolved problem. The detection and disambiguation method is too simple and causes too much errors. More preprocessing is necessary, as well: the context information can be enriched and accuracy increased with entity recognition, full-parsing, and so on.

6 Conclusions

The University of Alicante system presented at SENSEVAL-2 workshop joins the two general approaches to the WSD task: knowledge-based and corpus-based methods. The Specification Marks method belongs to the first one and Maximum Entropy-based method to the second one.

[2]For example, in a set of 100 examples of sense four of the noun "interest", if "bank" is observed 10 times or more ($K = 10\%$) then a feature for each possible sense of "interest" is defined with "bank".

Specification Marks for nouns, and Maximum Entropy for verbs and adjectives had been used in order to process the test data of the English and the Spanish lexical sample tasks. The training and the test data had been used with a minimum preprocessing, just cleaning of XML-tags in order to run the Tree-Tagger. Besides, the two WSD modules had been used in the same manner as for other corpora with minor modifications: no specific changes to the algorithms used in both methods had been made for SENSEVAL-2, apart from the necessary modules to make data files available to the computer programs.

Due to the distinct approaches used in each POS, the whole system has been classified as supervised system. In the English task, the system obtains a poor score when it is compared with other supervised systems, and a great result against the unsupervised systems (we have no such information of systems for Spanish). But the truth is that our system is unsupervised for nouns but supervised for verbs and adjectives. Therefore, comparing our results with the other systems must be done separating the results of nouns, verbs and adjectives.

7 Future and in progress work

At this moment, the two methods presented here are being improved with new knowledge sources like full parsing information and domain categories that in order to decrease the WordNet granularity. The WSD system will be completed with other NLP software like Name Entity recognition and multi-words detection modules.

Recent work in our research group indicates that it is possible to combine the two methods in a hybrid method that assign a sense to a context combining the answers of both methods with a relevant improvement of accuracy (Suárez and Montoyo, 2001). Our intention is to extent this combination with the help of other well known WSD methods and to establish a voting method or some other manner of cooperation.

Our main objective is to develop a complete WSD system in order to help other NLP activities in our research group. The work presented here is our first attempt to participate at Senseval and we hope to get the proper conclusions in order to improve our system and compete in the next Senseval.

References

Gerard Escudero, Lluis Màrquez, and German Rigau. 2000. Boosting applied to word sense disambiguation. In *Proceedings of the 12th Conference on Machine Learning ECML2000*, Barcelona, Spain.

A. Montoyo and M. Palomar. 2000. Word Sense Disambiguation with Specification Marks in Unrestricted Texts. pages 103–107.

A. Montoyo and M. Palomar. 2001. Specification Marks for Word Sense Disambiguation: New Development. In *Proceedings of 2nd International conference on Intelligent Text Processing and Computational Linguistics (CICLing-2001)*, pages 182–191.

A. Montoyo, M. Palomar, and G. Rigau. 2001. WordNet Enrichment with Classification Systems. In ACL, editor, *Proceedings of NAACL Workshop WordNet and Other Lexical Resources: Applications, Extensions and Customizations*, Pittsburgh, PA, USA.

Hwee Tou Ng and Hian Beng Lee. 1996. Integrating multiple knowledge sources to disambiguate word senses: An exemplar-based approach. In *Proceedings 34th Annual Meeting of the ACL-1996.*, San Francisco, USA.

Adwait Ratnaparkhi. 1998. *Maximum Entropy Models for Natural Language Ambiguity Resolution.* Ph.D. thesis, University of Pennsylvania.

Maximiliano Saiz-Noeda, Armando Suárez, and Manuel Palomar. 2001. Semantic pattern learning through maximum entropy-based wsd technique. In *Proceedings of CoNLL-2001*, pages 23–29. Toulouse, France.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings International Conference on New Methods in Language Processing.*, pages 44–49, Manchester, UK.

Armando Suárez and Andrés Montoyo. 2001. Estudio de cooperación entre métodos de desambiguación léxica: Marcas de especificidad vs. máxima entropía. *Procesamiento Lenguaje Natural*, 27(1):207–214, september.

Pasi Tapanainen and Timo Järvinen. A non-projective dependency parser. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 64–71.

Japanese word sense disambiguation using the simple Bayes and support vector machine methods

Masaki Murata, Masao Utiyama, Kiyotaka Uchimoto, Qing Ma, and **Hitoshi Isahara**
Communications Research Laboratory
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0289, Japan

Abstract

We submitted four systems to the Japanese dictionary-based lexical-sample task of SENSEVAL-2. They were i) the support vector machine method ii) the simple Bayes method, iii) a method combining the two, and iv) a method combining two kinds of each. The combined methods obtained the best precision among the submitted systems. After the contest, we tuned the parameter used in the simple Bayes method, and it obtained higher precision. An explanation of these systems used in Japanese word sense disambiguation was provided.

1 Introduction

We participated in the Japanese dictionary-based lexical-sample task of the SENSEVAL-2 contest. We used machine learning approaches and submitted four systems. After the contest, we tuned the parameter used in the simple Bayes method and carried out additional experiments. In this paper, we explain the systems and their experimental results.

2 Task Descriptions

The test data included 10,000 instances for evaluation. The RWC corpus (Shirai et al., 2001) was given as the training data. It was made from 3000 articles published in the Mainichi Newspaper. The nouns, verbs, and adjectives (the total number of which was about 150,000) were assigned sense tags defined on the basis of the Iwanami dictionary. The purpose of this task was to estimate the sense of a word by using its context.

3 Methods

Because the word sense assigned to each word is dependent on the word itself, estimations were conducted using machine learning methods for each word. That is, we constructed as many learning machines as there were individual words.

We used the simple Bayes and support vector machine methods as the machine learning method.[1] In this section, we explain each of the machine learning methods and then explain the method combining several of them.

3.1 Simple Bayes Method

This method estimates probability based on the Bayes theory. The category (i.e., the sense tag) with the highest probability is judged to be the desired one. This is a basic approach to the disambiguation of word sense. The probability of category a appearing in context b is defined as:

$$p(a|b) = \frac{p(a)}{p(b)} p(b|a) \quad (1)$$

$$\simeq \frac{\tilde{p}(a)}{p(b)} \prod_i \tilde{p}(f_i|a), \quad (2)$$

where context b is a set of features $f_j (\in F, 1 \leq j \leq k)$ that is defined in advance. $p(b)$ is the probability of context b, which is not calculated because it is a constant and is not dependent on category a. $\tilde{p}(a)$ and $\tilde{p}(f_i|a)$ are the probabilities estimated by using the training data and indicate the probability of the occurrence of category a in the examples of the training data and the probability of feature f_i occurring, given category a, respectively. When we use the maximum likelihood estimation to calculate $\tilde{p}(f_i|a)$, which often has a value of 0 and is therefore difficult to estimate the desired category, smoothing process is used. We used this

[1] We made preliminary experiments using various methods: the simple Bayes, the decision list, the maximum entropy, and the support vector machine. The results showed that the simple Bayes and support vector machine methods were better than the other two (Murata et al., 2001). We used these two methods in the contest.

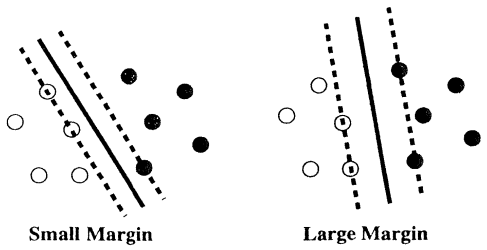

Figure 1: Maximizing the margin

equation for smoothing:

$$\tilde{p}(f_i|a) = \frac{freq(f_i, a) + \epsilon * freq(a)}{freq(a) + \epsilon * freq(a)}, \quad (3)$$

where $freq(f_i, a)$ is the number of events that have the feature f_i and whose category is a and $freq(a)$ is the number of events whose category is a. ϵ is a constant set by experimentation. In this study, we used 0.01 and 0.0001 as ϵ.[2]

3.2 Support Vector Machine Method

In this method, data consisting of two categories is classified by using a hyperplane to divide a space. When the two categories are, for example, positive and negative, enlarging the margin between the positive and negative examples in the training data (see Figure 1[3]) reduces the possibility of incorrectly choosing categories in test data. The hyperplane that maximizes the margin is thus determined, and classification is carried out using that hyperplane. Although the basics of this method are the same as those described above, in the extended versions of the method, the region between the margins through the training data can include a small number of examples, and the linearity of the hyperplane can be changed to a non-linearity by using kernel functions. The classification in the extended versions is equivalent to the classification using the following function (Equation (4)), and the two categories can be classified on the basis of whether the value output by the function is positive or negative (Cristianini and Shawe-Taylor, 2000; Kudoh, 2000):

$$f(\mathbf{x}) = sgn\left(\sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right) \quad (4)$$

$$b = \frac{max_{i,y_i=-1} b_i + min_{i,y_i=1} b_i}{2}$$

$$b_i = -\sum_{j=1}^{l} \alpha_j y_j K(\mathbf{x}_j, \mathbf{x}_i),$$

where $\mathbf{x}$ is the context (a set of features) of an input example, $\mathbf{x}_i$ indicates the context of a training datum, y_i ($i = 1, ..., l, y_i \in \{1, -1\}$) indicates its category, and the function sgn is

$$sgn(x) = \begin{array}{ll} 1 & (x \geq 0), \\ -1 & (otherwise). \end{array} \quad (5)$$

Each α_i ($i = 1, 2...$) is fixed as the value of α_i that maximizes the value of $L(\alpha)$ in Equation (6) under the conditions set by Equations (7) and (8).

$$L(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j K(\mathbf{x_i}, \mathbf{x_j}) \quad (6)$$

$$0 \leq \alpha_i \leq C \ (i = 1, ..., l) \quad (7)$$

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \quad (8)$$

Function K is called a kernel function and various functions are used as kernel functions. We have used the following polynomial function exclusively.

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^d \quad (9)$$

C and d are constants set by experimentation. For all of the experiments reported in this paper, C was fixed as 1 and d was fixed as 2.

A set of $\mathbf{x}_i$ that satisfies $\alpha_i > 0$ is called a support vector (SV_s)[4]. The summation portion of Equation (4) was calculated using only the examples that were support vectors.

Support vector machine methods are capable of handling data consisting of two categories. In general, data consisting of more than two categories is handled by using the pair-wise method (Kudoh and Matsumoto, 2000).

In this method, for data consisting of N categories, pairs of two different categories (N(N-1)/2 pairs) are constructed. The better cate-

[2] In the SENSEVAL-2 contest, we used 0.01 as ϵ. After the contest, we tested several values (0.1 to 0.00000001) as ϵ. We confirmed that $\epsilon = 0.0001$ produced the best results using 10-fold cross validation in the training data.

[3] In the figure, the white and black circles indicate positive and negative examples, respectively. The solid line indicates the hyperplane that divides the space, and the broken lines indicate the planes that mark the margins.

[4] In Figure 1, the circles in the broken lines indicate support vectors.

gory is determined by using a 2-category classifier (in this paper, a support vector machine[5] was used as the 2-category classifier), and the correct category is finally determined by "voting" on the N(N-1)/2 pairs that result from analysis using the 2-category classifier.

The support vector machine method is, in fact, performed by combining the support vector machine and pair-wise methods described above.

3.3 Combined Method

Our combined method changed the used machine-learning method for each word. The used method for each word was the best one for the word in the 10-fold cross validation[6] on the training data among the given methods for combination.

We used the following three kinds of combinations.

- Combined method 1
  a combination of the simple Bayes and support vector machine methods
- Combined method 2
  a combination of two kinds of the simple Bayes method and two kinds of the support vector machine method
  (Here, "the two kinds" indicate an instance where all features were used and where the syntactic feature alone were not).[7]
- Combined method 3
  a combination of two kinds of the simple Bayes method
  (Here, "the two kinds" indicate instance where $\epsilon = 0.0001$ and another where $\epsilon = 0.01$).

4 Features (information used in classification)

In this paper, the following are defined as features.

- **Features based on strings**
  - strings in the analyzed morpheme
  - strings of 1 to 3-grams just before the analyzed morpheme
  - strings of 1 to 3-grams just after the analyzed morpheme
- **Features based on the morphological information given by the RWC tags**
  - the part of speech (POS), the minor POS, and the more minor POS of the analyzed morpheme [8]
  - the previous morpheme, its 5-digit category number, its 3-digit category number, its POS, its minor POS, and its more minor POS[9]
  - the next morpheme, its 5-digit category number, its 3-digit category number, its POS, its minor POS, and its more minor POS
- **Features based on the morphological information given by JUMAN**
  The corpus was analyzed using the Japanese morphological analyzer, JUMAN (Kurohashi and Nagao, 1998), and the results were used as features.
  - the POS, the minor POS, and the more minor POS of the analyzed morpheme, which were determined from the results of JUMAN.
  - the previous morpheme, its 5-digit category number, its 3-digit category number, its POS, its minor POS, and its more minor POS
  - the next morpheme, its 5-digit category number, its 3-digit category number, its POS, its minor POS, and its more minor POS
- **Features based on syntactic information**
  The corpus was analyzed using the Japanese syntactic analyzer KNP (Kurohashi, 1998), and the results were used as features.
  - the *bunsetsu*,[10] including the analyzed morpheme information on whether or not

[5] We used Kudoh's TinySVM software (Kudoh, 2000) as the support vector machine.

[6] In the 10-fold cross validation, we first divide the training data into ten parts. The answers of the instances in each part are estimated by using the instances in the remaining nine parts as the training data. We then use all the results in the ten parts for evaluation.

[7] We used a case where the syntactic feature alone was not used because it obtained a higher precision than when all the features had been used in our preliminary experiments.

[8] The POS, the minor POS, and the more minor POS of a morpheme are the items in the third, fourth, and fifth fields of the RWC corpus, respectively.

[9] A Japanese thesaurus, the *Bunrui Goi Hyou* dictionary (NLRI, 1964), was used to determine the category number of each morpheme. This thesaurus is of the 'is-a' hierarchical type, in which each word has *a category number*, which is a 10-digit number that indicates seven levels of an 'is-a' hierarchy. The top five levels are expressed by the first five digits, the sixth level is expressed by the next two digits, and the final level is expressed by the final three digits.

