KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 41–49

A CHAIN DICTIONARY METHOD FOR WORD SENSE DISAMBIGUATION AND APPLICATIONS

DOINA TĂTAR(1), GABRIELA ŞERBAN(1), ANDREEA MIHIŞ(1), MIHAIELA LUPEA(1), DANA LUPŞA(1), AND MILITON FRENŢIU(1)

Abstract. A large class of unsupervised algorithms for Word Sense Disambiguation (WSD) is that of dictionary-based methods. Various algorithms have as the root Lesk's algorithm, which exploits the sense definitions in the dictionary directly. Our approach uses the lexical base WordNet [3] for a new algorithm originated in Lesk's, namely *chain algorithm for disambiguation* of all words (CHAD). We show how translation from a language into another one and also text entailment verification could be accomplished by this disambiguation.

1. The polysemy

Word sense disambiguation is the process of identifying the correct sense of words in particular contexts. The solving of WSD seems to be AI complete (that means its solution requires a solution to all the general AI problems of representing and reasoning about arbitrary) and it is one of the most important open problems in NLP [5],[6],[7], [10],[12],[13]. In the electronical on-line dictionary WordNet, the most well-developed and widely used lexical database for English, the polysemy of different category of words is presented in order as: the highest for verbs, then for nouns, and the lowest for adjectives and adverbs. Usually, the process of disambiguation is realized for a single, target word. One would expect the words closest to the target word to be of greater semantical importance for it than the other words in the text. The context is hence a source of information to identify the meaning of the polysemous words. The contexts may be used in two ways: a) as *bag of words*, without consideration of relationships with the target word in terms of distance, grammatical relations, etc.; b) with relational information. The *bag of words* approach works better for nouns than verbs but is less effective than methods that take other relations in consideration. Studies about syntactic relations determined some interesting conclusions: verbs derive more disambiguation

2000 *Mathematics Subject Classification.* 68T50,03H65.
Key words and phrases. WSD, machine translation, text entailment.

information from their objects than from their subjects, adjectives derive almost all disambiguation information from the nouns they modify, and nouns are best disambiguated by directly adjacent adjectives or nouns [5]. All these advocate that a global approach (disambiguation of all words) helps to disambiguate each POS.

In this paper we propose a global disambiguation algorithm called **chain algorithm** for disambiguation, CHAD, which presents elements of both points of view about a context: because this algorithm is *order sensitive* it belongs to the class of algorithms which depend of relational information; in the same time it doesn't require syntactic analysis and syntactic parsing.

In section 2 of this paper we review Lesk's algorithm for WSD. In section 3 we present "triplet" algorithm for three words and CHAD algorithm. In section 4 we describe some experiments and evaluations with CHAD. Section 5 introduces some conclusions of using the CHAD for translation (here from Romanian language to English) and for text entailment verification. Section 6 draws some conclusions and further work.

2. Dictionary-based methods

Work in WSD reached a turning point in the 1980s when large-scale lexical resources, such as machine readable dictionaries, became widely available. One of the best known dictionary-based method is that of Lesk (1986). It starts from the idea that a word's dictionary definition is a good indicator for the senses of this word and uses the definition in the dictionary directly.

Let us remember basic algorithm of Lesk [8]:

Suppose that for a polysemic target word w there are in a dictionary Ns senses $s_1, s_2, \cdots, s_{Ns}$ given in an equal number of definitions $D_1, D_2, \cdots, D_{Ns}$. Here we mean by D_i the set of words contained in the i-th definition.

Consider that the new context to be disambiguated is c_{new}. The **reduced form** of Lesk's algorithm is:

for $k = 1, Ns$ *do*
  $score(s_k) =\mid D_k \cap (\cup_{v_j \in c_{new}} \{v_j\}) \mid$
endfor
Calculate $s' = argmax_k score(s_k)$

The score of a sense is the number of words that are shared by the different sense definitions (glosses) and the context. A target word is assigned that sense whose gloss shares the largest number of words.

The algorithm of Lesk was successfully developed in [2] by using WordNet dictionary for English. It was created by hand in 1990s and includes definitions (glosses) for individual senses of words, as in a dictionary. Additionally it defines groups of synonymous words representing the same lexical concept (synset) and organizes them into a conceptual hierarchy. The paper [2] uses this conceptual hierarchy

for improving the original Lesk's method by augmenting the definitions with non-gloss information: synonyms, examples and glosses of related words (hypernyms, hyponyms). Also, the authors introduced a novel overlap measure between glosses which favorites multi-word matching.

