

Word Sense Disambiguation by Information Filtering and Extraction

Jeremy Ellman, Ian Klincke & John Tait

**School of Computing & Information Systems
University of Sunderland, UK.**

Email : Jeremy.Ellman@mari.co.uk

Abstract

We describe a simple approach to Word Sense Disambiguation using Information Filtering and Extraction. The method fully exploits and extends the Information available in the Hector Dictionary. The algorithm proceeds by the application of several filters to prune the candidate set of words senses returning the most frequent if more than one remains. The experimental methodology and its implication are also discussed.

1. Introduction

Our interest in Word Sense Disambiguation comes from experiences with "Hesperus", a research system that clusters Internet web pages based on their similarity to sample texts (Ellman and Tait 1997). Hesperus uses a development of Morris and Hirst's (1991) idea of Lexical Chains--that coherent texts are characterised by sets of words with meanings related to the topics and that these topics may be detected by reference to an external thesaurus.

Of course, many words are ambiguous and have meanings corresponding to different thesaural headwords. This represents a problem for Lexical Chaining, since the selection of an incorrect sense means that the word may be joined to an inappropriate chain. It will also, more problematically, not be included in the correct chain, consequently disrupting the apparent topic flow of the text, and degrading the accuracy of the procedure. This is counteracted using a word sense disambiguation pre-processor.

The function of the pre-processor is as much to filter out spurious sense assignments as to wholly provide unique sense identifications. This increases the accuracy of Word Sense Disambiguation which is one of the effects of the lexical chaining process (Okumura and Honda 1994) by early elimination of inappropriate senses.

The pre-processor follows the "sliding window" approach described in Sussna, (Sussna 1993), where ambiguous words are examined within the several words of their surrounding local context. This is compatible with the Senseval task, (based as it is on lexical samples) and was consequently re-implemented for Senseval where it competed as SUSS.

SUSS's principal objective in Senseval was to evaluate different disambiguation techniques that could be used to improve the performance of a future version of Hesperus. This excludes both training, and deep linguistic analysis. Training, as in machine learning approaches (Rosenzweig, this volume) implies the existence of training corpora. Such corpora tend only to exist in limited subject areas, or are restricted in scope. Machine Learning approaches are consequently excluded since Hesperus is intended to be applicable to any subject area. Indeed, we could argue that the associations found in thesauri contain the most common representations and subsume the associations found in normal text. Deep linguistic analysis is rarely robust, and often slow. This makes it incompatible with Hesperus, which is designed as a real-time system. A derived objective was to maximise the number of successful disambiguations--the key competition requirement!

SUSS extensively exploited the Hector machine readable dictionary entries used in

Senseval. There were two reasons for this: Firstly, Hector dictionary entries are extremely rich, and allowed us to consider disambiguation techniques that would not have been possible using Roget's Thesaurus alone (as used in Hesperus). Secondly, Hector sense definitions were much finer grained than those used in Roget. A system that used Roget would consequently have been at a considerable disadvantage since it would not have been able to propose exact Hector senses in the competition.

One noteworthy technique made possible by the Hector dictionary was the conversion and adaptation of dictionary fields to patterns, as used in Information Extraction (e.g. Onyshkevych 1993, Riloff 1994). Where possible, this allowed the unique selection of a candidate word sense, with minimal impact on the performance of the rest of the algorithm.

2. SUSS: The Sunderland University Senseval System

SUSS is a multi-pass system that attempts to reduce the number of candidate word senses by repeated filtering. Following an initialisation phase, different filters are applied to select a preferred sense tag.

The order of filter application is important. Word and sense specific techniques are applied first, more general techniques are used if these fail. Specific techniques are not likely to affect any other than their prospective targets, whereas general methods introduce probable misinterpretation over the entire corpus. For example, a collocate such as "brass band" uniquely identifies that sense of "band", with no impact on other word senses. Other techniques required careful assessment to ensure that their overall effect was positive. This was part of a structured development strategy.

The SUSS development strategy.

We extensively exploited the training data to develop SUSS. Not to train the system by setting parameters on different features, but to ensure that promising techniques for some types of ambiguity did adversely influence the overall performance of the system.

The strategy was as follows:

- 1 A basic system was implemented that processed the training data.
- 2 A statistics module was implemented that displayed disambiguation effectiveness by sense and percentage.
- 3 As different disambiguation techniques were developed effectiveness was measured on the whole corpus.
- 4 Techniques that improved performance (as measured by percentage successful disambiguations over the whole corpus) were further developed. Those that degraded performance were dropped. (Since the competition was time limited it was not cost effective to pursue interesting but unsuccessful approaches.)