[10] *Bunsetsu* is a Japanese grammatical term. A bunsetsu is similar to a phrase in English, but is a slightly smaller component. *Eki-de* "at the station" is a bunsetsu, and *sono*, which corresponds to "the" or "its," is also a bunsetsu. A bunsetsu is, roughly, a unit of items that refers to entities.

Table 1: Experimental results

Method	Precision
Baseline method	0.726
Support vector machine (CRL1)	0.783
Simple Bayes method, $\epsilon = 0.01$ (CRL2)	0.778
Simple Bayes method, $\epsilon = 0.0001$	0.790
Combined method 1 (CRL3)	0.786
Combined method 2 (CRL4)	0.786
Combined method 3	0.793
The best method in the contest	0.786

the bunsetsu was a noun phrase, the POS of the bunsetsu's particle, the minor POS of the particle, and the more minor POS of the particle

- the main word that the bunsetsu modifies, including the analyzed morpheme and its 5-digit category number, 3-digit category number, POS, minor POS, and more minor POS
- the main words of the modifiers of the bunsetsu including the analyzed morpheme and their 5-digit category numbers, 3-digit category numbers, POSs, minor POSs, and more minor POSs (In this case, the information on the particle, such as *ga* or *o*, was used as well).

- **Features of all words co-occurring in the same sentence**
  The corpus was analyzed using the Japanese morphological analyzer JUMAN (Kurohashi and Nagao, 1998), and lists of the results were used as features.
  - each morphology in the same sentence, its 5-digit category number, and its 3-digit category number

- **Features of the UDC code in a document**
  In the RWC corpus, each document has a universal decimal code (UDC), indicating its category.
  - the first digit, the first two-digits, and the first three-digits of the UDC in the document

5 Experiments

We submitted the four systems (CRL1 to CRL4), the support vector machine method, the simple Bayes method ($\epsilon = 0.01$), Combined method 1, and Combined method 2. After the contest, we carried out the experiments using the simple Bayes ($\epsilon = 0.0001$) and Combined method 3. Their experimental results are shown in Table 1. "Baseline method" selected the category that most frequently occurred in the training data as the answer. "The best method in the contest" was the best among all the systems submitted to the contest, which was CRL4 (0.786483). The precisions shown in the table are the mixed-grained scores calculated by software "scorer2", which was given by the committees of SENSEVAL-2. (In our systems, all the instances were attempted, so the recall rate was equal to its precision rate.)

We found the following items from the results.

- All the methods produced higher precision than the baseline method.
- Among the four submitted systems (CRL1 to CRL4), Combined method 2 was the best.
- The simple Bayes method using $\epsilon = 0.0001$ and Combined method 3 (the combination of the two simple Bayes methods) obtained higher precision. This indicates that the simple Bayes method was effective.

6 Conclusion

Our methods combining the simple Bayes and support vector machine methods obtained the best precision among all the submitted systems. After the contest, we tuned the parameter used in the simple Bayes method using the 10-fold cross validation in the training data, and it obtained higher precision. The best method was the combination of the two simple Bayes, whose precision was 0.793.

References

Nello Cristianini and John Shawe-Taylor. 2000. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods.* Cambridge University Press.

Taku Kudoh and Yuji Matsumoto. 2000. Use of support vector learning for chunk identification. *CoNLL-2000.*

Taku Kudoh. 2000. TinySVM: Support Vector Machines. http://cl.aist-nara.ac.jp/ taku-ku// software/TinySVM/ index.html.

Sadao Kurohashi and Makoto Nagao, 1998. *Japanese Morphological Analysis System JUMAN version 3.5.* Department of Informatics, Kyoto University. (in Japanese).

Sadao Kurohashi, 1998. *Japanese Dependency/Case Structure Analyzer KNP version 2.0b6.* Department of Informatics, Kyoto University. (in Japanese).

Masaki Murata, Masao Utiyama, Kiyotaka Uchimoto, Qing Ma, and Hitoshi Isahara. 2001. Experiments on word sense disambiguation using several machine-learning methods. In *IEICE-WGNLC2001-2.* (in Japanese).

NLRI. 1964. *Bunrui Goi Hyou.* Shuuei Publishing.

Kiyoaki Shirai, Wakako Kashino, Minako Hashimoto, Takenobu Tokunaga, Eiichi Arita, Hitoshi Isahara, Shiho Ogino, Ryuichi Kobune, HIronobu Takahashi, Katashi Nagao, Kôiti Hasida, and Masaki Murata. 2001. Text database with word sense tags defined by Iwanami Japanese dictionary. *Information Processing Society of Japan, WGNL 141-19.* (in Japanese).

Machine Learning with Lexical Features: The Duluth Approach to Senseval-2

Ted Pedersen
Department of Computer Science
University of Minnesota, Duluth
Duluth, MN 55812 USA
tpederse@d.umn.edu

Abstract

This paper describes the sixteen Duluth entries in the SENSEVAL-2 comparative exercise among word sense disambiguation systems. There were eight pairs of Duluth systems entered in the Spanish and English lexical sample tasks. These are all based on standard machine learning algorithms that induce classifiers from sense-tagged training text where the context in which ambiguous words occur are represented by simple lexical features. These are highly portable, robust methods that can serve as a foundation for more tailored approaches.

1 Introduction

The Duluth systems in SENSEVAL-2 take a supervised learning approach to the Spanish and English lexical sample tasks. They learn decision trees and Naive Bayesian classifiers from sense-tagged training examples where the context in which an ambiguous word occurs is represented by lexical features. These include unigrams and bigrams that occur anywhere in the context, and co-occurrences within just a few words of the target word. These are the only types of features used. There are no syntactic features, nor is the structure or content of WordNet employed. As a result these systems are highly portable, and can serve as a foundation for systems that are tailored to particular languages and sense inventories.

The word sense disambiguation literature provides ample evidence that many different kinds of features contribute to the resolution of word meaning. These include part–of–speech, morphology, verb–object relationships, selectional restrictions, lexical features, etc. When used in combination it is often unclear to what degree each type of feature contributes to overall performance. It is also unclear to what extent adding new features allows for the disambiguation of previously unresolvable test instances. One of the long term objectives of our research is to determine which types of features are complementary and cover increasing numbers of test instances as they are added to a representation of context.

2 Experimental Methodology

The training and test data for the English and Spanish lexical sample tasks is split into separate training and test files per word. A supervised learning algorithm induces a classifier from the training examples for a word, which is then used to assign sense tags to the test instances for that word.

The context in which an ambiguous word occurs is represented by lexical features that are identified using the Bigram Statistics Package (BSP) version 0.4. This is free software that extracts unigrams and bigrams from text using a variety of statistical methods. Each unigram or bigram that is identified in the training data is treated as a binary feature that indicates whether or not it occurs in the context of the word being disambiguated. The free software package SenseTools (version 0.1) converts training and test data into a feature vector representation, based on the output from BSP. This becomes the input to the Weka suite of supervised learning algorithms. Weka induces classifiers from the training examples and applies the sense tags to the test instances.

The same software is used for the English and Spanish text. BSP and SenseTools are written in Perl and are freely available from www.d.umn.edu/~tpederse/code.html. Weka is written in Java and is freely available from www.cs.waikato.ac.nz/~ml.

3 System Descriptions

There were eight pairs of Duluth systems in the English and Spanish lexical sample tasks. The only language dependent components are the tokenizers and stop–lists. For both English and Spanish a stop–list is made up of all words that occur ten or more times in five randomly selected word training files of comparable size. All Duluth systems exclude the words in the stop–list from being features.

Each pair of systems is summarized below. All performance results are based on accuracy (correct/total) using fine-grained scoring. The name of the English system appears first, followed by the Spanish system.

Duluth1/Duluth6 create an ensemble of three Naive Bayesian classifiers, where each is based on a different set of features. The hope is that these different views of the training examples will result in classifiers that make complementary errors, and that their combined performance will be better than any of the individual classifiers.

Separate Naive Bayesian classifiers are learned from each representation of the training examples. Each classifier assigns probabilities to each of the possible senses of a test instance. These are summed and the sense with the largest value is used. This technique is used in many of our ensembles and will be referred to as a weighted vote.

The first feature set is made up of bigrams, i.e., consecutive two word sequences, that can occur anywhere in the context with the ambiguous word. To be selected as a feature, a bigram must occur two or more times in the training examples and have a log-likelihood ratio (G^2) value ≥ 6.635, which is associated with a p-value of .01.

The second feature set is based on unigrams, i.e., one word sequences, that occur five or more times in the training data.

The third feature set is made up of co-occurrence features that represent words that occur on the immediate left or right of the target word. In effect, these are bigrams that include the target word. They must also occur two or more times and have a log-likelihood ratio ≥ 2.706, which is associated with a p-value of .10.

These systems are inspired by (Pedersen, 2000), which presents an ensemble of eighty-one Naive Bayesian classifiers based on varying sized windows of context to the left and right of the target word that define co-occurrence features. However, the current systems only use a three member ensemble to capture the spirit of simplicity and portability that underlies the Duluth approach to SENSEVAL-2.

English accuracy was 53%, Spanish was 58%.

Duluth2/Duluth7 learn an ensemble of decision trees via bagging. Ten samples are drawn, with replacement, from the training examples for a word. A decision tree is learned from each of these permutations of the training examples, and each of these trees becomes a member of the ensemble. A test instance is assigned a sense based on a weighted vote among the members of the ensemble. In general decision tree learning can be overly influenced by a small percentage of the training examples, so the goal of bagging is to smooth out this instability.

There is only one kind of feature used in these systems, bigrams that occur two or more times and have a log-likelihood ratio ≥ 6.635. This is one of the three feature sets used in the Duluth1/Duluth6 systems.

The set of bigrams that meet these criteria become candidate features for the J48 decision tree learning algorithm, which is the Weka implementation of the C4.5 algorithm. The decision tree learner first constructs a tree of features that characterizes the training data exactly, and then prunes features away to avoid over–fitting and allow it to generalize to the previously unseen test instances. Thus, a decision tree learner performs a second cycle of feature selection and is not likely to use all of the features that we identify prior to learning with BSP. The default C4.5 parameter settings are used for pruning.

These systems are an extension of (Pedersen, 2001), which learns a single decision tree where the representation of context is based on bigrams. This earlier work does not use bagging, and the top 100 bigrams according to the log-likelihood ratio are the candidate features.

English accuracy was 54%, Spanish was 60%.

Duluth3/Duluth8 rely on the same features as Duluth1/Duluth6, but learn an ensemble of three bagged decision trees instead of an ensemble of Naive Bayesian classifiers.

There is a strong contrast between these techniques, since decision tree learners attempt to characterize the training examples and find relationships among the features, while a Naive Bayesian classifier is based on an assumption of conditional independence among the features.

The feature set used in these systems is from Duluth1/Duluth6 and consists of bigrams, unigrams and co-occurrences. A bagged decision tree is learned for each of the three kinds of features. The test instances are classified by each of the bagged decision trees, and a majority vote is taken among the members to assign senses to the test instances.

These are the most accurate of the Duluth systems for both English (57%) and Spanish (61%). These are within 7% of the most accurate overall approaches for English (64%) and Spanish (68%).

Duluth4/Duluth9 uses a Naive Bayesian classifier based on a bag of words representation of context, where each unigram that occurs in the training data is taken as a feature. This is a common benchmark in word sense disambiguation studies and text classification problems.

In the English training examples any word that occurs five or more times is used as a feature, and in the Spanish data any word that occurs two or more times is used. These features are used to estimate the parameters of a Naive Bayesian classifier. This will assign the most probable sense to a test instance, given the surrounding context.

Accuracy for English was 54%, and for Spanish 56%. This Naive Bayesian classifier was one of the three member classifiers in the ensemble approach of Duluth1/Duluth7, which was 1% less accurate for English and and 2% more accurate for Spanish.

Duluth5/Duluth10 add a co–occurrence feature to the Duluth2/Duluth7 systems. In every other respect they are identical. The co-occurrence feature was also used in Duluth1/Duluth6, and is essentially a bigram where one of the words is the ambiguous word. These must occur two or more times in the training examples and have a log-likelihood ratio ≥ 2.706 to be included as a feature. In addition to the co-occurrence feature the bigram feature from Duluth2/Duluth7 is used, where a bigram must occur two or more times and have a log-likelihood ratio ≥ 6.635.

Accuracy for English was 55%, and for Spanish 61%. This was a slight improvement over Duluth2 (54%) and Duluth7 (60%).

DuluthA/DuluthX build an ensemble of three different classifiers that are induced from the same representation of the training examples. A weighted vote is taken to assign senses to test instances. The three classifiers are a bagged J48 decision tree, a Naive Bayesian classifier, and the nearest neighbor classifier IBk, where the number of neighbors parameter k is set to 1.

The context in which the ambiguous word occurs is represented by bigrams that may include zero, one, or two intervening words that are ignored. To be considered as features these bigrams must occur two or more times and have a log-likelihood ratio ≥ 10.827, i.e., a p-value of .001. The log-likelihood ratio threshold is set to 0 for the Spanish data due to the smaller volume of data.

English accuracy was 52%, Spanish was 58%.

DuluthB/DuluthY are identical to Duluth5/Duluth10, except that rather than learning an entire decision tree they stop the learning process once the root of the decision tree is selected. The resulting one node decision tree is called a decision stump. At worst a decision stump will reproduce the most common sense baseline, and may do better if the selected feature is particularly informative. In previous work we have observed that decision stumps can serve as a very aggressive lower bound on performance (Pedersen, 2001).

Decision stumps are the least accurate method for both English (DuluthB, 51%) and Spanish (DuluthY, 52%), but are more accurate than the most common sense baseline for English (48%) and Spanish (47%).

DuluthC/DuluthZ take a kitchen sink approach to ensemble creation, and combine the seven systems for English and Spanish into ensembles that assign senses to test instances by taking a weighted vote among the members.

Accuracy for English was 55%, and for Spanish 59%. This is less than the accuracy of some of the members systems, suggesting that the members of the ensemble are making redundant errors.