3. Chain algorithm for word sense disambiguation - CHAD.

First of all we present an algorithm for disambiguation of a triplet. In a sense, our triplet algorithm is similar with global disambiguation algorithm for a window of two words around a target word given [2]. Instead, our CHAD realizes disambiguation of all-words in a text with any length, ignoring the notion of "window" and "target word" and target word in similar studies, all that without increasing the computational complexity.

The algorithm for disambiguation of a triplet of words $w_1w_2w_3$ for Dice measure is the following:

begin
  for each sense $s^i_{w_1}$ do
    for each sense $s^j_{w_2}$ do
      for each sense $s^k_{w_3}$ do
        $score(i,j,k) = 3 \times \frac{|D_{w_1} \cap D_{w_2} \cap D_{w_3}|}{|D_{w_1}| + |D_{w_2}| + |D_{w_3}|}$
      endfor
    endfor
  endfor
  $(i^*, j^*, k^*) = argmax_{(i,j,k)} score(i,j,k)$ /* sense of w_1 is $s^{i^*}_{w_1}$, sense of w_2 is $s^{j^*}_{w_2}$, sense of w_3 is $s^{k^*}_{w_3}$ */
end

For the overlap measure the score is calculated as: $score(i,j,k) = \frac{|D_{w_1} \cap D_{w_2} \cap D_{w_3}|}{min(|D_{w_1}|, |D_{w_2}|, |D_{w_3}|)}$

For the Jaccard measure the score is calculates as: $score(i,j,k) = \frac{|D_{w_1} \cap D_{w_2} \cap D_{w_3}|}{|D_{w_1} \cup D_{w_2} \cup D_{w_3}|}$

Shortly, CHAD begins with the disambiguation of a triplet $w_1w_2w_3$ and then adds to the right the following word to be disambiguated. Hence it disambiguates at a time a new triplet, where first two words are already associated with the best senses and the disambiguation of the third word depends on these first two words. CHAD algorithm for disambiguation of the sentence $w_1w_2...w_N$ is:

begin
  Disambiguate triplet $w_1w_2w_3$
  $i = 4$
  while $i \leq N$ do
    Calculate $score(s_i) = 3 \times \frac{|D^*_{w_{i-2}} \cap D^*_{w_{i-1}} \cap D^{s_i}_{w_i}|}{|D^*_{w_{i-2}}| + |D^*_{w_{i-1}}| + |D^{s_i}_{w_i}|}$
    Calculate $s^*_i := argmax_{s_i} score(s_i)$
    $i := i + 1$
  endwhile
end

Due to the brevity of definitions in WN many values of $\mid D^*_{w_{i-2}} \cap D^*_{w_{i-1}} \cap D^{s_i}_{w_i} \mid$ are 0. We attributed the first sense in WN for s^*_i in this cases.

4. Some experiments with chain algorithm. Experimental evaluation of **CHAD**

In this section we shortly describe some experiments that we have made in order to validate the proposed chain algorithm **CHAD**.

4.1. **Implementation details.** We have developed an application that implements **CHAD** and can be used to:

- disambiguate words (4.2);
- translate words into Romanian language (5.1);
- text entailment verification (5.2).

The application is written in JDK 1.5.0. and uses *HttpUnit* 1.6.2 API [15]. Written in Java, HttpUnit is a free software that emulates the relevant portions of browser behavior, including form submission, JavaScript, basic http authentication, cookies and automatic page redirection, and allows Java test code to examine returned pages either as text, an XML DOM, or containers of forms, tables, and links [15].

We have used *HttpUnit* in order to search WordNet through the dictionary from [16]. More specifically, the following Java classes from [15] are used:

- *WebConversation*. It represents the context for a series of HTTP requests. This class manages cookies used to maintain session context, computes relative URLs, and generally emulates the browser behavior needed to build an automated test of a web site.
- *WebResponse*. This class represents a response to a web request from a web server.
- *WebForm*. This class represents a form in an HTML page. Using this class we can examine the parameters defined for the form, the structure of the form (as a DOM), and the text of the form. We have used *WebForm* class in order to simulate the submission of the form with corresponding parameters.

4.2. **Results.** We tested our CHAD on 10 files of Brown corpus, which are POS tagged. Recall that WN stores only stems of words. So, we first preprocessed the glosses and the input files, replacing inflected words with their stems.

