WSD based on three short context methods

C. de Loupy[1,2], M. El-Bèze[1] and P.-F. Marteau[2]

(1) Laboratoire d'Informatique d'Avignon (LIA)
339 ch. des Meinajaries, BP 1228
F-84911 Avignon Cedex 9 (France)
{claude.loupy,marc.elbeze}@lia.univ-avignon.fr

(2) Bertin & Cie
Z.I. des Gatines - B.P. 3
F-78373 Plaisir cedex
{deloupy,marteau}@bertin.fr

Abstract: *This article describes three methods for Word Sense Disambiguation using short context information. The first one is based on Semantic Classification Trees and the second and the third ones are a pattern-matching-like method. Their application within the SENSEVAL evaluation is shown here. Our participation in the ROMANSEVAL evaluation is described at the end of the article.*

Keywords: word sense disambiguation, semantic classification trees, WSD evaluation, SENSEVAL, ROMANSEVAL.

1 Introduction

Since the beginning of Natural Language Processing, Word Sense Disambiguation (WSD) has been considered as one of the most important problems. Recently, the amount of research on WSD has grown dramatically [Computational Linguistics, 1998], thanks to the availability of new semantic resources like WordNet [Miller et al., 1993a], and the increase in calculation capacity. Besides, the success of the First International Conference on Language Resources & Evaluation, held in Granada (28-30 May 1998) is an illustration of the growing interest of the research community in the evaluation paradigm. The SENSEVAL [Kilgarriff, 1998] and ROMANSEVAL projects are the natural consequences of both.

Within the framework of the development of a set of tools for textual database management [Loupy et al., 1998a], the Laboratoire Informatique d'Avignon and Bertin work on semantic disambiguation. Different methods are used and compared. As an initial step, we developed a tool based on different methods using Hidden Markov Models (HMM) and WordNet [Loupy et al., 1998b]. After several attempts to improve the HMM results, we came to the consideration that pattern matching could provide an answer to our expectations. Therefore, we have chosen to try three methodologies for SENSEVAL; the first one is based on Semantic Classification Trees (SCT) [Kuhn & De Mori, 1995], the second and third ones could be seen as pattern matching approaches. The models are based only on the lemmas, and not on a composition of pattern matching and semantic tags.

In this article, are described the model and the training phase, the SCT method, the pattern matching one, and their application in SENSEVAL. Lastly, is briefly mentioned our work in the ROMANSEVAL project. Since no training data were available for this task, the method we used must be seen as a study. Any comparison with other systems is beyond the scope of the present paper.

2 Preparation of the data

The methods used require a training corpus. We have only worked on nouns and on tagged words. Then, the evaluation of the proposed approach has been performed on 11 different nouns.

2.1 Some figures

In order to train the models, we have used the examples given by the dictionary and the training corpus. The following table gives, for each lemma of the evaluation, the number of semantic tags they have in the dictionary (this number takes into account suffixes *-x -m -p* and *?*), and the number of their occurrences in the dictionary (DIC) examples, in the training corpus (TRAIN) and in the test corpus (TEST).

Mot	Tags	DIC	TRAIN	Total	TEST
accident	16	30	1236	1266	267
behaviour	5	7	956	963	268
bet	19	36	110	146	275
excess	10	22	178	200	186
giant	11	24	343	367	118
knee	25	42	406	448	251
onion	5	15	26	41	214
promise	8	31	584	615	113
sack	12	13	60	73	51
scrap	17	39	20	59	156
shirt	16	22	508	530	181
TOTAL	150	283	4511	4794	2125

If we look at the ratio between the number of examples and the number of semantic tags, *scrap* seems to be very poor in training data and *behaviour* very rich. But the number of examples is not so important for WSD.

Here, the variety is preferable: one seeks for the larger coverage of the possible contexts of one word, to be able to identify its sense in a new document.

It would be interesting to have more data in order to calculate the difficulty of the task, like in [Loupy et al., 1998b]. In fact, it is very difficult to compare the different methods, those based on WordNet and the SemCor [Miller et al., 1993b] and those based on Hector and the SENSEVAL corpora.