SUSS Algorithm

Initialisation Phase:

- 1 **Load Dictionary** using perl SGML parser.
- 2 **Calculate Sense Occurrence Statistics** (see Note 1) from the Senseval training corpus
- 3 Foreach *EXAMPLE*
 - Eliminate Stopwords
 - Produce Window *w* words wide centred on word to be disambiguated

Processing Phase

- 1 For each *SAMPLE*:

- 2 Filter Possible Entries as Collocates.
(**DONE** if there is only one candidate sense.)
- 3 Filter remaining senses for Information Extraction Pattern.
(**DONE** if there is only one candidate sense.)
- 4 Filter remaining senses for Idiomatic Phrases.
(**DONE** if there is only one candidate sense.)
- 5 Eliminate Stopwords from sample.
- 6 Produce Window *w* words wide centred on word to be disambiguated
- 7 Foreach *EXAMPLE*
Match the sample window against the example window
Select the sense that has the highest example matching score.
- 8 If no unique match found, return the most frequent sense of those remaining from the training corpus (or first remaining dictionary entry – note 1).

Preparation

The Initialisation phase includes dictionary processing and other preparation that would otherwise be repeated. The HECTOR dictionary was loaded into memory using a public domain program that parses SGML instances. This made the definition available as an array of homographs that is further divided into an array of finer sense distinctions. Each of these contains the fields, such as the definition, part of speech information, plus examples of usage.

These examples are used in the "Example Comparison Filter" and the "Semantic Relations Filter" techniques (described below) are also prepared. These reduce the dictionary examples to a narrow window *W* words wide from which stopwords (Salton and McGill 1983) have been eliminated centred on the word to be disambiguated. This facilitates comparison with identically structured text windows produced from the test data.

We now go on to describe the specific techniques tested.

Collocation Filter

Collocations are short, set expressions which have undergone a process of lexicalisation. For example, consider the collocation 'brass band'. This expression, without context, is understood to refer to a collection of musicians, playing together on a range of brass instruments, rather than a band made of brass to be worn on the wrist. For these reasons it is possible for the Hector dictionary to define such expressions as distinct senses of the word.

Given the set nature of collocations, therefore, it was considered that to look for these senses early in the disambiguation process would be a simple method of identifying or eliminating them from consideration.

The collocation identification module, therefore, worked as a filter using simple string-matching. If a word occurrence passing through the module corresponded to one of the collocational senses defined in the dictionary it would be tagged as having that sense. If none of these senses were applicable, however, all senses taking a collocational form were filtered out.

Information Extraction Pattern Filter

The Information Extraction filter refers exclusively to enhancements to the Hector dictionary entries specifically to support Word Sense Disambiguation. The HECTOR dictionary is primarily intended for human readers. Many entries contain a clues field, or and in a restricted language that indicates typical usage. For example, phrases such as "learn at mother's knee, learn at father's knee, and variants", or "usu on or after". These can be converted into string matching patterns and successfully used to identify individual senses.

For example, "shake" contains the following:

<idi>shake in one's shoes, shake in one's boots</idi>

<clues>v/= prep/in pron-poss prep-obj/(shoes,boots,seat)</clues>

This can be used to convert the idiom field (using PERL patterns) as follows:

<idi>shake in \w (shoes|boots|seat)</idi>*

This may now be used to match against any of the idiomatic expressions "shake in her boots", "your boots", etc.

We call a related method "phrasal patterns". A phrasal pattern is a non idiomatic multiple word expression that strongly indicates use of a word in a particular sense. For example, "shaken up" seems to occur only in past passive forms. Adding appropriate phrasal patterns to a dictionary sense was found to increase disambiguation performance for that sense.

These methods are important since they are tightly focused on one word, and on one sense that word may be used in. They do not, if correctly designed, affect other word senses, and certainly can not influence the interpretation of other words.

Idiomatic Filter

Idiomatic forms identify some word senses. Unlike collocations, however, idiomatic expressions are not constant in their precise wording. This made it necessary to search for content words in a given order, rather than looking for a fixed string. Also due to the idioms variable nature, it was considered that the successful matching of a subset of the content words exceeding a certain threshold value would imply that the form was contained in the text.

Dictionary entries that contained idiomatic forms were processed as follows: Firstly, two word idioms were checked for specifically. If the idiom was longer, stopwords were removed from the idiomatic form listed, and remaining content words compared in order with words occurring in the text. If 60% of the content words were found in the region of the target word, the idiomatic filter succeeded, and senses containing that idiom selected. Otherwise senses containing that idiomatic form were excluded from further consideration.

Example Comparison Filter.

The Example Comparison Filter tries to match the examples given in the dictionary against the word to be disambiguated, looking at the local usage context. It assigns a score for each sense on the basis of identical words occurring in the text and dictionary examples and their relative positions. We take a window of words surrounding the target word, with a specified width and specified position of the target, in the text and in a similar window from each dictionary example.

For each example in each sense, all the words occurring in each window are compared and, where identical words are found, a score, S , is assigned, where

$$S = \sum_{w \in W} d_S d_E$$

and w is a word in window W , and d_S and d_E are functions of the distance of the word from the target word in the sample and example windows respectively, such that greater distances result in lower scores.

When all the example scores have been calculated for each word sense, the sense with the highest example score is chosen as the correct sense of that occurrence.

In cases where this does not produce a result, the most frequently occurring sense (or first dictionary sense -- see Note 1) that has not been previously eliminated is chosen.

Other Techniques Evaluated.

One of the objectives of SUSS was to evaluate different disambiguation techniques. Below we describe two methods that were evaluated, but not used in the final system, since they lead

to decreased overall performance.