4 Discussion

There are several hypotheses that underly and motivate these systems.

4.1 Features Matter Most

This hypothesis is at the core of much of our recent work. It holds that variations in learning algorithms matter far less to disambiguation performance than do variations in the features used to represent the context in which an ambiguous word occurs. In other words, an informative feature set will result in accurate disambiguation when used with a wide range of learning algorithms, but there is no learning algorithm that can overcome the limitations of an uninformative or misleading set of features.

There are a number of demonstrations that can be made from the Duluth systems in support of this hypothesis, but perhaps the clearest is found in comparing the systems Duluth1/Duluth6 and Duluth3/Duluth8. The first pair learns three Naive Bayesian classifiers and the second learns three bagged decision trees. Both use the same feature set to represent the context in which ambiguous words occur. There is a 3% improvement in accuracy when using the decision trees. We believe this modest improvement when moving from a simple learning algorithm to a more complex one supports the hypothesis that the true dividends are to be found in improving the feature set.

4.2 50/25/25 Rule

We hypothesize a 50/25/25 rule for supervised approaches to word sense disambiguation. This loosely holds that given a classifier learned from a sample of sense–tagged training examples, about half of the test instances are easily disambiguated, a quarter are harder but still possible, and the remaining quarter are extremely difficult. This is a minor variant of the 80/20 rule of time management, which holds that 20% of effort accounts for 80% of results.

When the two highest ranking systems in the official English lexical sample results are compared there are 2180 test instances (50%) that both disambiguate correctly using fine-grained scoring. There are an additional 1183 instances (28%) where one of the two systems are correct, and 965 instances (22%) that neither system can resolve. If these two systems were optimally combined, their accuracy would be 78%. If the third-place system is also considered, there are 1939 instances (44.8%) that all three systems can disambiguate, and 816 (19%) that none could resolve.

For all the Duluth systems for English, there are 1705 instances (39%) that all eight systems got correct. There are 1299 instances (30%) that none can resolve. The accuracy of an optimally combined system would be 70%. The most accurate individual system is Duluth3 with 57% accuracy.

For the Spanish Duluth systems, there are 856 instances (38%) that all eight systems got correct. There are 478 instances (21%) that none of the systems got correct. This results in an optimally combined result of 79%. The most accurate Duluth system was Duluth8, with 1369 correct instances (62%). If the top ranked Spanish system (68%) and Duluth8 are compared, there are 1086 instances (49%) where both are correct, 737 instances (33%) where one or the other is correct, and 402 instances (18%) where neither system is correct.

This is intended as a rule of thumb, and suggests that a fairly substantial percentage of test instances can be resolved by almost any means, and that a hard core of test instances will be very difficult for any method to resolve.

5 Acknowledgments

This work has been partially supported by a National Science Foundation Faculty Early CAREER Development award (#0092784).

The Bigram Statistics Package and SenseTools have been implemented by Satanjeev Banerjee.

References

T. Pedersen. 2000. A simple approach to building ensembles of Naive Bayesian classifiers for word sense disambiguation. In *Proceedings of the First Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 63–69, Seattle, WA, May.

T. Pedersen. 2001. A decision tree of bigrams is an accurate predictor of word sense. In *Proceedings of the Second Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 79–86, Pittsburgh, July.

Anaphora Resolution with Word Sense Disambiguation

Judita Preiss*
Computer Laboratory
JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
Judita.Preiss@cl.cam.ac.uk

Abstract

We describe a simple word sense disambiguation system equipped with the Kennedy and Boguraev (1996) anaphora resolution algorithm, evaluated on the SENSEVAL-2 English all-words task. The system relies on the structure of the WordNet hierarchy to pick optimal senses for nouns in the text. Since anaphoric references are known to indicate the topic of the text (Boguraev et al., 1998), they may aid disambiguation.

1 Introduction

We investigate the effect of repeating pronominalized nouns in the input to our Word Sense Disambiguation (WSD) algorithm (Preiss, 2001). The WSD algorithm is based on the WordNet 1.7 hierarchy (Miller et al., 1990), and assigns (WordNet) senses to all nouns. The enriched version we evaluate in this paper makes use of our re-implementation of an anaphora resolution algorithm of Kennedy and Boguraev (1996).

If, as claimed by Boguraev et al. (1998), the topic of the discourse is thus repeated, then the main topic words will be more likely to be disambiguated correctly. The subsequent WSD algorithm makes use of this extra topic information, and this will in turn affect the disambiguation of all other nouns in the discourse.

The system is evaluated on the English all-words task in SENSEVAL-2.

2 Algorithms

2.1 Overview of the Algorithm

Our WSD algorithm has three components, as depicted in Figure 1. Taking as input the test data parsed using the Briscoe and Carroll (1993) parser (which uses the grammar described in Carroll and Briscoe (1996)), the first step is to identify and discard the pleonastic pronouns. Our pleonastic component is described in section 2.2.

In the next phase (section 2.3), third person pronouns are resolved to a noun antecedent and replaced in the text by the noun antecedent. The purpose of this is to increase the number of topic words in the text, to aid the disambiguation of other nouns. This approach assumes firstly that pronouns refer mainly to topic words, and secondly that repeating topic words in the text helps overall disambiguation.

The final phase of the algorithm is the WSD component, described in section 2.4. Using simulated annealing, it attempts to find a sense assignment for every noun that minimizes an overall 'distance' function using the WordNet hierarchy. In addition, for the repeated nouns added in the previous phase, the senses are tied together. This means that if the sense of one word in a tie is changed during simulated annealing, the sense of all words in the tie are simultaneously changed.

The advantage of this approach can be shown on the following discourse: *The parrot, like the chicken, is kept by people as a domesticated bird. It can speak.* Suppose firstly that there is no anaphora resolution phase. The words *parrot, chicken, person, bird* are given to the word sense disambiguation algorithm, and the system chooses senses which are related to people (*parrot* in the sense of mimicking people, *chicken* a wimp and so on). This is clearly incorrect. Now suppose we resolve the pronoun *it* to *parrot*, and repeat the word *parrot* in the text. Now the words *parrot, chicken, person, bird, parrot* are passed to the WSD system (where the two

* This work was supported by the EPSRC while the author was at the University of Sheffield.

parrots are sense-tied together), and the system now chooses the correct bird-related senses.

2.2 Pleonastic Pronouns Component

It can be a pleonastic pronoun (pronoun with no antecedent), for example in the sentence: *It is raining.* We label the pronoun *it* as pleonastic if it is a subject of a raising verb (these were extracted from the ANLT lexicon (Boguraev and Briscoe, 1987)) or if it was used in conjunctions with the verb *to be* and one of a particular set of adjectives (for example *It is possible to go to town.*).

The component was evaluated on a manually anaphorically resolved portion of the BNC (the initial 2000 sentences of w01). It has a precision (proportion of pronouns deemed pleonastic which really are pleonastic) of 94% and recall (proportion of pleonastic pronouns recognized as pleonastic) of 61%.

2.3 Anaphora Resolution Component

The pronominal anaphora resolution is carried out by our re-implementation of the Kennedy and Boguraev (Kennedy and Boguraev, 1996) anaphora resolution algorithm. This algorithm is based on that of Lappin and Leass (Lappin and Leass, 1994), but does not require a full parse. It treats the cases of third person pronouns and lexical anaphors.[1] Its cited accuracy is 75% on general corpora (Kennedy and Boguraev, 1996), but note that their published algorithm uses the LINGSOFT morphosyntactic tagger.

The algorithm creates coreference classes which join together words which are believed by the algorithm to be referring to the same object. These classes are assigned a salience value based on the presence of the features in Table 1. The salience value of a class is the sum of the feature weights of its members, scaled down by the number of sentences ago that the feature last occurred. The correct antecedent is chosen to be the closest word from the coreference class with the highest salience.

2.4 WSD Component

We define a notion of distance between any two WordNet noun senses which is based on the WordNet hierarchy.[2] As pointed out by Resnik (1999), it is naive to assume that the distance between any two nodes in the hierarchy is equal. We therefore assign a weight w to every noun sense x:

$$\text{weight}(x) = \frac{\text{number of children below } x \text{ in hierarchy}}{\text{total nodes in hierarchy}}$$

This is used to define the distance between two distinct noun senses x and y:

$$\text{dist}(x,y) = \min_{z \in h(x) \cap h(y)} \text{weight}(z) - \tfrac{1}{2}\text{weight}(x) - \tfrac{1}{2}\text{weight}(y)$$

where $h(s)$ denotes the hypernym chain of noun sense s.[3] If the hypernym chains of x and y do not intersect, the distance is set to the maximum value of 1. In Preiss (2001), we investigated scaling the distance function such that for noun senses x and y at positions in the corpus n and m respectively:

$$\text{dist}^*(x,y) = \frac{\text{dist}(x,y)}{|n-m|^\alpha}$$

Note that we do not explicitly use a window of surrounding nouns, but the $|n-m|$ denominator means that contributions from far away nouns are usually negligible. We showed that it was not possible to guess the optimal value of α

Condition	Weight
Current sentence	100
Current context	50
Subject	80
Existential construct	70
Possessive	65
Direct object	50
Indirect object	40
Oblique	30
Non embedded	80
Non adjunct	50

Table 1: Salience values

[1] Lexical anaphors are reflexives and reciprocals.

[2] In the SENSEVAL-2 task we identify nouns by using an enhanced version of the GATE tagger and lemmatizer (Cunningham et al., 1995).

[3] The hypernym chain of s consists of the word s, the parent word of s, the grandparent of s, etc, all the way to a root word.

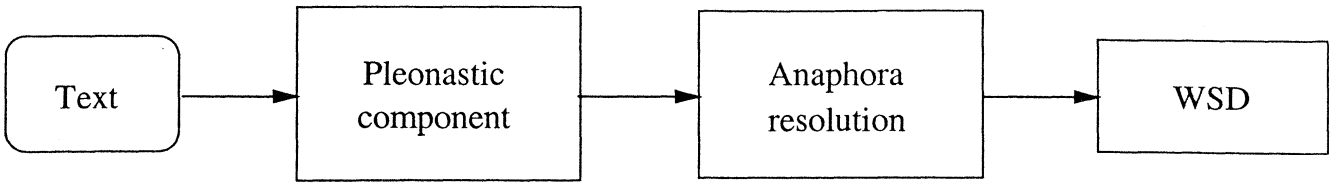

Figure 1: Integration of components

in advance for any set of texts covered in SEMCOR. However, averaged over all words there is a slight peak around $\alpha = 1$, so this is the value we take.

The distance between two adjacent nodes in the hierarchy may now not be equal. To illustrate this, consider the following example adapted from a paper of Resnik (1999). In WordNet 1.7 (prerelease), VALVE is the parent node of SAFETY VALVE, and MACHINE is the parent of INFORMATION PROCESSING SYSTEM. However, the intuitive distance between the first pair of nodes seems to be less than the distance between the second pair. Using our distance function outlined above, the distance between SAFETY VALVE and VALVE is 0.000121, while the distance between INFORMATION PROCESSING SYSTEM and MACHINE is 0.00229. This is depicted in Figure 2.

We want to assign precisely one sense to each noun in the text; we call this a path. We find the 'optimal' path by simulated annealing (Bertsimas and Tsitsiklis, 1992). Simulated annealing is a probabilistic method for finding the global optimum of a function which may have a number of local optima. We define the function to be minimized, the energy function, to be the sum of all the pairwise scaled distances.

Our version of simulated annealing starts with a randomly chosen path which it attempts to improve. It performs a number of iterations in which it randomly chooses a word and then chooses a new sense for this word.[4] If this change is an improvement in terms of the energy function, it is kept. Otherwise, it may or may not be accepted depending on the current value of the temperature. Over time the temperature decreases, making it less likely to keep changes that increase the energy. The algorithm terminates when no changes were made in the last 1000 iterations.

When simulated annealing terminates, it outputs what it deems the optimal sense assignment for all the nouns in the text. For a more detailed description of the WSD algorithm, please refer to Preiss (2001).

This algorithm was implemented in C and executed on a Pentium III 500MHz. Each text took 1 hour to initialize, and 2 hours to perform 20 runs of simulated annealing. A majority vote then decided the sense assignment.

Article	Words	Senses	Ties
1	363	1698	38
2	575	2098	46
3	340	1495	60

Table 2: Test data for the English all words task

3 Results

The WSD component enhanced with the anaphora resolution algorithm was submitted for the English all-words task in SENSEVAL-2. The test data for this task consisted of three articles, and information gathered from each article is displayed in Table 2. The words column shows the number of words marked as nouns by the part of speech tagger in the parser. The senses column contains the total number of senses for all of these words. The ties column shows the number of ties in the text, where each tie contains a noun and some pronouns that refer to it. The system achieved 44% precision and 20% recall fine-grained, and 45.2% precision and 20.5% recall coarse-grained.[5]

[4] We slightly skewed the probability distribution of the senses towards the more frequent sense. The probability of the nth sense is proportional to $\frac{1}{n^2}$.

[5] The system assigns senses to all nouns but to no other part of speech. It also has no mechanism for marking a word undecidable.

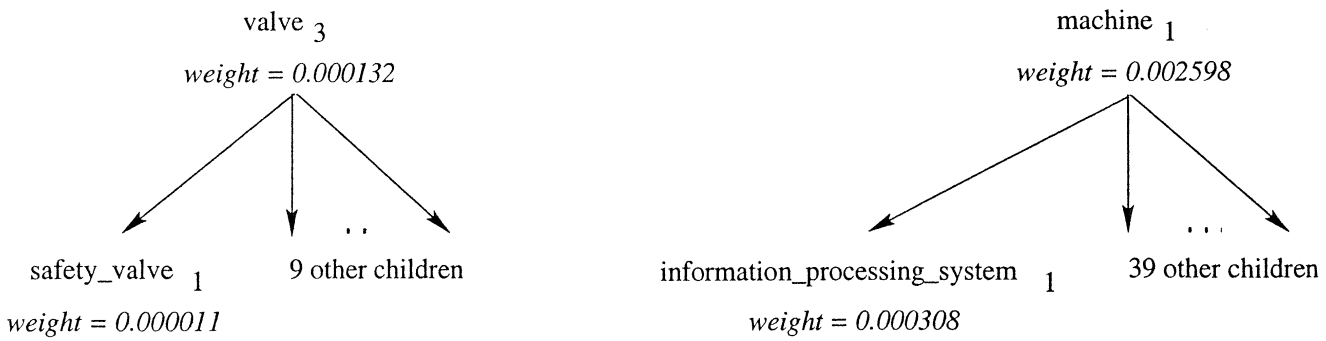

Figure 2: Distance between adjacent nodes

4 Future Work

We would like to investigate the performance of the WSD system with and without anaphora resolution, with a view to also extending links in text to other entities.

