ROMANIAN ACADEMY
Research Institute for Artificial Intelligence

Word Sense Disambiguation Methods Applied to English and Romanian

Radu ION

Advisor
prof. dr. **Dan TUFIŞ**
Corresponding Member of the Romanian Academy

Bucharest, May 2007

Contents

Key words

Word Sense Disambiguation, WSD, knowledge-based WSD, unsupervised WSD, lexical alignment, word alignment, parallel corpora, WordNet, lexical semantic networks, lexical ontology, SUMO/MILO, IRST domains, Inter-Lingual Index, ILI, sentence splitting, word splitting, token splitting, Named Entity Recognition, NER, Part of Speech tagging, POS tagging, lemmatization, Lexical Attraction Models, Meaning Text Model, MTM, dependency parsing.

1. Introduction

Word Sense Disambiguation (WSD) is a research problem of the broader Artificial Intelligence field, Natural Language Processing and is concerned with the automatic identification of the meaning of a word in its context. WSD was practically invented in the early experiments on Machine Translation in which the meaning of the source word had to be known in order to generate its translation accordingly.

Word Sense Disambiguation became useful in many other areas of Natural Language Processing such as question answering systems, speech transcription, document classification and especially, in natural language understanding systems.

WSD is known to be AI-complete. That is, it cannot be solved unless the other "hard" problems of AI are solved. Among these, Knowledge Representation plays a central role with a special emphasis on the so-called "common sense" knowledge. Existing WSD methods can only approximate the human capacity of disambiguating words in their contexts by modeling empirical assumptions of this capacity. One of the most influential assumptions is that the meaning of a word depends on its context of occurrence.

Context identification and formalization is one of the main problems to be dealt with when attempting to implement a WSD algorithm. Some WSD methods consider that the context of a word can be viewed as a window of words centered on the target word (the "bag of words" model of context). Others impose restrictions on this window such as the order in which context words appear with respect to the target word or the relevance level of these words to the target word. Parallel WSD has one great advantage over the bag of words formalization of the context: the context of the target word becomes the translation equivalent(s) of it into the language(s) of the parallel corpus (thus eliminating much of the noise of the bag of words model).

This work is concerned with simple (traditional) WSD on simple texts and also with WSD performed on parallel corpora. On the traditional side of the problem we are interested in studying the effect of some kind of syntactic analysis of the sentence as context formalization. On the other side, we are keen to explore the possibility of specifying contexts as lists of translation equivalents.

Syntactic analysis as a context representation is not new in the realm of WSD algorithms. By and large, constituent grammars have been used to parse the sentences

thus specifying the correspondences between words. By their very nature, constituent grammars are generative grammars that ultimately try to explain the surface form of the sentences without much consideration as to the (formal) correspondence between syntax and semantics. On the other hand, dependency formalisms such as Mel'cuk's Meaning Text Model (MTM) treat the surface form of the syntactic analysis as a means to get the final goal, that of the semantic representation of it.

In this work, we will approximate the dependency syntactic relation of MTM with a constrained Lexical Attraction Model that produces a link structure of a sentence (a connected, planar, acyclic and undirected graph with the sentence words as vertices). This approximation will certainly not be a better representation than the proper dependency analysis but it has one undeniable advantage: can be automatically obtained from free running texts with little or no processing at all.

As to the parallel corpora WSD, we will be interested to investigate the degree in which different meanings of a word translate with different translation equivalents and how this property can be utilized in building a disambiguation algorithm.

2. Text preprocessing. NLP computational resources

To perform WSD, one needs to properly identify sentences, words, their parts of speech (POS) and lemmas as lemmas are recorded by the sense inventories. This chapter presents **TTL**, a Perl module developed by the author, which does all of the above and also Named Entity Recognition and chunking for English and Romanian (and more recently for Bulgarian but without an expert validation of the performances).