The reason for choosing Brown corpus was the possibility offered by SemCor corpus (the best known publicly available corpus hand tagged with WN senses) to evaluate the results. The correct disambiguated words means the disambiguated words as in SemCor. We ran separately CHAD for: 1. nouns, 2. verbs, and 3. nouns, verbs, adjectives and adverbs. In the case of CHAD addressed to nouns, the output is the sequence of nouns tagged with senses. The tag *noun#n#i*

means that for noun *noun* the WN sense i was found. Analogously for the case of disambiguation on verbs and of all POS. The results are presented in tables 1 and 2. As our CHAD algorithm is dependent on the length of glosses, and as nouns have the longest glosses, the highest precision is obtained for nouns. In Figure 3, the Precision Progress can be traced. By dropping and rising, the precision finally stabilizes to value 0.767 (for the file Br-a01). The most interesting part of this graph is that he shows how this Chain Algorithm works and how the correct or incorrect disambiguation of first two words from the first triplet influences the disambiguation of the next words.

It is known that, at Senseval 2 contest, only 2 out of the 7 teams (with the unsupervised methods) achieved higher precision than the WordNet 1^{st} sense baseline. We compared in figures 1 , 2 and 3 the precision of CHAD for 10 files in Brown corpus, for Dice, Overlap and Jaccard measures with WordNet 1^{st} sense.

Comparing the precision obtained with the Overlap Measure and the precision given by the WordNet 1^{st} sense for 10 files of Brown corpus (Br-a01, Br-a02, Br-11, Br-12, Br-13, Br-14, Br-a15, Br-b13, Br-b20 and Br-c01), we obtained the following results:

- for Nouns, the minimum difference was 0.0077, the maximum difference was 0.0706, the average difference was 0.0338;
- as a whole, for 4 files difference was greater or equal to 0.04, and for 6 files was lower;
- in case of all Parts of Speech, the minimum difference was 0.0313, the maximum difference was 0.0681, the average difference was 0.0491;
- as a whole, for 7 files difference was greater or equal to 0.04, and for 3 files was lower;
- relatively to Verbs, the minimum difference was 0.0078, the maximum difference was 0.0591, the average difference was 0.0340;
- as a whole, for 4 files difference was greater or equal to 0.04, and for 6 files was lower.

Let us remark that in our CHAD the standard concept of windows better size parameter [2] is not working: simply, a window is the variable space between the previous and the following word in respect to the current word.

5. Applications of CHAD algorithm

5.1. **Application to Romanian-English translation.** WSD is only an intermediate task in NLP. In Machine Translation WSD is required for lexical choise for words that have different translation for different senses and that are potentially ambiguous within a given document. However, most Machine Translation models do not use explicit WSD [1] (in Introduction). The algorithm implemented by us consists in the translation word by word of a Romanian text (using dictionary at http://lit.csci.unt.edu/≈rada/downloads/RoNLP/R.E. tralexand), then the application of chain algorithm to the English text. As the translation of a

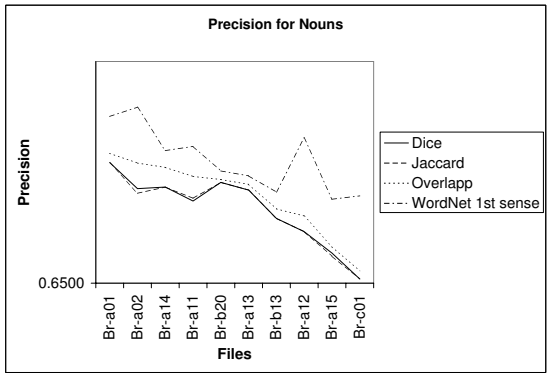

FIGURE 1. Noun Precision

File	Words	Dice	Jaccard	Overlap	WN1
Bra01	486	0.758	0.758	0.767	0.800
Bra02	479	0.735	0.731	0.758	0.808
Bra14	401	0.736	0.736	0.754	0.769
Bra11	413	0.724	0.726	0.746	0.773
Brb20	394	0.740	0.740	0.743	0.751
Bra13	399	0.734	0.734	0.739	0.746
Brb13	467	0.708	0.708	0.717	0.732
Bra12	433	0.696	0.696	0.710	0.781
Bra15	354	0.677	0.674	0.682	0.725
Brc01	434	0.653	0.653	0.661	0.728

TABLE 1. Precision for Nouns, sorted descending by the precision of Overlap measure