Although the precision of the pleonastic component is currently quite high, we intend to boost recall possibly by including some of the rules devised by Lappin and Leass (1994).

Acknowledgements

I would like to thank John Carroll for parsing the SENSEVAL-2 corpus for me.

References

D. Bertsimas and J. Tsitsiklis. 1992. Simulated annealing. In *Probability and Algorithms*, pages 17–29. National Academy Press, Washington, D. C. .

B.K. Boguraev and E.J. Briscoe. 1987. Large lexicons for natural language processing: utilising the grammar coding system of the *longman dictionary of contemporary english. Computational Linguistics*, 13(4):219–240.

B. Boguraev, C. Kennedy, R. Bellamy, S. Brawer, Y. Y. Wong, and J. Swartz. 1998. Dynamic presentation of document content for rapid on-line skimming. In *Proceedings of AAAI Spring Symposium on Intelligent Text Summarisation*, pages 118–128.

E. Briscoe and J. Carroll. 1993. Generalised probabilistic LR parsing of natural language (corpora) with unification-based grammars. *Computational Linguistics*, 19(1):25–60.

J. Carroll and T. Briscoe. 1996. Apportioning development effort in a probabilistic LR parsing system through evaluation. In *Proceedings of the ACL SIGDAT Conference on Empirical Methodsin Natural Language Processing*, pages 92–100.

H. Cunningham, R. Gaizauskas, and Y. Wilks. 1995. A general architecture for text engineering (GATE) — a new approach to language R&D. Technical Report CS–95–21, University of Sheffield.

C. Kennedy and B. Boguraev. 1996. Anaphora for everyone: Pronominal anaphora resolution without a parser. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING'96)*, pages 113–118.

S. Lappin and H. Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561.

G. Miller, R. Beckwith, C. Felbaum, D. Gross, and K. Miller. 1990. Introduction to WordNet: An on-line lexical database. *Journal of Lexicography*, 3(4):235–244.

J. Preiss. 2001. Local versus global context for WSD of nouns. In *Proceedings of CLUK 4*, pages 1–8.

P. Resnik. 1999. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130.

KUNLP system using Classification Information Model at SENSEVAL-2

Hee-Cheol Seo, Sang-Zoo Lee, Hae-Chang Rim
Dept. of Computer Science and Engineering,
Korea University
1, 5-ka, Anam-dong
Seongbuk-Gu, Seoul, 136-701, Korea
{hcseo,zoo,rim}@nlp.korea.ac.kr

Ho Lee
Astronest Inc.
135-090 3rd floor, Hanam BD
157-18 Samsung-Dong
Kangnam-Gu, Seoul, Korea
leeho@astronest.com

Abstract

The classification information model or CIM classifies instances by considering the discrimination ability of their features, which was proven to be useful for word sense disambiguation at SENSEVAL-1. But the CIM has a problem of information loss. KUNLP system at SENSEVAL-2 uses a modified version of the CIM for word sense disambiguation.

We used three types of features for word sense disambiguation: local, topical, and bigram context. Local and topical context are similar to Chodorow's context and refer to only unigram information. The window of a bigram context is similar to that of a local context but a bigram context refers to only bigram information.

We participated in the English lexical sample task and the Korean lexical sample task, where our systems ranked high.

1 Introduction

The classification information model(Ho, 1997) is the model that classifies instances by considering the discrimination ability of their features. In the CIM, a feature with high discrimination ability contributes to the classification more than one with low discrimination ability. Hence, we can omit the feature selection procedure.

The CIM has a kind of information loss problem due to the assumption that a feature contributes to only one class. We devised a modified version of the CIM where a feature can contribute to all classes.

Word sense disambiguation task can be treated as a kind of classification process(Ho, 2000). When a classification technique is applied to word sense disambiguation, an instance corresponds to a context containing a polysemous word and its class to the proper sense of the word, and one of its features to a piece of context information. As a classification problem, word sense disambiguation task can be solved by the CIM.

We used three types of features for word sense disambiguation: local, topical, and bigram context. Local and topical context are similar to Chodorow's context(Chodorow, 2000) and consist of only unigram information. A bigram context has a similar window to a local context but consists of only bigram information.

2 KUNLP system

To disambiguate senses, we did two phases: corpus preprocessing and sense disambiguation. Figure 1 shows the flow chart of our system.

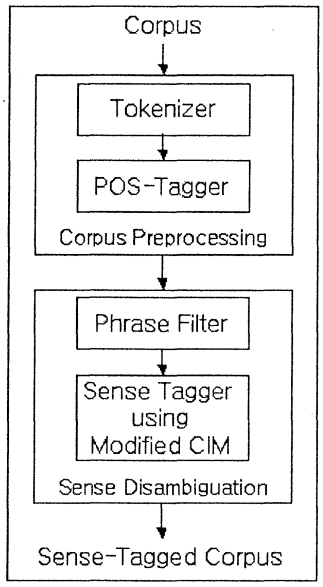

Figure 1: Flow chart of KUNLP system

2.1 Corpus preprocessing

At the corpus preprocessing phase, we tokenized a corpus and then tagged it with parts-of-speech using Brill's Tagger(Brill, 1994). The tokenizer just separates symbols from a word. For example, a sentence "*I'm straight, white, no longer middle class, anti-IRA, have ...*" is tokenized to "*I 'm stright , white , no longer middle class , anti - IRA , have ...*". Unlike other symbols, an apostrophe is not separated from the following characters.

2.2 Phrase filtering

At the phrase filtering phase, we filtered senses using the satellite feature, which is marked with *sat* tag in training and test corpus given by the task organizer. For example, in a sentence *This air of disengagement <head sats="carry_over.067:0">carried</head> <sat id="carry_over.067:0">over</sat> to his apparent attitude toward his things, carried over* is a phrase and also a satellite feature.

Phrase filtering is applied to sense disambiguation as in Table 1

Table 1: phrase filtering and sense disambiguation

if the number of filtered senses = 1 then determine sense
else if the number of filtered senses > 1 then execute sense-tagger with the filtered senses
else if the number of filtered senses = 0 then execute sense-tagger with all senses

There are satellite features in the English lexical sample, but not in the Korean lexical sample. Hence, phrase filtering was applied only in the English lexical sample task.

2.3 Classification Information Model (CIM)

The CIM is a kind of classification model based on the entropy theory. Given an input instance, the CIM decides the proper class of the instance by considering individual decisions made by each feature of the instance. In the model, the proper class of an instance,X, is determined by Equation 1.

$$Class(X) \stackrel{\text{def}}{=} \arg\max_{class_j} \text{Rel}(class_j, X) \quad (1)$$

where $class_j$ is the j-th class and $\text{Rel}(class_j, X)$ is the relevance between the j-th class and the instance X. Here, if we assume that features are independent of each other, the relevance can be defined as in Equation 2.

$$\text{Rel}(class_j, X) = \sum_{i=1}^{m} x_i w_{ij} \quad (2)$$

where m is the size of the feature set, x_i is the value of the i-th feature and w_{ij} is the weight of the i-th feature for the j-th class. In Equation 2, x_i has a binary value (1 if the feature occurs within the window, 0 otherwise) and w_{ij} is defined in terms of classification information.

The classification information of a feature is composed of two components. One is the discrimination score (DS), which represents the discrimination ability of classifying instances. The other is the most probable class (MPC), which represents the most closely related class to the feature. w_{ij} is defined by using these two components as follows:

$$w_{ij} \stackrel{\text{def}}{=} \begin{cases} \text{DS}_i & \text{if } class_j = \text{MPC}_i \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

In Equation 3, DS_i and MPC_i represent the DS and MPC of the i-th feature, respectively. In the CIM, DS and MPC are defined in terms of the conditional probability of a class given a feature, which is normalized by the corpus size. The normalized conditional probability is defined as follows:

$$\begin{aligned} \hat{p}_{ji} &\stackrel{\text{def}}{=} \frac{p(class_j|f_i)\frac{\overline{N(class)}}{N(class_j)}}{\sum_{k=1}^{n} p(class_k|f_i)\frac{\overline{N(class)}}{N(class_k)}} \\ &= \frac{p(f_i|class_j)}{\sum_{k=1}^{n} p(f_i|class_k)} \end{aligned} \quad (4)$$

In Equation 4, $\hat{p}_{ji}$ is a normalized conditional probability, $N(class_j)$ is the number of instances belonging to the j-th class in the training data, $\overline{N(class)}$ is the average number of instances for each class and n is the number of classes. Given the normalized conditional probability distribution, DSs and MPCs are defined as follows:

$$\begin{aligned} DS_i &\stackrel{\text{def}}{=} \log_2 n - H(\hat{p}_i) \\ &= \log_2 n + \sum_{j=1}^{n} \hat{p}_{ji} \log_2 \hat{p}_{ji} \end{aligned} \quad (5)$$

$$\begin{aligned} MPC_i &\stackrel{\text{def}}{=} \arg\max_{class_j} \hat{p}_{ji} \\ &= \arg\max_{class_j} p(f_i|class_j) \end{aligned} \quad (6)$$

In Equation 5, $H(\hat{p}_i)$ is the entropy of the i-th feature over the normalized conditional probability distribution.

2.4 Modifying CIM

The CIM has a problem caused by using MPCs, which is information loss. For example, let us consider the situation in Table 2 and Table 3. Table 2 shows the normalized conditional probability distribution, DSs and MPCs of features in an instance. Table 3 shows the weights and the relevance values at the CIM using w_{ij} and at the modified CIM using $\hat{w}_{ij}$, for the instance of Table 2. The feature f_1 co-occurred with $class_1$ and $class_2$ and the MPC of f_1 is $class_1$ at Table 2. In the CIM, this feature

Table 2: A normalized conditional probability, DSs and MPCs of features of an instance

feature	normalized conditional probability($\hat{p}_{ji}$)				DS	MPC
	$class_1$	$class_2$	$class_3$	$class_4$		
f_1	0.7	0.3	0	0	1.1187	$class_1$
f_2	0	0.4	0.6	0	1.0290	$class_3$
f_3	0	0.4	0.1	0.5	0.6390	$class_4$

Table 3: The weights and the relevance values at the CIM using w_{ij} and at the modified CIM using $\hat{w}_{ij}$, for the instance of Table 2

feature	weight(w_{ij})				weight($\hat{w}_{ij}$)			
	$class_1$	$class_2$	$class_3$	$class_4$	$class_1$	$class_2$	$class_3$	$class_4$
f_1	1.1187	0	0	0	0.7831	0.3356	0	0
f_2	0	0	1.0290	0	0	0.4116	0.6174	0
f_3	0	0	0	0.6390	0	0.2556	0.0639	0.3195
$Rel(class_j, X)$	1.1187	0	1.0290	0.6390	0.7831	1.0028	0.6813	0.3195

contributes to only $class_1$. Actually the feature f_1 can contribute to distinguishing $class_2$ from $class_3$ if it consults the normalized conditional probability distribution. In the CIM, however, the feature can not distinguish them because their weights have the same value.

Another aspect of the problem is that the CIM fails to capture the minor contribution of features, which is crucial in the case where the sum of the minor contribution of features to a non-MPC class dominates that of the major contribution of features to MPC classes. For example, at Table 2, all features, f_1, f_2, and f_3, have different MPCs: $class_1$, $class_3$ and $class_4$, respectively. it is also obvious that they have some minor contribution to the $class_2$. The CIM will classify the instance as $class_1$ because $Rel(class_1, X) = 1.1187$ is the maximum number among the $Rel(class_j, X)$. However, if we consider the minor contribution of all the features, we prefer $class_2$ to $class_1$ because $class_2$ intuitively gains the total contribution more than $class_1$.

A solution to the problem may be not to use MPCs, but to use a measure of contribution of a feature to a class which is proportional to the discrimination score of the feaure and the normalized conditional probability of the class given the feature. The modified CIM can be defined as follows:

$$\mathrm{Rel}(class_j, X) = \sum_{i=1}^{m} x_i \hat{w}_{ij} \quad (7)$$

$$\hat{w}_{ij} \stackrel{\mathrm{def}}{=} DS_i \times \hat{p}_{ji} \quad (8)$$

As shown in Table 3, the $\hat{w}_{12}$ is larger than $\hat{w}_{13}(0.3356 > 0)$ and the instance is classified not as $class_1$ but as $class_2$ because $Rel(class_2, X) = 1.0028 > Rel(class_1, X) = 0.7831$, which is based on the modified CIM.

2.5 Feature Space

We used three types of features for word sense disambiguation: local, topical and bigram context. In the preliminary experiment, we have observed that, when the CIM considered all these three types of features, it mostly achieved the best result.

2.5.1 Local context

In a local context, there can be features of the following templates for all words within its window:

- in the English lexical sample task
  - *word_position* : a word and its position
  - *word_POS* : a word and its part-of-speech
  - *POS_position* : the part-of-speech and position of a word
- in the Korean lexical sample task
  - *morpheme_position* : a morpheme[1] and its position.
  - *morpheme_POS* : a morpheme and its part-of-speech.
  - *POS_position* : the part-of-speech and position of a morpheme

In the English lexical sample task, *word* is a surface form and can be either one of open-class words whose POS is one of the noun, verb, adjective, and adverb; or one of closed-class words whose POS is

[1] A Korean sentence is composed of one or more *eojeols*, which are separated by spaces, and an *eojeol* consists of one or more morphemes.

one of the determiner, preposition, pronoun, and punctuation. The window size of ±3 words in the English lexical sample task and the window size from −2 to +3 word in the Korean lexical sample task were empirically chosen.