2.2 The training data

"Yarowsky […] suggests that local ambiguities need only a window of *k=3* or *4*, while semantic or topic-based ambiguities require a larger window of 20-50 words." [Ide & Véronis, 1998]. Therefore, for this evaluation task, we look at 3 lemmas before the ambiguous one (call it Λ) and 3 lemmas after. Consider a window:

$\lambda_{i-3}\ \lambda_{i-2}\ \lambda_{i-1}\ \lambda_i\ \lambda_{i+1}\ \lambda_{i+2}\ \lambda_{i+3}$. Most often, λ_i is the lemma we work on ($\lambda_i = \Lambda$). But if the system detects an end of sentence at the position (i+k), k varying in the range [1-4], Λ is moved to the position (i-4+k), in order to keep a 7 lemmas window (for k=1, 4 before and 2 after Λ). Likewise, if an end of sentence is encountered at the position (i-k), Λ is placed at the position (i+4-k).

If two possible tags are given for Λ in the example, the information is duplicated: one for each tag.

DIC and TRAIN are treated exactly the same way and all the examples have the same importance for training, whatever their origin.

2.3 Preprocessing

In order to improve WSD, it is important to know the grammatical tag of the words. For such a task, we have used ECSTA [Spriet, El-Bèze, 1997]. Yarowsky [1993] determines various behaviors based on syntactic categories: directly adjacent adjectives or nouns best disambiguate nouns. Our assumption is quite different: we would like to check to which extent verbs and nouns could disambiguate nouns. The words belonging to four grammatical classes are therefore not kept for the noun disambiguation process: possessive adjectives, determiners, adverbs and adjectives. The other words are replaced by their lemma and unknown words are left unchanged.

Some words are so highly related that, in almost all the cases, it is possible to change one of them by another without consequence for the sense of the phrase. For instance, it is not necessary to keep a precise information on months. These words are grouped in a class represented by a pseudo-lemma. January, February, etc. are replaced by a pseudo-lemma: MONTH. By the same way, the pseudo-lemma DAY replaces Monday or Tuesday, etc. CD stands for a number, PRP for a pronoun, NNPL for a location (Paris, France, etc.), NNPG for a group (a company, an association, etc.), NNP for the name of a person and UNK for an Out of Vocabulary Word if its initial letter is an uppercase. All these substitutions intend to decrease the number of possible examples for one sense.

For example, if we consider *onion*, some of the examples extracted from the dictionary are given here. In the definition of the sense ***528376***, we find:
the multicoloured onion domes of St Basil's Cathedral.
This sentence is used to produce the following context example of (onion, **528376**):
- / ***onion*** / *dome* / *of* / *UNK* / *basil* / *'s* / *cathedral* / : sense **528376**.
UNK represent the out of vocabulary word *St*.
By the same way, we have:
- / *with* / *potato* / *and* / ***onion*** / *float* / *in* / *parsley* /: sense **528347**.

From the training corpus, we can extract:
- / *and* / *carrot* / *and* / ***onion*** / *and* / *lee*k / . /: sense ***528347***.
- / *as* / *lettuce* / *and* / ***onion*** / *be* / , / *while* /: sense ***528344***.

3 Semantic classification trees

We provide, here, a very short description of the SCT method. For more information, one will refer to [Kuhn & De Mori, 1995].

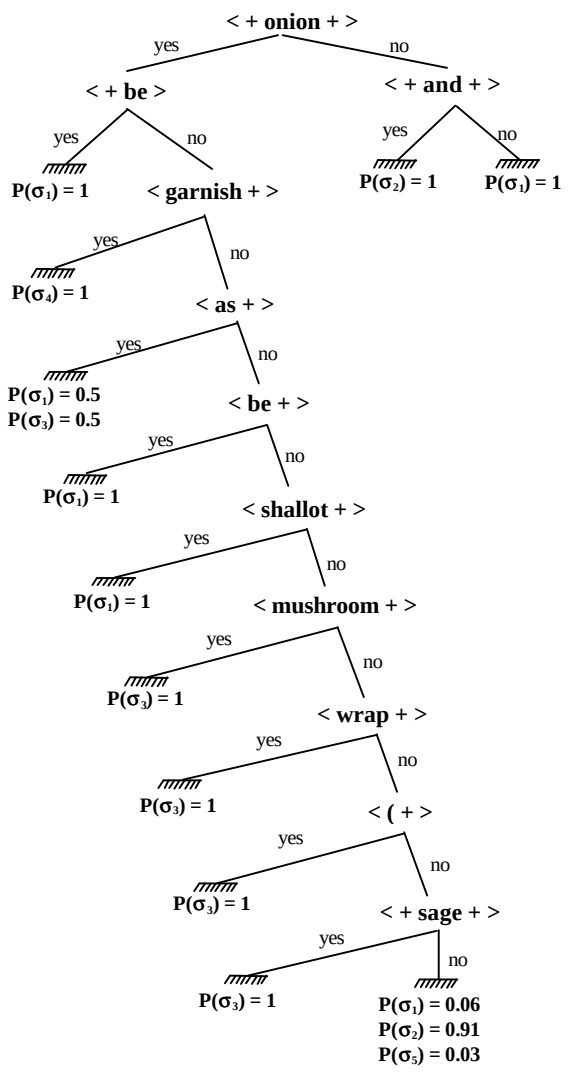
A SCT is a specialized classification tree that learns semantic rules from training data. At each node of the tree a question is put, and there are two branches that correspond with the answers "Yes" or "No". For instance, let us look at the SCT created for the word *onion*.