Part of Speech Filter

Wilks and Stevenson (1996) have claimed that much of sense tagging may be reduced to part-of-speech tagging. Consequently, we used the Brill Tagger (Brill 1992) on the subset of the training data set that required part-of-speech discrimination. This should have improved disambiguation performance by filtering out possible senses not appropriate to the assigned part of speech. However, due to tagging inaccuracy, this was just as likely to eliminate the correct word sense too. Consequently, it did not make a positive contribution.

Another routine which used the part-of-speech tags, attempted to filter out the senses of words marked as noun modifiers by the dictionary grammar labels where the following word was not marked as a noun by the tagger. This routine also checked words which contained an 'after' specification in the grammar tag and eliminated these senses where the occurrence did not follow the word given. This routine provided no overall benefit to the results either. One possible cause of this is in occurrences where there are two modifiers joined by a conjunction so that the first is, legitimately, not followed immediately by a noun.

Semantic Relations Filter

The Semantic Relations Filter is an extension of the example comparison filter that uses overlapping categories and groups in Roget's thesaurus, rather than identical word matching. This should allow us to recognise that "accident" is used in the same sense in "car accident", and "motor-bike accident", since both are means of transport.

Appropriate scores are allocated for each category in Roget that the test sentence window has in common with the dictionary example window. As in the example comparison, the sense that contains the highest scoring example is selected as the best.

Disappointingly, this technique finds many spurious relations where words in the local context are interpreted ambiguously. This led to an overall performance degradation over the test set, and so the technique was not part of the final SUSS algorithm.

3. Discussion and Conclusion

SUSS consisted of a number of individual disambiguation techniques that were applied to the data sequentially. Each of these techniques were designed to have one of two effects; either to attempt to assign a unique dictionary sense for the occurrence, or to eliminate one or more invalid senses from consideration.

During development a range of techniques were tested to determine whether they were effective in increasing the disambiguation accuracy of the algorithm. The testing procedures utilised the training data, with the algorithm being applied both with and without the technique activated. The results of these applications were compared over different senses and different words, against the human tagged training data.

The statistics produced were used to determine whether the technique improved the overall accuracy of the disambiguation and, hence, whether it was a useful technique. Some techniques, for example, produced a great improvement in accuracy on particular words or specific senses, yet the overall effect was a reduction in accuracy over all words.

This result reflects the interaction between word specific and generic sense disambiguation methods. A generic disambiguation technique needs to have better accuracy than that would be given by selecting a default sense. For example, in the training corpus, "wooden" means "made of wood" in 95% of the samples. Thus a generic technique applied to "wooden" needs to exceed this level of accuracy. If it does not, it will degrade overall performance.

Regular Information Extraction patterns provided a particularly effective sense specific disambiguation. However, it was necessary to convert each pattern by hand. A clear

next step would be the development of a module to automatically produce the patterns from the relevant dictionary fields.

SUSS performed surprisingly well considering its lack of sophistication, (see Rosenzweig, this volume) with above average performance compared to other systems. It is particularly interesting to note that it was placed in the first three systems where no training data was supplied.

4. Notes

The Calculation of Sense Occurrence Statistics was designed to counter a perceived deficiency in Hector, where the ordering of senses did not appear to match that of sense frequency in the corpus. This was considered to be a training technique, and SUSS was classified as a learning system. The SUSS-Dictionary system did not use this technique and was considered as an "all words" system.

5. References

- Brill, E. (1992). *A simple rule-based part-of-speech tagger*. Proceeding of the Third Conference on Applied Natural Language Processing. Trento, Italy.
- Ellman J, and Tait J. *"Using Information Density to Navigate the Web"* UK ISSN 0963-3308 IEE Colloquium on Intelligent World Wide Web Agents. March 1997
- Okumura M. and Honda T "Word Sense Disambiguation and text segmentation based on lexical cohesion" Proc COLING 1994 vol 2 pp 755-761
- Onyshkevych B *Template design for Information Extraction* Proceeding of the Fifth Message Understanding Conference (MUC-5)1993
- Morris, J. and Hirst, G. (1991). Lexical Cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1), pp21-48.
- Riloff, E. & Lehnert, W. *Information extraction as a basis for high-precision text classification* ACM Transactions on Information Systems Vol.12, No. 3 (July 1994), pp. 296-333
- Salton, G. and McGill, M. (1983), *Introduction to Modern Information Retrieval*. McGraw-Hill.
- Stairmand, M. (1996). Unpublished PhD Thesis. Dept of Computational Linguistics UMIST.
- St-Onge, D. (1995). *Detecting and Correcting Malapropisms with Lexical Chains*. MSc Thesis, University of Toronto. (Available via WWW).
- Sussna, M. (1993). *Word Sense Disambiguation for Free-text Indexing Using a Massive Semantic Network*. Proceedings of the Second International Conference on Information and Knowledge Base Management, pp67-74.
- Wilks, Y. and Stevenson, M. (1996). *The grammar of sense: Is word-sense tagging much more than part-of-speech tagging?* Technical Report CS-96-05, University of Sheffield.