In the first phase of the experiments, we used just one complicated template, *word_position_POS*(in Korean *morpheme_position_POS*), which brought about data sparseness problem. So we split the template into three simpler templates.

2.5.2 Topical context

A topical context includes features of the following templates for all open-class words within its window:

- in the English lexical sample task
  - *word* : an open-class word.
- in the Korean lexical sample task
  - *morpheme* : an open-class morpheme.

The window size of ±1 sentences in the English lexical sample task and the window size of all sentences in the Korean lexical sample task were empirically chosen.

2.5.3 Bigram context

In a bigram context, there can be features of the following templates for all word-pairs within its window:

- in the English lexical sample task
  - $(word_i, word_j)$: the i-th word and j-th word $(i>j)$
  - $(word_i, POS_j)$: the i-th word and j-th part-of-speech $(i>j)$
- in the Korean lexical sample task
  - $(eojeol_i, eojeol_j)$: the i-th eojeol and j-th eojeol $(i>j)$

Unlike local and topical contexts, bigram contexts are composed of only bigram information surrounding the polysemous word. The window size of ±2 words in the English lexical sample task and the window size from −2 to +3 word in the Korean lexical sample task were empirically chosen.

3 Experimental Result

The following tables show the results of our systems at SENSEVAL-2 (Table 4). For the Korean lexical sample task at SENSEVAL-2, only fine-grained sense distinction was made.

Table 4: Results of KUNLP systems at SENSEVAL-2

task	prec.	recall
English Lexical Sample (fine g.)	0.629	0.629
English Lexical Sample (coarse g.)	0.697	0.697
Korean Lexical Sample (fine g.)	0.698	0.74

4 Conclusion

We have described the modified CIM used for word sense disambiguation at SENSEVAL-2. In the experiments, three types of features; local, topical, and bigram context, are used. Our system ranked as the highest at the Korean lexical sample task and as the topmost group at the English lexical sample task among the supervised models at SENSEVAL-2. Consequently, the results back up the fact that the modified CIM and three types of features are useful for discriminating word senses.

References

Eric Brill 1994. Some advances in rule-based part of speech tagging. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94).*

Martin Chodorow, Claudia Leacock and George A. Miller 2000. A Topical/Local Classifier for Word Sense Identification. In *Computers and the Humanities 34: 115-120.*

Ho Lee, Dae-Ho Baek and Hae-Chang Rim 1997. Word Sense Disambiguation Based on The Information Theory. In *Proceedings of Research on Computational Linguisitcs Conference.*

Ho Lee, Hae-Chang Rim and JungYun Seo 2000. Word Sense Disambiguation Using the Classification Information Model. In *Computers and the Humanities 34: 141-146.*

WASP-Bench: a Lexicographic Tool Supporting Word Sense Disambiguation

David Tugwell & Adam Kilgarriff
ITRI, University of Brighton
Lewes Road, Brighton BN2 4GJ, UK
{David.Tugwell,Adam.Kilgarriff}@itri.bton.ac.uk

Abstract

We present WASP-Bench: a novel approach to Word Sense Disambiguation, also providing a semi-automatic environment for a lexicographer to compose dictionary entries based on corpus evidence. For WSD, involving lexicographers tackles the twin obstacles to high accuracy: paucity of training data and insufficiently explicit dictionaries. For lexicographers, the computational environment fills the need for a corpus workbench which supports WSD. Results under simulated lexicographic use on the English lexical-sample task show precision comparable with supervised systems[1], without using the laboriously-prepared training data.

1 Introduction

WASP-Bench[2] is a web-based tool supporting both corpus-based lexicography and Word Sense Disambiguation. The central premise behind the initiative is that deciding what the senses for a word are, and developing a WSD program for it, should be tightly coupled. In the course of the corpus analysis, the lexicographer explores the textual clues that indicate a word is being used in one sense or another; given an appropriate computational environment, these clues can be gathered and used to seed a bootstrapping WSD program.

This strategy clearly requires human input for each word to be disambiguated, which may raise the objection that the lexicon is far too large for any word-by-word work to be viable. However, the amount of human interaction needed is far less than that involved in preparing training data[3] and lexicographers are already in the position of having to inspect every word in the vocabulary. If they use a interactive tool such as the WASP-Bench to help them in this, then total coverage becomes a feasible proposition.

2 WASP-Bench Methodology

The workbench is implemented in perl and uses cgi-scripts and a browser for user interaction.

2.1 Grammatical relations database

The central resource is a collection of all grammatical relations holding between words in the corpus. The corpus currently used in WASP-Bench is the British National Corpus[4] (BNC): . Using finite-state techniques operating over part-of-speech tags, we process the whole corpus finding quintuples of the form: {**Rel**, **W1**, **W2**, **Prep**, **Position**}, where **Rel** is a relation, **W1** is the lemma of the word for which **Rel** holds, **W2** is the lemma of the other open-class word involved, **Prep** is the preposition or particle involved and **Position** is the position of **W1** in the corpus. Relations may have null values for **W2** and **Prep**. The database contains 70 million quintuples.

The current inventory of relations is shown in Table 1. All inverse relations, ie. **subject-of** etc, found by taking W2 as the head word instead of W1 are explicitly represented, to give a total of twenty-six distinct relations. These provide a flexible resource to be used as the basis of the computations of the workbench. Keeping

[1]It should be noted that the lower figure for recall reflects solely the fact that not all words were attempted due to time constraints.

[2]The system has been developed under EPSRC project M54971. A demo is available at *http://wasps.itri.bton.ac.uk*. The second author was also a co-ordinator for the SENSEVAL-2 evaluation exercise–to limit any conflict of interest only the first author was involved applying the system to the SENSEVAL-2 task, and had no prior knowledge of the format of the task.

[3]See results section for details.

[4]100 million words of contemporary British English. see http://info.ox.ac.uk/bnc

relation	example
bare-noun	the angle of **bank**^{1}
possessive	my **bank**^{1}
plural	the **banks**^{1}
passive	was **seen**^{1}
reflexive	**see**^{1} herself
ing-comp	**love**^{1} eating fish
finite-comp	**know**^{1} he came
inf-comp	**decision**^{1} to eat fish
wh-comp	**know**^{1} why he came
subject	the **bank**^{2} **refused**^{1}
object	**climb**^{1} the **bank**^{2}
adj-comp	**grow**^{1} **certain**^{2}
noun-modifier	**merchant**^{2} **bank**^{1}
modifier	a **big**^{2} **bank**^{1}
and-or	**banks**^{1} and **mounds**^{2}
predicate	**banks**^{1} are **barriers**^{2}
particle	**grow**^{1} **up**^{p}
Prep+gerund	**tired**^{1} **of**^{p} eating fish
PP-comp/mod	**banks**^{1} **of**^{p} the **river**^{2}

Table 1: Grammatical Relations

the position numbers of examples allows us to find associations between relations and to display examples.

2.2 Word Sketches

The user enters the word and using the grammatical relations database, the system composes a **word sketch** for this word. This is a page of data such as Table 2, which shows, for the word in question (W1), ordered lists of high-salience grammatical relations, relation-W2 pairs, and relation-W2-Prep triples for the word.

The number of patterns shown is set by the user, but will typically be over 200. These are listed for each relation in order of salience, with the count of corpus instances. The instances can be instantly retrieved and shown in a concordance window. Producing a word sketch for a medium-to-high frequency word takes in the order of ten seconds.

Salience is calculated as the product of Mutual Information I (Church and Hanks, 1989) and log frequency. I for a word $W1$ in a grammatical relation Rel[5] with a second word $W2$ is calculated as:

$$I(W1, Rel, W2) = log(\frac{\|*, Rel, *\| \times \|W1, Rel, W2\|}{\|W1, Rel, *\| \times \|*, Rel, W2\|})$$

The notation here is adopted from (Lin, 1998) (who also spells out the derivation from the definition of I). $\|W1, Rel, W2\|$ denotes the frequency count of the triple $\{W1, Rel, W2\}$[6] in the grammatical relations database. Where $W1$, Rel or $W2$ is the wild card ($*$), the frequency is of all the dependency triples that match the remainder of the pattern.

The word sketches are presented to the user as a list of relations, with items in each list ordered according to salience. Our experience of working lexicographers' use of Mutual Information or log-likelihood lists shows that, for lexicographic purposes, these over-emphasise low frequency items, and that multiplying by log frequency is an appropriate adjustment.

2.3 Matching patterns with senses

The next task is to enter a preliminary list of senses for the word, possibly in the form of some arbitrary mnemonics: for example, MONEY, CLOUD and RIVER for three senses of *bank*.[7] This inventory may be drawn from the user's knowledge, from a perusal of the word sketch, or from a pre-existing dictionary entry.

As Table 2 shows, and in keeping with "one sense per collocation" (Yarowsky, 1993) in most cases, high-salience patterns or **clues** indicate just one of the word's senses. The user then has the task of associating, by selecting from a pop-up menu, the required sense for unambiguous clues. The number of relations marked will depend on the time available, as well as the complexity of the sense division to be made. The act of assigning senses to patterns may very well lead the user to discover fresh, unconsidered senses usages of the word.

The pattern-sense associations are then submitted to the next stage: automatic disambiguation.

2.4 The Disambiguation Algorithm

The workbench currently uses Yarowsky's decision list approach to WSD (Yarowsky, 1995). This is a bootstrapping algorithm that, given

[5] {Grammatical-relation, preposition} pairs are treated as atomic relations in calculating MI.

[6] Or, strictly, of the quintuple $\{W1, Rel - part - 1, W2, Rel - part - 2, ANY\}$.

[7] WASP-Bench can also be used for Machine Translation lexicography, where arbitrary mnemonics would be replaced by target language translations.

subj-of	num	sal	obj-of	num	sal	modifier	num	sal	n-mod	num	sal
lend	95	21.2	burst	27	16.4	central	755	25.5	merchant	213	29.4
issue	60	11.8	rob	31	15.3	Swiss	87	18.7	clearing	127	27.0
charge	29	9.5	overflow	7	10.2	commercial	231	18.6	river	217	25.4
operate	45	8.9	line	13	8.4	grassy	42	18.5	creditor	52	22.8
modifies			**PP**			**inv-PP**			**and-or**		
holiday	404	32.6	of England	988	37.5	governor of	108	26.2	society	287	24.6
account	503	32.0	of Scotland	242	26.9	balance at	25	20.2	bank	107	17.7
loan	108	27.5	of river	111	22.1	borrow from	42	19.1	institution	82	16.0
lending	68	26.1	of Thames	41	20.1	account with	30	18.4	Lloyds	11	14.1

Table 2: Extract of word sketch for *bank*

some initial seeding, iteratively divides the corpus examples into the different senses. Yarowsky notes that the most effective initial seeding option he considered was labelling salient corpus collocates with different senses. The user's first interaction with the workbench is just this.

At the user-input stage, only clues involving grammatical relations are used. At the WSD algorithm stage, some "bag-of-words" and n-gram clues are also considered. Any content word (lemmatised) occurring within a k-word window of the nodeword is a bag-of-words clue.[8] N-gram clues capture local context which may not be covered by any grammatical relation. The n-gram clues are all bigrams and trigrams including the nodeword. N-grams and context-word clues frequently duplicate the grammatical relations already found, but the merit of the decision list approach is that probabilities are not combined, so such dependencies are not a problem.

2.5 Sense Profiles

The output of the algorithm is both a sense disambiguated corpus, and a decision list. The decision list can be viewed as a lexical entry or as a WSD program. It will contain {Rel, W2} pairs (as in the original word sketch), bag-of-words words, and n-grams. The components of the decision list which assign to a particular sense can be displayed as "sense profiles", in a manner comparable to the original word sketch. They will contain new clues, not originally seen in the word sketch and may point to new senses or usages needing addition to the lexical entry. Users can then re-run the WSD algorithm, iterating until they are satisfied with the sense inventory, and with the accuracy of the disambiguation performed.

[8]The user can set the value of k. The default is currently 30.

3 Evaluating the workbench

3.1 Lexicographic evaluation

For the last two years, a set of 6000 word sketches has been used in a large dictionary project (Rundell, 2002), with a team of thirty professional lexicographers covering every medium-to-high frequency noun, verb and adjective of English. The feedback received is that they are hugely useful, and transform the way the lexicographer uses the corpus. They radically reduce the amount of time the lexicographers need to spend reading individual instances, and give the dictionary improved claims to completeness, as common patterns are far less likely to be missed.

3.2 Results for senseval-2

Performance as a WSD system was evaluated on the SENSEVAL-2 English lexical sample exercise.

The words to be tested were divided between the first author and one paid volunteer, who had no previous experience of WASP-Bench. We carried out the procedure as above, with the difference that instead of having to establish a sense inventory, the inventory was already given as that of WordNet. After assigning sufficient clues to cover the various senses, these assignments were submitted as seeds to the disambiguation algorithm. Using the example sentences from the BNC this gave us a decision list of clues, which could then be used to disambiguate the test sentences.

The marking of senses took anywhere from 3 to 35 minutes, depending upon the subtlety of the sense divisions to be made. The average time was around 15 minutes per word. A substantial part of this was taken up by reading and understanding the dictionary entry even before patterns were marked. Crucially we made no use of the training data,[9] although this would certainly have been of use as a reference in clarifying the sense distinctions to be made. Unfortunately, due to severe time constraints, it only proved possible to carry out analysis for the 29 nouns and 15 adjectives in the lexical sample, and there was no time to carry out the analysis of the verbs.[10]

Results on the task were 66.1% for coarse-grained precision and 58.1% for fine-grained.[11] This was significantly higher than other systems which did not use the training data (the best scores being 51.8% for coarse-grained and 40.2% for fine-grained precision), demonstrating that the relatively small amount of human interaction is very beneficial. Indeed, the system's performance was similar to the majority of systems which had used the training data.

3.2.1 Significant problems

The most pervasive problem was the difficulty of getting a clear conception of the sense distinctions made in the inventory, here WordNet. Without this, assigning putative senses to clues could be an exasperating and painful task.