The symbols '<' and '>' mark the boundary of a pattern. '+' indicates a gap of at least one word. The following notations indicate the senses of onion:

σ_1 = 528344, σ_2 = 528347, σ_3 = 528347-m,
σ_4 = 528348, σ_5 = 528376

For example,
- if the context of *onion* follows the pattern < *garnish* + *onion* + >. That is to say *garnish* is the first word of the pattern, one or several words follow, then *onion*, then again one or several words
- *be* is not the last lemma

then the sense σ4 is assigned to *onion*.



4 A second method

4.1 Quantity of information

Different lemmas do not have the same importance for a disambiguation task. Very frequent words, which appear in almost all the examples related to Λ, do not give any information on the sense of Λ. A measure of the quantity of information $Q(\lambda, \Lambda)$ is calculated for each lemma λ appearing in the context of Λ. The higher this quantity is, the more confidant is the information given by λ. Let us define:

$$Q(\lambda, \Lambda) = -\log \frac{1 + N(\lambda, \Lambda)}{1 + N_t(\Lambda)} \quad (1)$$ where:

- $N(\lambda, \Lambda)$ is the number of semantic tags of Λ for which there is at least one example containing λ
- $N_t(\Lambda)$ is the number of possible semantic tags for Λ.

It can be seen that if for all the semantic tags of Λ, λ appears at least in one example[1], then $Q(\lambda, \Lambda) = 0$.

4.2 Similarity

A similarity between the lemma Λ in a particular context and a sense σ is now calculated. Let us call τ a sample of the TEST containing an occurrence of Λ. Since a lemma and its context define one (or several when the ambiguity cannot be solved) semantic tag(s), we can define a similarity between τ and σ:

$$S(\sigma, \tau) = \underset{\chi \in X(\Gamma)}{\text{Max}} \left(S(\tau, \chi) \right) \quad (2)$$

where:

- $X(\Lambda)$ is the set of examples associated with Λ
- χ is an example belonging to $X(\Lambda)$
- $S(\tau, \chi)$ is the similarity between the context τ and a context χ from the set of examples $X(\sigma)$.

$S(\tau, \chi)$ is calculated by a dynamic time wrapping algorithm allowing to search the optimal alignment between τ and χ.

Let $\chi = \{\lambda_i\}_{1 \le i \le I}$ and $\tau \quad \{\mu_i\}_{1 \le j \le J}$

One has $S(\tau, \chi) = s(I, J)$ where $s(i, j)$ is defined by:

$$\begin{cases} d(0,0) = 0 \\ d(i, j) = \text{Max} \begin{vmatrix} d(i-1, j-1) + V_{i,j} & \text{(a)} \\ d(i-1, j) - Q(\lambda_i, \Lambda)) & \text{(b)} \\ d(i, j-1) - Q(\mu_j, \Lambda) & \text{(c)} \end{vmatrix} \end{cases}$$

with $V_{i,j}$ defined by:

$$\begin{vmatrix} V_{i,j} = 2 \times Q(\lambda_i, \Lambda) & \text{if } \lambda_i = \mu_j \\ V_{i,j} = 6 \times Q(\lambda_i, \Lambda) & \text{if } \lambda_i = \mu_j = \Gamma = \text{treated lemma} \\ V_{i,j} = -\dfrac{\left(Q(\lambda_i, \Lambda) + Q(\mu_j, \Lambda)\right)}{2} & \text{if } \lambda_i \neq \mu_j \end{vmatrix}$$

[1] One calculates $(1 + N(\lambda, \Lambda))$ instead of $N(\lambda, \Lambda)$ because several lemmas, appearing in the test corpus, do not appear in the examples and we must be able to calculate $Q(\lambda, \Lambda)$ when λ is unknown. In that case, we have $Q(\lambda, \Lambda) = \dfrac{1}{1 + N_t(\Lambda)}$.