TTL functionalities are:

- Named Entity Recognition: achieved by extensive use of Perl regular expressions that define sequences of tokens that constitute named entities. The named entities are: integers, real numbers, dates, times, names of persons (female and male), different quantities, lengths, volumes and weights;
- Sentence splitting: TTL maintains a list of abbreviations which may pose problems with their final punctuation which is ambiguous between the end of the sentence, end of an abbreviation or both;
- Token splitting: at first the sentence is parted at space boundaries. Then, any affixed lexeme is stripped out from the token and finally, idiomatic expressions are recognized;
- Part of Speech tagging: follows closely the Hidden Markov Models POS tagger of Thorsten Brants with some supplementary heuristics in case of unknown words;
- Lemmatization: is lexicon based and for the unknown words (not in the lexicon) a statistical procedure is applied to select the best lemma of the given word form;
- Chunking: done with regular expressions over sequences of POS tags; available for English and Romanian.

This chapter also deals with the construction and processing of the parallel English-Romanian corpus **SemCor**. The Romanian translation was done at "Alexandru Ioan

Cuza" University of Iaşi. The corpus was preprocessed with TTL and then word-aligned with the YAWA lexical aligner. The sense transfer from English to Romanian follows closely the WSDTool procedure. From a total of 88874 occurrences of content words in Romanian, 54.54% received sense annotation by the transfer procedure.

Lastly, the Romanian WordNet lexical semantic network (developed within the BalkaNet EC funded project IST-2000-29388) is described along with its conceptual alignment to the reference lexical semantic network of English, the Princeton WordNet (version 2.0), developed by George Miller and his colleagues at Princeton University in the USA. This network will serve the purpose of the semantic inventory with which the WSD algorithms described here operate.

3. WSD on parallel corpora

This chapter introduces **YAWA**, a lexical aligner developed by the author and **WSDTool**, an unsupervised, knowledge-based WSD algorithm also developed by the author, that runs on parallel corpora and relies on lexical alignments produced by YAWA.

YAWA is a 4 stage lexical aligner that uses a translation equivalents dictionary to generate an initial, stage 1, "skeleton" alignment. Using this alignment, in stage 2 a language dependent module takes over and produces alignments of the remaining lexical tokens within aligned chunks (non-recursive noun phrases, prepositional phrases, verbal and adjectival/adverbial complexes). Stage 3 is specialized in aligning blocks of consecutive unaligned tokens and stage 4 deletes alignments that are likely to be wrong. YAWA's individual performance measure (F) is **81.22%** and it is a part of the **COWAL** combined lexical alignment platform that won the first place in the lexical alignment competition held with the occasion of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05) workshop on "Building and Using Parallel Texts: Data Driven Machine Translation and Beyond", Ann Arbor, USA.

WSDTool assumes that given a parallel corpus, the translations of the target word[1] in the languages of the parallel corpus reduce the semantic field[2] of the target word knowing that the translation preserves the meaning of the source word. Thus, the translation equivalents of the target word become a materialization of the context of the target word because the translator(s) chose them based on this context.

The WSDTool disambiguation procedure broadly follows these steps:

1. Identify all the translation equivalents of the target word using the lexical alignments provided by YAWA;
2. Compute the intersection D between all senses of the target word and its translation equivalents using the collection of conceptually aligned lexical semantic networks;
3. If D contains more than one sense, apply Hierarchical Clustering Analysis on the occurrences of the target word to eliminate the ambiguity.

[1] The word to be disambiguated.
[2] The set of meanings that a word can have.

The sense inventory used by WSDTool is the collection of conceptually aligned lexical semantic networks of English and Romanian and the sense annotation is done using Inter-Lingual Indexes (ILI) that are unique identifiers of the EQ-SYNONYM related English and Romanian concepts. This way, the sense annotation is performed in a uniform way and both in English and Romanian and we can see the associated concepts just by following the ILI pointer. The performances of WSDTool are presented in Figure 1. The measures were computed on the SemCor parallel corpus, for English and Romanian for 79595 occurrences of content words in English and 48392 occurrences of content words in Romanian.