To illustrate, for the adjective *simple* there were no less than 13 sense distinctions to be made, the first two of which were particularly hard to distinguish:

1. simple (vs. complex) – (not complex or complicated or involved): *a simple problem*

2. elementary, simple, uncomplicated, unproblematic – (not involved or complicated): *an elementary problem in statistics*

[9]In fact, we had to download the data to find out the words to be tested, but made no other use of it.

[10]Also no results were returned for the noun *day*, as processing the 93,000+ examples in the BNC led to an processing delay that could not be fixed in time.

[11]Due to the limited number of words attempted the figures for recall were 36.3% and 31.9%. It should be understood that there was no precision/recall tradeoff here—the system returned an answer for all sentences in the words it covered.

Unsurprisingly, the system fared particularly badly here with 37.9% precision, while inter-annotator agreement was also low at 67.8%.

3.2.2 Previous results

We previously measured the performance of the system on the dataset from the SENSEVAL-1 exercise (Kilgarriff and Palmer, 2000) under similar conditions of use. Results for the WASP-Bench here were significantly higher at 74.9% precision which was very close to the best supervised system (within 1%). This was undoubtedly due to the clearer sense distinctions and greater number of examples to be found in the sense inventory used for this task in SENSEVAL-1, which made it possible to assign senses to clues with more confidence.

4 Summary

The results for the WASP-Bench show that high-quality disambiguation can be achieved with much less human interaction than is needed for preparing a training corpus. Furthermore, this interaction can be motivated since it has been shown to be of proven benefit for the users of the system: lexicographers. Establishing this synergy may prove to be of great importance for both camps.

References

Kenneth Church and Patrick Hanks. 1989. Word association norms, mutual information and lexicography. In *ACL Proceedings, 27th Annual Meeting*, pages 76–83, Vancouver.

Adam Kilgarriff and Martha Palmer. 2000. Introduction, Special Issue on SENSEVAL: Evaluating Word Sense Disambiguation Programs. *Computers and the Humanities*, 34(1–2):1–13.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *COLING-ACL*, pages 768–774, Montreal.

Michael Rundell. 2002. *Macmillan English Dictionary for Advanced Learners.* Macmillan.

David Yarowsky. 1993. One sense per collocation. In *Proc. ARPA Human Language Technology Workshop*, Princeton.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivalling supervised methods. In *ACL 95*, pages 189–196, MIT.

Word Translation Based on Machine Learning Models Using Translation Memory and Corpora

Kiyotaka Uchimoto†, Satoshi Sekine‡, Masaki Murata†, and Hitoshi Isahara†

†Communications Research Laboratory
2-2-2, Hikari-dai, Seika-cho, Soraku-gun,
Kyoto, 619-0289 Japan
{uchimoto, murata, isahara}@crl.go.jp

‡New York University
715 Broadway, 7th floor
New York, NY 10003, USA
sekine@cs.nyu.edu

Abstract

SENSEVAL-2 was held in Spring, 2001. It consisted of several tasks in various languages. In this paper, we describe our system used for one of these tasks: the Japanese translation task. With an accuracy of 63.4%, our system was the third best system in the contest among nine systems developed by seven groups.

1 Introduction

In the Japanese translation task, the senses of a word were defined in terms of the word's translations. Given an input sentence and a target word in the sentence, our system first estimates the similarity between the input sentence and parallel example sets called "Translation Memory". It then selects an appropriate translation of the target word by using the example set with the highest similarity. The similarity is calculated using dynamic programming and a machine learning model, which assesses the similarity based on the similarity of a string, words to the left and to the right of the target word in the input sentence, content words in the input sentence and their translations, and co-occurrence of content words in bilingual and monolingual corpora in English and Japanese.

2 Japanese Translation Task

In general, the definition of word senses depends on the goal of a task. The goal of the Japanese translation task is word selection in translation, where the target language is English. Therefore, word senses are defined as translations (translated words/phrases).

Before the contest, a Japanese-English parallel phrase/sentence set (Translation Memory, henceforth referred to as *TM*) was given to the participants as training data. In the TM, for each Japanese headword, there was a set of pairs of a Japanese expression including a headword and an English translation of the expression. We call these pairs *examples*. Some of the examples are shown in Figure 1.

```
<entry id="1" headword= "遠慮">
    <sense id="1-1">
        <jexpression> 母に遠慮する </jexpression>
        <eexpression>to feel constrained for one's
        mother</eexpression>
    </sense>
    <sense id="1-2">
        <jexpression> 母への遠慮 </jexpression>
        <eexpression>constraint toward one's
        mother</eexpression>
        <transmemo>UC</transmemo>
    </sense>
    <sense id="1-3">
        <jexpression> 献金を遠慮してもらう
        </jexpression>
        <eexpression>to request to refrain from
        donation</eexpression>
    </sense>
    .....
</entry>
```

Figure 1: Examples in TM.

In the formal test (contest), the participants were given a set of texts each of which was marked by a target word. For each target word, the participants were required to submit either a sense id of the example (the number assigned to each example in the TM), which can be used to translate the target word, or a translation of the target word. In the latter case, a translation of the word itself, a translation of a sequence of words including the target word, or a translation of the whole sentence could be submitted.

Answers were prepared for each target word in the formal test. The answers could consist of one or more sense id's in the TM, or of possible translations. The output of each system was evaluated in terms of accuracy, defined as a percentage of answers identified correctly by the system. An answer was judged to have been identified correctly when a sense id or a translation selected by the system was found in the answer.

3 Word Translation Model

Given an input sentence and a target word in the sentence, our model selects an appropriate translation of the target word or a sense id of examples appropriate for the translation of the target word by using examples with the highest similarity, estimated between the examples and the input sentence. In this paper, we call this model a *word translation model.* The source language is Japanese and the target language in translation is English. Henceforth we call a headword translation an *English headword.*

The similarity between an input sentence and examples is calculated by the following two methods:

1. A method based on the similarity of a string of characters (Method 1) : The similarity is defined as the amount of agreement between an input sentence and a Japanese example, expressed as a percentage.
2. A method based on machine learning models (Method 2) : The similarity is defined as the confidence or probability estimated by machine learning models. English headwords are used as classes (or categories) in machine learning models. Since the TM has examples with the same English headword, the similarity estimated by a model is the similarity between the input sentence and a set of examples.

A model is prepared for each Japanese headword. Given an input sentence, the similarity between the input sentence and each example is calculated by a model using Method 1. If the similarity is equal to or greater than a certain threshold, the model returns either the sense id of the example with the highest similarity or an English headword of the example. Otherwise, a model in Method 2 selects and returns an English headword.

The following sections describe the two methods in greater detail.

3.1 Method Based on the Similarity of A String of Characters (Method 1)

When an example with the highest similarity is found, it is given the highest priority, and either the sense id or the English headword of the example is selected as an output.

When calculating the agreement rate between an input sentence and an example, the rightmost word of the Japanese example is stemmed. In other words, when the rightmost word is a function word or a auxiliary verb such as "SURU (do)", it is eliminated. When the rightmost word is a predicate, its inflectional part is also eliminated. For example, the stemmed examples in Figure 1 are "母に遠慮", "母への遠慮", and "献金を遠慮", respectively. The agreement rate is calculated as a percentage of characters in the Japanese example that correspond to those in the input sentence. The correspondence is evaluated by comparing the Japanese example and the input sentence character by character. This can be done by using the UNIX command "diff" in a dynamic programming method. [1] The similarity is calculated by using the following equation.

$$Similarity = \frac{\left(\begin{array}{l}\text{the number of characters}\\ \text{corresponding to characters}\\ \text{in input sentence}\end{array}\right)}{\left(\begin{array}{l}\text{the number of characters in}\\ \text{stemmed Japanese example}\end{array}\right)} \quad (1)$$

When several examples with the highest similarity are found, the one having the longest Japanese example is selected except when the length of corresponding part is shorter than that of the Japanese headword.

However, it is unrealistic to expect that an example that is almost the same as the input sentence can be found because it is difficult to install all possible examples into the TM. So, when there is no example whose similarity is equal to or greater than the threshold, the method described in the next section is used.

3.2 Method Based on Machine Learning Models (Method 2) [2]

To select an appropriate example with the same usage as that of the input sentence, the similarity must be calculated by extracting the most important information from various conflicting sources of information related to the input sentence and examples. Since we want to avoid making complicated rules, we use machine learning models to calculate the similarity. Instead of all examples in the TM, English headwords are used as classes in machine learning models. Therefore, examples having the same English headword are put into the same class and are considered to have the same similarity.

[1] A description on how to use "diff" can be found in (Murata and Isahara, 2001).

[2] Work on using machine learning methods for the translation of tenses, aspects, and modalities can be found in (Murata et al., 2001a).

Classes identified by machine learning models are basically English headwords in TM, and they are detected manually. For example, English headwords of the examples in Figure 1 are "feel constrained", "constraint", and "refrain", respectively. When English headwords are verbs, they are represented by their basic forms. English words obtained when a Japanese headword is looked up in a Japanese-English dictionary are also used as classes.

For the training data, we use not only examples in the TM but also other data collected from bilingual dictionaries or a parallel corpus. The collected data consist of Japanese-English parallel phrases/sentences including both Japanese and English headwords, and they are used as complements of the training data.

For the machine learning models, we use SVM (Support Vector Machine), ME (Maximum Entropy), DL (Decision list), and SB (Simple Bayes). For each Japanese headword, the best model with the highest accuracy in 10-hold cross-validation on the training data is used for testing. The confidence of each class is estimated by probability distribution $p(a, b)$, where b is a context in a set of contexts, B, and a is a class in a set of classes, A. SVM is a classifier, and in this model, the confidence of each class cannot be represented by a probability distribution, but for the sake of convenience, we assign probability 1 to the most confident class estimated by SVM, and 0 to all other classes. The parameters in each model follow those used in (Murata et al., 2001b). Context b is represented by a set of features, that is, information derivable from the training data. The features used in our experiments were as follows:

1. Morphological information
   The string, basic form, major and minor parts of speech, and inflection type on six morphemes, three morphemes to the left and three morphemes to the right of the target word in an input sentence.
2. Character n-gram
   Character n-grams in an input sentence. Each n-gram must include the target word.
3. Highest matching
   An English headword in the example that has the longest string matching that of the input sentence and its length are used as features.
4. Frequency of a content word and its translation candidates

   We define a set of examples including the same English headword as an example set. For each English headword, we define the following six example sets:

   Example set 1 Japanese examples

   Example set 2 English examples

   Example set 3 Sentences similar to examples in Example set 1. They are collected from a Japanese monolingual corpus.

   Example set 4 Sentences similar to examples in Example set 2. They are collected from an English monolingual corpus.

   Example set 5 Union of Example sets 1 and 3

   Example set 6 Union of Example sets 2 and 4

   For each example set, Japanese-English parallel phrases/sentences including both Japanese and English headwords are collected from bilingual dictionaries or parallel corpora, and are added to the example set.

   Sentences similar to a certain example are defined as sentences that include a substring of the example. The substring must include the headword of the example. In our model, we use sentences collected from a monolingual corpus because we want the model to reflect a real distribution of words, both headwords and words to the left and right of the headwords.

   As content words, we used nouns, verbs, adjectives, adverbs, and attributives, except headwords, in the input sentence. For each content word in an input sentence and its translation candidates, the frequencies in each example set were used as features. The translation candidates of a content word were obtained when the content word was looked up in a Japanese-English dictionary. Each feature is represented by a combination of an example set, a headword, and the frequency of content words in the example set. When we find that the total frequency of content words in an example set is n, we assume that every feature whose frequency is between 1 and n is observed. For example, when the content word found in the given sentence is "mother", and it is found three times in the example set 1 for the headword "buy", the features "Example set 1 : buy : 1," "Example set 1 : buy : 2," and "Example set 1 : buy : 3" are assumed to be observed. By using these features, our model handles information about co-occurrence words of a headword in each corpus as a clue to translating the headword.

4 Experiment

4.1 Experimental conditions

The input and evaluation of the systems followed those of the Japanese translation task in SENSEVAL-2. A TM for 320 headwords was given to each participant in the middle of March, 2001. The average number of examples prepared for each headword was approximately 20. For the formal test, 40 target words (20 nouns and 20 verbs) were selected from the headwords. For each target word, 30 texts including the target words were prepared. The total number of the target words was 1,200.

As a bilingual dictionary, we used "EIJIRO" available at the web site of NIFTY, a network provider. As monolingual corpora, we used MAINICHI newspapers from 1991 to 2000, NIKKEI newspapers from 1995 to 1999, SANKEI newspapers from 1994 to 1999, and LDC data collected in 1994 and 1995, which include English newspaper articles for several years published by the Wall Street Journal, the Associated Press Writer, and the New York Times.

In the formal test, the threshold of similarity used in Method 1 was 1. JUMAN (Kurohashi and Nagao, 1999), a Japanese morphological analyzer, was used for morphological analysis in Method 2. As sentences similar to a certain example in Method 2, sentences that included a string obtained by stemming Japanese examples were extracted for Japanese examples, and sentences that included English headwords were extracted for English examples. As for the machine learning models, we could not select the most appropriate set of models by cross validation because not all learning processes could be finished by the deadline for submission. The models finally selected for the formal test were as follows:

- SVM : 23 words (12 nouns and 11 verbs)
- DL : 12 words (8 nouns and 4 verbs)
- SB : 5 words (5 verbs)

4.2 Experimental Results and Discussion

The accuracy obtained by our system in the formal test was 63.4% (761/1,200). The accuracy obtained by Method 1 and 2 were 91.0% (91/100) and 60.9% (670/1,100), respectively. Based on our results, we can draw the following conclusions:

- The system performance was related to the amount of training data per class in Method 2.
- The accuracy obtained for words whose English headwords were general words was not high even though there were more training data for these words than for other headwords for which the accuracy was high. We believe that this is due to the quality of automatically collected training data because general words appear in corpora quite frequently, and sometimes parallel sentences where Japanese and English headwords are not related to each other are collected. Therefore, we need to select automatically collected parallel sentences by aligning Japanese and English headwords.
- Method 1 improved the accuracy, especially for idiomatic expressions that rarely appeared in the training data. We applied Method 2 to the target words to which Method 1 was applied in the formal test, and achieved an even lower accuracy of 34.0%(34/100).
- The accuracy obtained by the SB model was low. We speculate that the SB model is not suitable for the feature sets used in the test.