The case (a) corresponds to a substitution (λ_i in τ is replaced by μ_j in χ), or to an equality ($\lambda_i = \mu_j$). (b) is for an omission and (c) for an insertion.

4.3 Calculation of the most probable semantic tag

All the examples are aligned with the sample of the test corpus and their similarity with this test is calculated with the formula (2).

Then, we assign Λ the sense unit σ such as:

$$S(\sigma,\tau) = \underset{\sigma \in T(\Lambda)}{\text{Max}}\left(S(\sigma,\tau)\right) \quad (3)$$

where $T(\Lambda)$ is the set of semantic tags of the lemma Λ.

5 A third method: weighted quantity of information

Let us define a weighted quantity of information:

$$I(\lambda,\sigma) = Q(\lambda,\Lambda)\times\log(\varepsilon + N(\sigma))\times\frac{N(\lambda,\sigma)}{N(\sigma)} \quad (4)$$

where:

- σ is one of the semantic tag associated with Λ; it indicates a sense unit, that is to say, a sense associated with a lemma, not a synonym set.
- $N(\sigma)$ is the number of example for σ,
- $N(\lambda,\sigma)$ is the number of examples containing λ and associated with σ.
- $\frac{N(\lambda,\sigma)}{N(\sigma)}$ stands for the probability to see λ in the context of Λ, given σ.
- $\log(\varepsilon + N(\sigma))$ stands for the reliability of the previous measure. The more examples for σ, the more the previous probability is reliable. ε is added because one example is better than nothing.

Then, a similarity between the sense unit σ and the sample τ can be defined:

$$I(\tau,\sigma) = \sum_{\lambda\in\tau} I(\lambda,\sigma) \quad (5)$$

And for a lemma Λ, the most probable semantic tag $\bar{\sigma}$ in the context τ is such as:

$$\bar{\sigma} = \underset{\sigma\in T(\Lambda)}{\text{ArgMax}}\left(I(\tau,\sigma)\right) \quad (6)$$

6 Comparison of the different methods for SENSEVAL

Within the SENSEVAL evaluation, the three methods have been used for the semantic assignation task. Moreover, an average between the three values is then calculated. For each method, the tag having the best score is assigned to the lemma. If, for a given tag, the average is below 0.1, this tag is not considered. Therefore, in some cases, when all the possible senses have an average probability below 0.1, no tag is assigned to the lemma.

The following table gives the scores obtained by each of the four methods. The first column gives the lemma, the second one the score for a coarse grained semantic tagging (without a lot of precision), the third, a normal-grained tagging, the fourth, a fine-grained tagging (great precision) and the last column represent the average between the coarse, the middle and the fine-grained scores.

	coarse	**middle**	**fine**	**Average**
SCT	0.844	0.816	0.765	0.808
second	0.661	0.619	0.510	0.597
third	0.673	0.632	0.510	0.605
average	0.767	0.733	0.649	0.716

It is clear that the best method is the SCT. And even, the average is less efficient.

The submitted method to SENSEVAL is the "average" one. The following table gives its score in SENSEVAL for the 11 treated nouns.

class	**coarse**	**middle**	**fine**	**average**
trainable-nouns	0.767	0.733	0.649	0.716
accident-n	0.823	0.767	0.644	0.745
behaviour-n	0.877	0.877	0.777	0.844
bet-n	0.557	0.531	0.485	0.524
excess-n	0.723	0.663	0.532	0.639
giant-n	0.695	0.557	0.503	0.585
knee-n	0.800	0.768	0.720	0.763
onion-n	0.762	0.762	0.762	0.762
promise-n	0.786	0.748	0.606	0.713
sack-n	0.732	0.732	0.693	0.719
scrap-n	0.698	0.661	0.489	0.616
shirt-n	0.841	0.806	0.727	0.791

It can be seen that for *bet* and *scrap*, the scores are lower than for the other words. And for these words, the number of senses is very high (19 and 17 respectively), whereas they have not a lot of training examples (146 and 59). One could conclude that there is a direct consequence from the ratio between the number of examples and the number of senses. But the example of *shirt* is an illustration of a word having a lot of senses (16), only few example (181, less than 12 by sense) and a very good score.