	Mărime		Engleză(en)				Română(ro)			
	en	ro	P(%)	R(%)	F(%)	S/C	P(%)	R(%)	F(%)	S/C
ILI	115424	33421	70.217	66.882	68.509	1	53.478	49.805	51.576	1
SUMO	2008	1774	76.788	73.144	74.921	1	65.059	60.572	62.735	1
IRST	168	164	87.636	83.463	85.498	1.092	85.015	79.124	81.964	1.11

Figure 1 WSDTool performance measures on three sense inventories: ILI, SUMO categories and IRST domains. P is the precision, R is the recall and F is the usual combination between P and R. S/C is the average of the number of sense labels assigned to a content word and is a measure of the discriminating power of the algorithm.

4. WSD with syntactic dependency structures

WSD algorithms have used a variety of context formalizations in order to identify contexts that are specific to a certain meaning of the target word. The disambiguation process makes use of the context formalization of the target word in order to assign the meaning that is indicated by the context representation.

One of the most widely used context formalization is that of the context features. In a window of words centered on the target word, some of following properties of the window words have been used as context features[3]:

- The words themselves either reduced to their lemma forms or not;
- The POS tags of the words;
- Collocations with the target word.

The syntactic representation of the context of the target word is not new in the realm of the WSD algorithms. It has the advantage that the target word is related only with the "relevant" words in its context, words that have a direct influence on determining the target word's meaning. The syntactic dependency relation of the MTM model best describes this situation. However, here we will not be able to make use of it because a dependency parser is not available for Romanian. So instead of a fully-fledged syntactic dependency analysis, we will make use of a dependency-like structure of a sentence obtained through the use of a Lexical Attraction Model (LAM).

A LAM generates a dependency-like structure of a sentence (linkage) that is a connected, undirected, acyclic graph with sentence's words as its vertices. Moreover, the planarity property[4] is very frequent both in English and Romanian and as such, is

[3] This is of course not an exhaustive list.
[4] Edges drawn over the surface form of the sentence do not cross.

also imposed on the resulting graph. **LexPar**, is the author's implementation of a constrained LAM, that is, a LAM where not all possible links are allowed due to syntactic restrictions of combination such as agreement.

SynWSD is an unsupervised, knowledge-based WSD algorithm that uses the linkage of a sentence to estimate the meaning affinity between two linked words and also to disambiguate the words using the meaning affinity model constructed. It has two main phases:

1. **Training**: given a large corpus containing sentences that were TTL and LexPar processed, for each pair of linked words in every sentence, records the frequency of the pairs of meanings specific to each pair member;
2. **Disambiguation**: for a new input sentence that is also TTL and LexPar processed, search for the meaning configuration that maximizes meaning affinity as given by the trained model. The main original element here is that the sentence receives the best interpretation overall without regard to the order in which the words in it are processed.

Figure 2 gives the performance measures of SynWSD on the same data on which WSDTool was tested.