5 Conclusion

This paper described our system used in SENSEVAL-2. Our model for word translation has the following characteristics: (1) It puts together examples having the same English headword into a set of examples, and selects a set of examples most similar to the input sentence by using machine learning models. (2) If an example that is almost the same as the input sentence is found, our model gives it the highest priority. (3) It automatically collects training data and information used for training from other language resources that are not only a bilingual corpus but also monolingual corpora of English and Japanese. We do not have to supervise anything except the detection of headword pairs in the examples.

References

Sadao Kurohashi and Makoto Nagao, 1999. *Japanese Morphological Analysis System JUMAN Version 3.61.* Department of Informatics, Kyoto University.

Masaki Murata and Hitoshi Isahara. 2001. NLP using DIFF. In *IPSJ-WGNL NL144-18*, pages 127–134. (in Japanese).

Masaki Murata, Kiyotaka Uchimoto, Qing Ma, and Hitoshi Isahara. 2001a. Using a Support-Vector Machine for Japanese-to-English Translation of Tense, Aspect, and Modality. In *ACL Workshop on the Data-Driven Machine Translation.*

Masaki Murata, Masao Utiyama, Kiyotaka Uchimoto, Qing Ma, and Hitoshi Isahara. 2001b. Experiments on Word Sense Disambiguation Using Several Machine-leaning Methods. In *IEICE-WGNLC2001-2.* (in Japanese).

Automatic WSD: Does it make sense of Estonian?

Kadri Vider and **Kaarel Kaljurand**
University of Tartu
Department of General Linguistics
Tiigi 78, 50410 Tartu, Estonia
kvider@psych.ut.ee and kaarel@ut.ee

Abstract

This paper describes a fully automatic Estonian word sense disambiguation system called *semyhe* which is based on Estonian WordNet (EstWN) hyponym/hypernym hierarchies and meant to disambiguate both nouns and verbs.

1 Short description of the system

The main inspiration for our system is Agirre and Rigau (1996) similar system that disambiguates the English noun senses based on WordNet hyponym/hypernym hierarchy, taking into consideration the distances between the nodes corresponding to the word senses in the WordNet tree as well as the density of the tree. They have also experimented with using meronyms/holonyms in addition to hyponyms/hypernyms but report that it does not improve the results.

Our main object was not to focus on the homonymous words only (lexical sample), but to try to disambiguate all nouns and verbs in the text. The Estonian WordNet (EstWN) also contains adjectives but they are not linked by hyponym/hypernym relations. The word sense disambiguation could also try to describe a unique sense for adverbs but in our case such words have not yet been included in the thesaurus.

As far as we know this is the first attempt on automatic Estonian word sense disambiguation.

1.1 Input

The input text for our system must be morphologically analyzed, meaning that each word is provided with its lemma and morphological reading. Taking those two into account we can localize the senses that correspond to the word in EstWN hyponym/hypernym tree (Vider et al., 1999). It must be mentioned that although the morphological description in the input can be quite detailed, we only use the information on whether the word is a noun or a verb.

A simple morphological analysis that only looks at the word-form and not its context can result in very ambiguous output. On average 45% of the words are morphologically ambiguous in Estonian texts (Kaalep, 1997). The ambiguity can be greatly reduced by also applying the Estonian morphological disambiguator (Kaalep and Vaino, 1998) to the text before the word sense disambiguation. Since even then the words can in principle stay morphologically ambiguous, our system doesn't require each word to have exactly one morphological reading assigned to it in the input text.

1.2 Output

Similarly to the morphological analysis, we do not try to provide each word with exactly one sense. In case two (or more) senses have equal evaluation results then both of those prevail in the output.

1.3 Sense disambiguation algorithm

We apply the exact same algorithm for both nouns and verbs. Nouns and verbs cannot be compared with each other since in terms of hyponym/hypernym hierarchy they are located in different trees in the thesaurus. So the disambiguation is carried out in 2 runs, first nouns are disambiguated, then verbs, or vice versa.

A window is shifted on the text and as a word moves through the window its senses are compared with the senses of other words in the window. The context is either made out of nouns or verbs depending on which part of speech is being disambiguated.

The basis of the comparison is the similarity between the senses which is defined through

the notion of conceptual distance, the distance between the nodes corresponding to the senses in EstWN tree. Winners are the senses that minimize the total distance between the word senses in the window, all the rest are removed from the list of candidates for the correct reading. *semyhe* leaves the word ambiguous when there are more than one senses with equal result. This usually happens when the senses of the context words are located in different hierarchies and hence can not be compared. Currently there are 108 different top nodes in EstWN, 29 corresponding to nouns and 79 to verbs.

In addition, the work of the system can be modified via several options in the configuration file:

- The window-size can be changed, increasing it makes the output less ambiguous since there is a higher possibility that the comparable senses end up in one window. On the other hand a bigger window may span across several sentences making the compared words possibly irrelevant to each other. For the moment we have used window of 5 words.
- Since we use no syntactic analysis before word sense disambiguation, the context of any word under observation is unstructured, the only syntactic information that we can use is therefore only the distance of the words from each other in terms of running text. A set of weights can be defined that is mapped to the distances, so that the similarity of the senses of the words that are far away from each other is less relevant for the total score.
- We can also take into account the average depth of the compared nodes in the tree — the bigger the depth the more reliable the score.

So far we haven't yet experimented with any of those options much.

2 Analysis of the results

For the purposes of analyzing the quality of disambiguation, tests were made with 12 manually sense tagged texts. These text samples were mainly from fiction, in a part also from newspapers and they contained approximately 10,000 tokens that corresponded to either nouns or verbs.

Manual tagging naturally had to remove all the morphological ambiguity, therefore the results obtained on those texts should be better than on the texts that have only been treated automatically before the word sense disambiguation. Words that occurred in the text but were not present in EstWN were marked as having 0 senses, approximately 30% of such words are proper names.

Manual tagging also recognized multi-word units, which in our case are mostly non-contiguous verbal phrases that are hard to detect automatically even if we had a complete list of such units.

Using *semyhe*, we set out to disambiguate all the nouns and verbs contained in the texts. Since *semyhe* can leave a word ambiguous, it makes sense to evaluate its work in terms of recall and precision. Table 1 lists *semyhe* results when the window of context words has size 5. The table also shows the results obtained with a random method which chooses exactly one sense for every word randomly (in this case recall and precision have equal values).

The row groups of the table refer respectively to the results with polysemous words and the overall results. Note that the words which were manually marked as having 0 senses were considered monosemous and so they are always correctly analyzed, with unique sense selected for every word.

	POS	recall	precision	random
polysem	nouns	0.543	0.347	0.423
	verbs	0.495	0.249	0.251
	both	0.514	0.283	0.292
overall	nouns	0.839	0.700	0.773
	verbs	0.601	0.338	0.412
	both	0.745	0.522	0.630

Table 1: *semyhe* results with 10,000 nouns and verbs

Figure 1 shows the distribution of words between different number of senses according to those texts. This shows the ambiguity that any automatic analysis has to cope with.

Note that there is an unusually large number of words with 9 different senses. The main

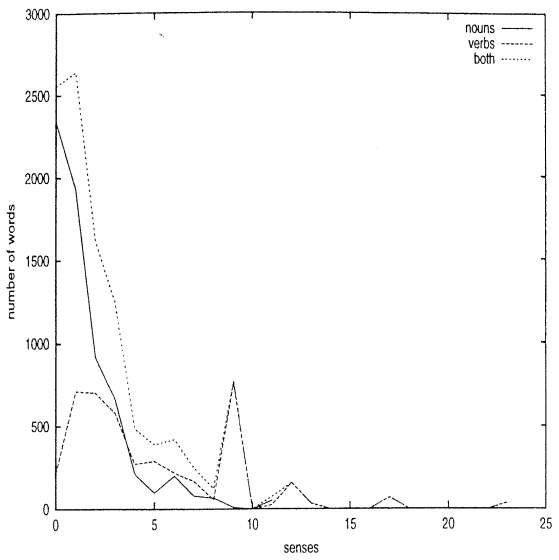

Figure 1: distribution of words in running text

reason for this is the frequent word 'olema' (*to be, to have*). Fortunately the distribution of its senses is highly skewed, meaning that mostly this word is used in one or two senses. Including the sense frequency information in the disambiguation process could considerably improve the results.

3 Problems and solutions

Several problems have been discovered concerning the relatively simple approach described above.

The output of the morphological analyzer often contains valuable information for word sense disambiguation which we have currently ignored.

- in some cases the word-form used in the text can uniquely specify the sense of the word, although its lemma is ambiguous, e.g. the word 'palk' can either mean *salary* or *log, tree trunk*, but its genitive form is different in each meaning (either 'palga' or 'palgi'). By using only the lemma we ignore this distinction that can be explicitly present in the text. The number of words behaving this way, though, is not very large.
- the modal verbs are explicitly marked in the output of the morphological disambiguator, when a verb is marked as such, then the senses that don't correspond to the modal senses could be removed and the winning sense should be chosen from the prevailing ones, e.g. the word 'saama' has all together 12 senses in the thesaurus, but only 2 of them correspond to the modal use of the word (either *can* or *may*).

Right now the frequency information of the senses has not been used. Most probably the results that could be obtained with the "commonest" baseline (Kilgariff and Rosenzweig, 2000) would beat the results of *semyhe*. We think that even the frequency information that could be calculated using the 10,000 manually sense-tagged words can be very useful for disambiguating purposes.

At the moment the input text contains no information about its syntactic structure, most importantly the verbal phrases and other multi-word units are not marked as such, therefore the analyzer tries to disambiguate all the components of a multi-word unit separately, this of course results in an incorrect analysis. Also, having the information about the syntactic structure of the sentences could help to reduce the number of possible senses to choose from. For example the word 'olema' that was already mentioned above has five more frequent senses:

1. *be* — copula, used with an adjective or a predicate noun
2. *exist* — have an existence, be extant
3. stay in place, be stationary or spend a certain length of time
4. be somewhere, occupy a certain area, occupy a certain position
5. *have, have got, hold* — have or possess, either in a concrete or an abstract sense

The first sense is present in complementary clauses; senses 2, 3 and 4 appear in existential sentences and the last one in possessive sentences. If the information about the nature of the sentence was present in the input text it would certainly help the disambiguation process.

The output of *semyhe* stays often very ambiguous. This either happens when the sense-

nodes of the context words are located in different trees so that their similarity cannot be calculated; or when different nodes of one word have the same parent node and are equally distant from the rest of the sense-nodes so that the similarity measure for them will be equal. The second reason may not be a big problem considering WordNet's fine-grainedness and the fact that for some applications a detailed sense distinction is not needed. The disambiguation result in this case can be simply seen as the union of the prevailed senses. Often, though, this approach does not hold, e.g. it is crucial for translation that the senses of the word 'naine' which can either stand for *woman*, *wife* or generally *female person*, are fully disambiguated, although the senses stand for more or less the same thing.

References

E. Agirre and G. Rigau. 1996. Word Sense Disambiguation using Conceptual Density. In *COLING-96*.

H.-J. Kaalep and T. Vaino. 1998. Kas vale meetodiga õiged tulemused? Statistikale tuginev eesti keele morfoloogiline ühestamine. *Keel ja Kirjandus*, 1:30–36. In Estonian. English title: Getting right result with a wrong method? Statistical morphological disambiguation of Estonian.

H.-J. Kaalep. 1997. An Estonian morphological analyser and the impact of a corpus on its development. *Computers and the Humanities*, 31:115–133.

A. Kilgariff and J. Rosenzweig. 2000. Framework and Results for English SENSEVAL. *Computers and the Humanities*, 34:15–48.

K. Vider, L. Paldre, H. Orav, and H. Õim. 1999. The Estonian Wordnet. In C. Kunze, editor, *Final Wordnets for German, French, Estonian and Czech*. EuroWordNet (LE-8328), Deliverable 2D014.

The Johns Hopkins SENSEVAL2 System Descriptions

**David Yarowsky, Silviu Cucerzan, Radu Florian,
Charles Schafer and Richard Wicentowski**
{yarowsky,silviu,rflorian,cschafer,richardw}@cs.jhu.edu

Department of Computer Science
Johns Hopkins University
Baltimore, Maryland, 21218, USA

Abstract

This article describes the Johns Hopkins University (JHU) sense disambiguation systems that participated in seven SENSEVAL2 tasks: four supervised lexical choice systems (Basque, English, Spanish, Swedish), one unsupervised lexical choice system (Italian) and two supervised all-words systems (Czech, Estonian). The common core supervised system utilizes voting-based classifier combination over several diverse systems, including decision lists (Yarowsky, 2000), a cosine-based vector model and two Bayesian classifiers. The classifiers employed a rich set of features, including words, lemmas and part-of-speech informatino modeled in several syntactic relationships (e.g. verb-object), bag-of-words context and local collocational n-grams. The all-words systems relied heavily on morphological analysis in the two highly inflected languages. The unsupervised Italian system was a hierarchical class model using the Italian WordNet.

1 The Feature Space

The JHU SENSEVAL2 systems utilized a rich feature space based on raw words, lemmas and part-of-speech (POS) tags in a variety of positional relationships to the target word. These positions include traditional bag-of-word context, local bigram and trigram collocations and several syntactic relationships based on predicate-argument structure (described in Section 1.2). Their use is illustrated on a sample English sentence for *train* in Figure 1.

1.1 Part-of-Speech Tagging and Lemmatization

Part-of-speech tagger availability varied across the languages included in this sense-disambiguation system evaluation. Transformation-based taggers (Ngai and Florian, 2001) were trained on standard data for English (Penn Treebank), Swedish (SUC-1 corpus) and Estonian (MultextEast corpus). For Czech, an available POS tagger (Hajič and Hladká, 1998), which includes lemmatization, was used. The remaining languages – Spanish, Italian and Basque – were tagged using an unsupervised tagger (Cucerzan and Yarowsky, 2000). Lemmatization was performed using a combination of supervised and unsupervised methods (Yarowsky and Wicentowski, 2000), and using existing trie-based supervised models for English.