7 The ROMANSEVAL evaluation of French WSD

Since no training corpus are available for the French WSD evaluation, it has been very difficult for us to apply one of the methods we have implemented. Therefore, the work we have done is only a study, and experiments have been performed on nouns and verbs.

In order to have training data, we have used the test corpora themselves. Considering the words to be tagged, the set of examples (context of these words) has been automatically extracted. Then, we have assigned manually one (or several) tag(s) to the words given their short context when possible. This work was done for the French corpus and the English counterpart. Of course, we never have looked at the entire sentence, except for some English sentences, when there is a problem to find the word used to translate the French one. Moreover, examples of use have been extracted from the definitions. The confidence of an example depends both on the number of times it appears and an arbitrary score given by a human judge.

For instance, if we consider the noun *chef*, the following table gives some pattern matching examples with the source of the example, the number of occurrences in both corpus and dictionary and the sense of *chef* represented.

examples	origin	nb. occ.	sense
chef de famille	dictionary	1	1a
chef d'entreprise	dictionary	12	1a
chef de bataillon	dictionary	1	1b
chef de cabinet	dictionary	1	1b
chef d'état	corpus	18	1b
chef de projet	corpus	1	1b

When there are more than one possibility for an example, it is duplicated with the same score. For instance, the sense of *chef* in "*chef religieux*" could be ***1a*** or ***1b***. Some lacks in the definitions have caused problems for the manual assignation of sense. For instance, the very frequent *chef-d'œuvre* is not represented.

8 Conclusion

The different techniques described in this article have given some interesting results. It is difficult to compare them with the HMM model described in [Loupy et al., 98b]: there is a great difference between the training corpora. Moreover, for the HMM system, we assign tags not only to nouns, but also to verbs, adverbs and adjectives.

Although the best method seems to be the SCT, only 11 nouns were used for the evaluation. Further experiments are necessary.

9 References

[Computational Linguistics, 1998]: *Special Issue on Word Disambiguation*; Computational Linguistics, Vol. 24, No. 1; March 1998.

[Kilgarriff, 1998]: A. Kilgarriff; *SENSEVAL: An Exercise in Evaluating Word Sense Disambiguation Programs*; in Proceedings of the First International Conference on Language Resources & Evaluation; pp. 1255-1258; Granada, Spain; 28-30 May 1998.

[Kuhn & De Mori, 1995]: R. Kuhn & R. De Mori; *The Application of Semantic Classification Trees to Natural Language Understanding*; IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 17, No. 5; May 1995.

[Ide & Véronis, 1998]: N. Ide & J. Véronis; *Introduction to the Special Issue on WSD: The State of the Art*; Special Issue on Word Disambiguation, Computational Linguistics, Vol. 24, No. 1; March 1998.

[Loupy et al., 98a] Claude de Loupy, Pierre-François Marteau & Marc El-Bèze; *Navigating in Unstructured Textual Knowledge Bases*; in Proceedings of Nîmes'98 - La Lettre de l'IA; pp. 82-85; May 1998.

[Loupy et al., 98b] Claude de Loupy, Marc El-Bèze & Pierre-François Marteau; "*Word Sense Disambiguation using HMM Tagger*"; in Proceedings of the First International Conference on Language Resources & Evaluation; pp. 1255-1258; Granada, Spain; 28-30 May 1998.

[Miller et al., 1993a] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross & K. Miller; "*Introduction to WordNet: An On-Line Lexical Database*"; http://www.cosgi.princeton.edu/~wn; August 1993.

[Miller et al., 1993b]: Miller G., Leacock C., Randee T., Bunker R. (1993); *A Semantic Concordance*; In

Procedings of the 3rd DARPA Workshop on Human Language Technology (pp 303-308); Plainsboro, New Jersey.

[Spriet & El-Bèze, 1997]: T. Spriet & M. El-Bèze; *Introduction of Rules into a Stochastic Approach for Language Modelling*; in Computational Models for Speech Pattern Processing, NATO ASI Series F, editor K.M. Ponting; 1997.

[Yarowsky, 1993]: D. Yarowsky; *One Sense per Collection*; in Proceedings of ARPA Human Technology Workshop; Princeton, NJ; 1993.