		Engleză				Română			
		P(%)	R(%)	F(%)	S/C	P(%)	R(%)	F(%)	S/C
ILI	dice	46.985	46.874	46.930	1.477	40.627	40.122	40.373	1.769
	prob	46.459	46.349	46.404	1.429	41.588	41.070	41.327	1.787
	mi	47.859	47.746	47.803	1.729	41.204	40.685	40.942	2.084
	ll	42.977	42.876	42.927	1.239	36.170	35.720	35.943	1.384
	int	**69.773**	26.638	38.556	1.163	**59.845**	22.214	32.401	1.373
	majv	43.952	43.848	43.900	1.285	37.212	36.747	36.978	1.493
	union	68.164	**68.001**	**68.082**	2.805	59.647	**58.896**	**59.269**	3.353
SUMO	dice	50.246	49.958	50.102	1.237	40.971	40.472	40.720	1.234
	prob	49.688	49.408	49.548	1.169	41.954	41.442	41.696	1.214
	mi	57.831	57.236	57.532	1.334	51.188	50.570	50.877	1.413
	ll	47.249	46.979	47.114	1.096	39.550	39.065	39.306	1.120
	int	**74.067**	32.503	45.180	1.009	**69.405**	25.609	37.413	1.008
	majv	48.135	47.915	48.025	1.087	39.566	39.084	39.323	1.092
	union	73.505	**73.165**	**73.335**	2.140	66.708	**65.901**	**66.302**	2.363
IRST	dice	78.042	77.658	77.849	1.090	77.461	76.516	76.986	1.089
	prob	76.351	75.974	76.162	1.018	76.685	75.749	76.214	1.032
	mi	75.437	74.983	75.210	1.274	65.235	64.440	64.835	1.276
	ll	75.735	75.359	75.546	1.010	76.140	75.210	75.672	1.004
	int	88.399	59.352	71.020	1.002	87.368	50.449	63.963	1.001
	majv	76.371	76.026	76.198	1.016	76.612	75.677	76.142	1.020
	union	**91.413**	**91.005**	**91.209**	1.621	**90.305**	**89.202**	**89.750**	1.719

Figure 2 SynWSD performance figures on English-Romanian SemCor. There are four meaning affinity functions (dice, prob, pointwise mutual information and log-likelihood) and three combination methods (intersection, majority voting and union).

5. Conclusions

This work focused on the presentation of all algorithms one needs to perform WSD on a free running text. **The author's contributions are**:

- The assembly of the **SemCor** English-Romanian parallel corpus with Romanian meaning annotation. SemCor is the reference corpus in testing English WSD algorithms;

- The development of the **TTL** free running text processing module capable of named entity recognition, sentence and token splitting, POS tagging, lemmatization and chunking. Sentence and token splitting, POS tagging and lemmatization are needed by any WSD algorithm;
- The development of **YAWA**, a lexical aligner in 4 stages. This aligner is needed by WSDTool to find the translation equivalents. **YAWA** is a part of the **COWAL** combined lexical alignment platform that **won the first place** in the lexical alignment competition held with the occasion of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05) workshop on "Building and Using Parallel Texts: Data Driven Machine Translation and Beyond", Ann Arbor, USA.
- The development of **WSDTool, an unsupervised, knowledge-based WSD solution that is able to disambiguate parallel corpora** using three different sense inventories: WordNet concepts, SUMO categories and IRST domains. Also, WSDTool was used at validating the correctness of conceptual alignment between Romanian WordNet and Princeton English WordNet;
- The development of **LexPar,** a constrained LAM that provides the dependency-like graph of the sentence to be disambiguated by SynWSD;
- The development of **SynWSD**, an unsupervised, **knowledge-based WSD solution that is able to disambiguate simple texts**, using as WSDTool, the three sense inventories. Recently, SynWSD participated in the 4th evaluation of WSD algorithms SensEval-4/SemEval-2007 where it ranked the 8th in the English All Words task out of 14 competing systems being outrun mostly by supervised WSD systems.

In his doctoral stage, the author has published **23 papers** and **3 abstracts** at leading conferences in the field of Computational Linguistics and Natural Language Processing. Some of these are:

- The Association for Computational Linguistics (ACL)[5];
- The North American Chapter of the Association for Computational Linguistics (NAACL)[6];
- The International Conference on Computational Linguistics (COLING)[7];
- The European Chapter of the Association for Computational Linguistics (EACL)[8];
- The Language Resources And Evaluation Conference (LREC)[9];
- The International Florida Artificial Intelligence Research Society Conference (FLAIRS)[10];
- Language Resources and Evaluation Journal, ISSN: 1574-020X (print version), Journal no. 10579, Springer Netherlands

[5] http://www.aclweb.org/
[6] http://www.cs.cornell.edu/home/llee/naacl/
[7] http://www.issco.unige.ch/coling2004/
[8] http://eacl.coli.uni-saarland.de/
[9] http://www.lrec-conf.org/
[10] http://www.flairs.com/