Romanian word in English is multiple, the disambiguation of a triplet is modified as following. Let be the word w_1 with k_1 translations $t^m_{w_1}$, the word w_2 with k_2 translations $t^n_{w_2}$ and the word w_3 with k_3 translations $t^p_{w_3}$. Each triplet $t^m_{w_1} t^n_{w_2} t^p_{w_3}$ is disambiguated with the triplet disambiguation algorithm and then the triplet with the maxim score is selected:

begin
 for $m = 1, k_1$ do
  for $n = 1, k_2$ do
   for $p = 1, k_3$ do
    Disambiguate triplet $t^m_{w_1} t^n_{w_2} t^p_{w_3}$ in $(t^m_{w_1})^*(t^n_{w_2})^*(t^p_{w_3})^*$
    Calculate $score((t^m_{w_1})^*(t^n_{w_2})^*(t^p_{w_3})^*)$
   endfor
  endfor
 endfor

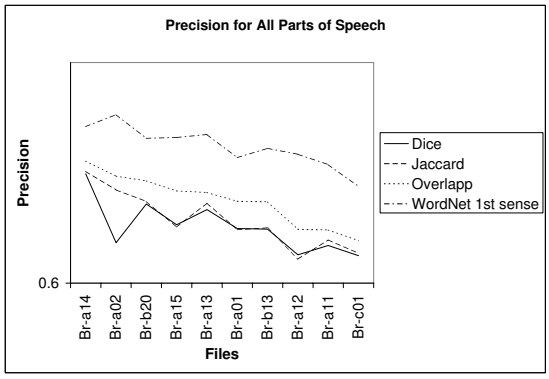

FIGURE 2. All Parts of Speech Precision

File	Words	Dice	Jaccard	Overlap	WN1
Bra14	931	0.699	0.701	0.711	0.742
Bra02	959	0.637	0.685	0.697	0.753
Brb20	930	0.672	0.674	0.693	0.731
Bra15	1071	0.653	0.651	0.684	0.732
Bra13	924	0.667	0.673	0.682	0.735
Bra01	1033	0.650	0.648	0.674	0.714
Brb13	947	0.649	0.650	0.674	0.722
Bra12	1163	0.626	0.622	0.649	0.717
Bra11	1043	0.634	0.639	0.648	0.708
Brc01	1100	0.625	0.627	0.638	0.688

TABLE 2. Precision for all POS, sorted descending by the precision of Overlap measure

Calculate $(m*, n*, p*) = argmax_{(m,n,p)} score((t^m_{w_1})^*(t^n_{w_2})^*(t^p_{w_3})^*)$
Optimal translation of triplet is $(t^{m*}_{w_1})^*(t^{n*}_{w_2})^*(t^{p*}_{w_3})^*$
end

Let us remark that $(t^{m*}_{w_1})^*$, for example, is a synset which corresponds to the best translation for w_1 produced by CHAD algorithm. However, since in Romanian are used many words linked by different spelling signs, these composed words are not found in the Romanian-English dictionary. Accordingly, not each Romanian word produces an English correspondent as output of the above algorithm. However, many translations are still correct. For example, the translation of expression *vreme trece* (in the poem "Glossa" of our national poet Mihai Eminescu), is *Word: (Rom)vreme (Eng)Age#n#4 , Word: (Rom)trece (Eng)Flow#v#1* . As another example from the same poem, where the synset of a word occurs (as an output of our application), *ţine toate minte*, is translated in *Word: (Rom) tine (Eng)*

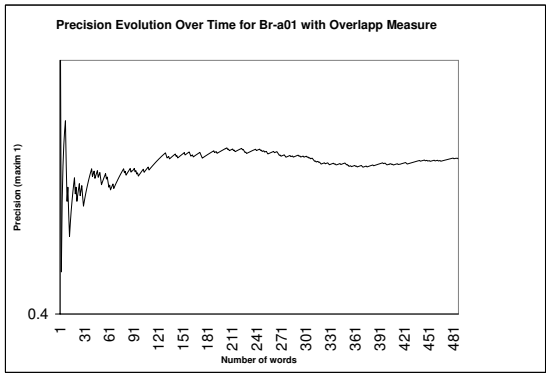

FIGURE 3. Precision in progress

Keep#v#8 :{keep, maintain}, Word: (Rom) toate (Eng) All#adv#3 :{wholly, entirely, completely, totally, all, altogether, whole}, Word: (Rom) minte (Eng) Judgment#n#2 :{judgment, judgement, assessment}.