"Many mothers do not even try to toilet **train** their children until the age of 2 years or later ..."

Feature type	Word	POS	Lemma
Context	...	...	...
Context	try	VB	try/V
Context	to	TO	to/T
Context	toilet	NN	toilet/N
Context	train	VBP	train/V
Context	their	DT	their/D
Context	...	...	...
Syntactic (predicate-argument) features			
Object	children	NNS	child/N
Prep	until	IN	until/I
ObjPrep	age	NN	age/N
Ngram collocational features			
-1 bigram	toilet	NN	toilet/N
+1 bigram	their	DT	their/D
-2/-1 trigram	to toilet *	TO-NN	to/T toilet/N *
-1/+1 trigram	to * their	TO-DT	to/T * their/D
+1/+2 trigram	their children	DT-NN	their/D child/N

Figure 1: Example sentence and extracted features

1.2 Syntactic Features

Extracted syntactic relationships in the feature space depended on the keyword's part of speech:

- for verb keywords – the head noun of the verb's object, particle/preposition and object-of-preposition were extracted when available.
- for noun keywords – the headword of any verb-object, subject-verb or noun-noun relationships identified for the keyword.
- for adjective keywords – the head noun modified by the adjective (if identifiable).

These syntactic features were extracted using simple heuristic patterns and regular expressions over the parts-of-speech surrounding the keyword.

2 Supervised Lexical Choice Systems

The supervised JHU systems utilize classifier combination merging the results of five diverse learning models.

2.1 Core Algorithm Design

The lexical choice task can be cast as a classification task: training data is given in the form of a set of word-document pairs $\mathcal{T} = [(w_i, D_{ij}), S_{ij}]_{i,j}$ (S_{ij} being the sense associated with the document D_{ij} of keyword w_i), labeled with the corresponding gold standard class. The goal is to establish the classification of a set of unlabeled word-document pairs $\mathcal{T}' = \{(w_i, D'_{ij})\}_{i,j}$, not previously seen in the training data. The training data $\mathcal{T}$ is used to estimate class probabilities and then the sense classification is made by choosing the class with the maximum a posteriori class probability:

$$S = \arg\max_{S'} P\left(S' | D\right) = \arg\max_{S'} P\left(S'\right) \cdot P\left(D | S'\right)$$

The disambiguation models used in our experiments are feature-based models. A feature is a boolean function defined as $f_w : F \times \mathcal{D} \rightarrow \{0, 1\}$, where F is the entire set of features and $\mathcal{D}$ is the document space. An overview of the exploited feature space was given in Section 1.

2.2 Vector-based Algorithms

Our Bayesian and cosine-based models use a common vector representation, capturing both traditional bag-of-words features and the extended Ngram and predicate-argument features in a single data structure.

In these models, a vector is created for each document in the collection:

$$D_i = (D_{ij})_{j=1,|F|}$$

where F is the entire utilized feature space

$$D_{ij} = \frac{c_{ij}}{N_i} W_j$$

where c_{ij} is the the number of times the feature f_j appears in document D_i, N_i is the number of words in the document D_i and W_j is the weight associated with the feature f_j.

To avoid confusion between the same word in multiple feature roles, feature values are marked with their positional type (e.g. *children_object*, *toilet_L*, and *their_R* as distinct from *children*, *toilet* and *their* in unmarked bag-of-words context).

The basic sense disambiguation algorithm proceeds as follows:

1. Vectors in the training data are assigned to classes based on their classification;
2. For each vector in the test data, the a posteriori class distribution is computed as

$$P\left(S | D\right) = \frac{\mathrm{Sim}\left(D, C_S\right)}{\sum_{S'} \mathrm{Sim}\left(D, C_{S'}\right)}$$

where C_S is the centroid corresponding to the sense S and Sim is the similarity measure used by the algorithm (cosine or Bayes).

3. The sample D is labeled with sense S if $S = \arg\max_{S'} P\left(S' | D\right)$.

2.2.1 The Cosine-based Model

In this model, traditional cosine similarity is used to compute similarity between a document D and a centroid C. The weight associated with a feature (F_j) is its inverse document frequency $W_j = \log \frac{N}{N_j}$, where N is the total number of documents and N_j is the number of documents containing feature f_j. Function words and POS tags were excluced from the cosine vectors.

2.2.2 The Bayesian Models

In the Bayes model, the Bayes similarity is computed as:

$$\mathrm{Sim}\left(D_i, S_j\right) = P\left(D_i, S_j\right) = P\left(S_j\right) P\left(D_i | S_j\right)$$

and the following assumption of independence is made:

$$P\left(D_i | C_S\right) = \prod_{f_j \in D_i} P\left(f_j | C_S\right)$$

The probability distribution $P\left(f_j | C_S\right)$ is obtained by smoothing the word relative frequencies in the cluster C_S. Given the lack of independence between the word-based and lemma-based feature spaces, these are utilized in two separate Bayesian models with output combined in Section 2.5.

2.3 Decision Lists

The decision list model we used in our system is a non-hierarchical variant of the method of interpolated decision lists described in Yarowsky (2000). For each feature f_i a smoothed log of likelihood ratio ($\log \frac{P(f_i|S_j)}{P(f_i|\neg S_j)}$) is computed for each sense S_j, with smoothing based on an empirically estimated function of feature type and relative frequency. Candidate features are ordered by this smoothed ratio (putting the best evidence first), and the remaining probabilities are computed via the interpolation of the global and history-conditional probabilities. By utilizing the single strongest-matching evidence in context, non-independent feature spaces combine readily without inflated confidence, and can be mapped to accurate and robust probability estimates as shown in Figure 2.

2.4 Additional Details

The English task differs slightly from the other lexical-choice tasks in that phrasal verbs are explicitly marked in the training and test data. To make reasonable use of this information, when a phrasal verb is marked, only corresponding phrasal senses are considered; conversely when a phrasal

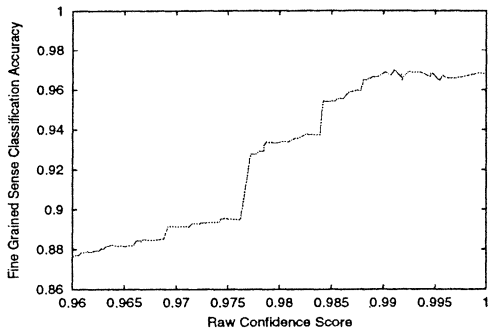

Figure 2: Mapping between raw confidence scores and classification accuracy for English decision lists

verb is not marked, no phrasal senses are considered. Likewise, when a training or test sentence matches a compound noun in the observed sense inventory (e.g. *art_gallery%1:06:00::*) only the matching phrasal sense(s) are considered unless there is at least one non-phrasal sense tagged in the training data for that compound (indicating the potential for both compositional and non-compositional interpretations).

2.5 Classifier Combination

Several classifier combination approaches were investigated in the system development phase. They are outlined below, along with their cross-validated performance on the English lexical-sample training data (in Table 1). In each case four individual classifiers were combined: the cosine model, two Bayes models (one based on words and one based on lemmas[1]), and the decision-list model.

The first two model combination approches simply averages the output of the participating classifiers over each candidate sense tag, in terms of $P(S_j|D_i)$ and $rank(S_j|D_i)$ respectively, with each classifier given an equal vote[2].

The remaining methods assign potentially variable weights to the votes of different classifiers. Interestingly, Equal Weighting of all four classifiers slightly outperforms classifier weighting proportional to each model's aggregate accuracy (Performance-Weighted voting), similar to the technique used for classifier combination in part-of-speech tagging in van Halteren et al. (1998). Finally, it was observed that on sentences where decision lists have high model confidence their accuracy exceeds other classifiers. Thus the most effective approach, based on training-data cross validation, was found to be a very basic Thresholded Model Voting:

- If the decision_list_confidence$\geq$ 0.985 (an empirically selected threshold) then return the output of the decision list;
- Otherwise, each system votes for the sense that is most likely under it and, another vote is obtained from the most probable class yielded by linear interpolation of the 4 classifiers.

This simple top-performing approach was utilized in the evaluation system, and is reasonably close to the performance of an Oracle upper bound for classifier combination (using the output of the single best classifier on each test instance – unknowable in practice).

Classifier Combination Method	Accuracy	
	Fine	Coarse
Model Averaging (excluding decision lists):		
Probability interpolation voting	.657	.728
Rank-averaged voting	.652	.709
Weighted Model Voting (includes decision lists):		
Equal-weighted Model Voting	.667	.736
Performance-Weighted Voting	.655	.724
Thresholded Model Voting	**.676**	**.746**
Oracle Voting (Upper Bound)	.734	.761

Table 1: Comparison of classifier combination methods on English (using 5-fold cross-validation)

3 Supervised All-Words Systems

3.1 Estonian All-words Task

Because of the importance of morphological analysis in a highly inflected language such as Estonian, a lemmatizer based on Yarowsky and Wicentowski (2000) was first applied to all words in the training data (and, at evaluation time, the test data). For each lemma, the P(sense|lemma) distribution was measured on the training data. For all lemmas exhibiting only one sense in the training data, this sense was returned. Likewise, if there was insufficient data for word-specific training (the sum of the minority sense examples for the word in training data was below a threshold) the majority sense in training was returned for all instances of that lemma. In the remaining cases where a lemma had more than one sense in training, with sufficient minority examples to adequately be modeled, the generic JHU lexical sample sense classifier was trained and applied.

3.2 Czech All-words Task

Czech is another example of a highly inflected language. A part-of-speech tagger and lemmatizer kindly provided by Jan Hajič of Charles University (Hajič and Hladká, 1998) was first applied to the data. Consistent with the spirit of evaluating sense disambiguation rather than morphology, the JHU system focused on those words where more than one sense was possible for a root word (e.g.

[1] On training-set cross-validation it was observed that the two systems were uncorrelated enough to make it useful to keep both of them.

[2] Decision lists are not included because they only assign a probability to their selected classifier output but not to lower-ranked candidates.

the -1 and -2 suffixes in the Czech inventory). In these cases, the fine-grained output of the Czech lemmatizer was ignored (in both training and test) and a generic lexical-sample sense classifier was applied to the sense-distinction tags extracted from the lemmatized training data (see Section 2), using the classification models employed in Estonian. Whenever insufficient numbers of minority tagged examples were available for training a word-specific classifier, the majority sense for the POS-level lemma was returned. Likewise, if only one possible sense tag was observed for any POS-level lemma analysis, then this unambiguous sense tag was returned.

4 Unsupervised Italian System

The Italian task stands out from the group of lexical choice tasks because no labelled training was data provided for Italian; instead a subset of the Italian Wordnet was provided. To obtain a sense classifier for Italian, we employed an unsupervised method that used hierarchical class models of the Wordnet relationships among words (synonymy, hypernomy, etc) and a large unannotated corpus of Italian newspaper data to obtain sense centroids.

First, every relationship type in the Italian Wordnet received an initial weight, based on a roughly estimated measure of the relative dissimilarity of two words in that relationship. For instance, the *synonymy* relationship received a small weight (words are semantically "close"), while other relationships (*has_near_synonym*, *causes*, *has_hypernym*) received proportionately larger weights (words are more semantically distant). Starting from the senses S of a target k, the wordnet relationships graph was explored, up to a given distance (two links away), creating "clouds" of similar words, M_S, together with a similarity[3] to the original sense, S.

For each of the words w in M_S, we extracted sentences from the unannotated corpus that contained the word w, and then considered them as being examples of context for the sense S of target k, and assigned them to the centroid C_S (the centroid of the sense S) with a weight corresponding to the similarity between the word w and the sense S (computed using the wordnet graph). After all the documents were distributed, the test documents were also assigned to the most probable cluster, similar to the other lexical choice tasks.

The centroids were then allowed to adjust in a manner similar to k-means clustering. At each step, the centroids were recomputed, after which each document migrated to the closest cluster (i.e. $\arg\max_S P(C_S|D)$), and the process was repeated. After the process converged, each test document was assigned the label corresponding to the sense centroid it converged into. This process is completely unsupervised, and the only structured resource that was used is the provided Italian Wordnet subset.

[3]The weight on a path was computed as the sum of the weights on the path, and the similarity was computed as $\mathrm{Sim}(w,S) = e^{-c(w,S)}$ – large weights result in 0 similarity.

Task	Accuracy on Test Data	
	Fine-Grained	Coarse-Grained
Basque	.757	.971
English	.642	.713
Spanish	.712	–
Swedish	.701	1.00
Italian	.353	.423
Czech	.935	–
Estonian	.666	–

Table 2: Official JHU system performance

5 Results

Table 2 lists the official performance of the JHU systems on unseen test data in the final SENSEVAL2 evaluation. Coarse-grained performance scores are based on a hierarchical sense clustering given by the task organizers in 4 of the languages. In the lexical sample tasks, these scores were obtained after correction of a simple bug in the merger of final system output as provided for in the SENSEVAL evaluation protocols.

As illustrated in the comparative performance tables elsewhere in this volume, the JHU systems are consistently very successful across all 7 languages and 3 major system types described here.

References

S. Cucerzan and D. Yarowsky. 2000. Language independent minimally supervised induction of lexical probabilities. In *Proceedings of ACL-2000*, pages 270–277, Hong Kong.

J. Hajič and Hladká. 1998. Tagging inflective languages: Prediction of morphological categories for a rich, structured tagset. In *Proceedings of COLING/ACL-98*, pages 483–490, Montréal.

G. Ngai and R. Florian. 2001. Transformation-based learning in the fast lane. In *Proceedings of NAACL-2001*, pages 40–47, Pittsburgh.

H. van Halteren, J. Zavrel and W. Daelemans. 1998. Improving Data Driven Wordclass Tagging by System Combination In *Proceedings of COLING/ACL-1998*, pages 491–497, Montreal.

D. Yarowsky and R. Wicentowski. 2000. Minimally supervised morphological analysis by multimodal alignment. In *Proceedings of ACL-2000*, pages 207–216, Hong Kong.

D. Yarowsky. 2000. Hierarchical decision lists for word sense disambiguation. *Computers and the Humanities*, 34(2):179–186.