5.2. **Application to text entailment verification.** The recognition of text entailment is one of the most complex task in Natural Language Understanding [14]. Thus, a very important problem in some computational linguistic applications (as question answering, summarization, segmentation of discourse, and others) is to establish if a text *follows* from another text. For example, a QA system has to identify texts that entail the expected answer. Similarly, in IR the concept denoted by a query expression should be entailed from relevant retrieved documents. In summarization, a redundant sentence should be entailed from other sentences in the summary. The application of WSD to text entailment verification is treated by authors in the paper "Text entailment verification with text similarity" in this Volume.

6. Conclusions and further work

In this paper we presented a new algorithm of word sense disambiguation. The algorithm is parametrized for: 1. all words (that means nouns, verbs, adjectives, adverbs); 2. all nouns; 3. all verbs. Some experiments with this algorithm for ten files of Brown corpus are presented in section 4.2. The stemming was realized using the list from http://snowball.tartarus.org/algorithms/porter/diffs.txt. The precision is calculated relative to the corresponding annotated files in SemCor corpus. Some details of implementation are given in 4.1.

We showed in section 5 how the disambiguation of a text helps in automated translation of a text from a language into another language: each word in the first text is translated into the most appropriated word in the second text. This

appropriateness is considered from two points of view: 1. the point of view of possible translation and 2. the point of view of the real sense (disambiguated sense) of the second text. Some experiments with Romanian - English translations and text entailment verification are given (section 5).

Another problem which we intend to address in the further work is that of optimization of a query in Information Retrieval. Finding whether a particular sense is connected with an instance of a word is likely the IR task of finding whether a document is relevant to a query. It is established that a good WSD program can improve performance of retrieval. As IR is used by millions of users, an average of some percentages of improvement could be seen as very significant.

References

[1] E. Agirre and P. Edmonds (editors). 2006. *WSD: Algorithms and Applications*. Springer.
[2] S. Banarjee and T. Pedersen. 2003. *Extended Gloss Overlaps as a Measure of Semantic Relatedness*. Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, August 9-15, Acapulco, Mexico.
[3] C. Fellbaum (editor). 1998. *WordNet An Electronic Lexical Database*. The MIT Press.
[4] S. Harabagiu and D.Moldovan. 1999. *A parallel system for Textual Inference*. IEEE Transactions parallel and distributed systems, 10(11), 254–270.
[5] N. Ide and J. Veronis. 1998. *Introduction to the special issue on WSD: the state of the art*. Computational Linguistics, 24(1):1–40.
[6] D. Jurafsky and J. Martin. 2000. *Speech and language processing*. Prentice Hall.
[7] A. Kilgarriff. 1997. *What is WSD good for?* ITRI Technical Report Series- August.
[8] C. Manning and H. Schutze. 1999. *Foundation of statistical natural language processing*. MIT.
[9] T. Pedersen, S. Patwardhan, and J. Michelizzi. 2004. *Wordnet::Similarity-measuring the relatedness of concepts*. 1024–1025.
[10] P. Resnik and D. Yarowsky. 1998. *Distinguishing Systems and Distinguishing sense: new evaluation methods for WSD*. Natural Language Engineering, 1(1).
[11] V. Rus. 2001. *Logic form transformation for WordNet glosses and its applications*. PhD Thesis, Southern Methodist University, CS and Engineering Department.
[12] G. Serban and D. Tatar. 2004. *UBB system at Senseval3*. Proceedings of Workshop in Word Disambiguation, ACL 2004, Barcelona, July, 226–229.
[13] D. Tatar and G. Serban. 2001. *A new algorithm for WSD*. Studia Univ. Babes-Bolyai, Informatica, 2, 99–108.
[14] D. Tătar and M. Frenţiu. 2006. *Textual inference by theorem proving and linguistic approach*. Studia Universitatis Babes-Bolyai, Informatica, LI(2), 31–41.
[15] http://httpunit.sourceforge.net/, 2006.
[16] http://wordnet.princeton.edu/perl/webwn, 2006.

(1) Babes-Bolyai University, Cluj-Napoca

E-mail address: dtatar@cs.ubbcluj.ro, gabis@cs.ubbcluj.ro
E-mail address: mihis@cs.ubbcluj.ro, mlupea@cs.ubbcluj.ro
E-mail address: dlupsa@cs.ubbcluj.ro, mfrentiu@cs.ubbcluj.ro