SENSEVAL-3: Third International Workshop on the Evaluation of Systems
for the Semantic Analysis of Text, Barcelona, Spain, July 2004
Association for Computational Linguistics

Using a Word Sense Disambiguation system for translation disambiguation: the LIA-LIDILEM team experiment

Grégoire MOREAU DE MONTCHEUIL
Marc EL-BEZE
Laboratoire informatique d'Avignon, Université d'Avignon
{moreau, elbeze}@lia.univ-avignon.fr

Boxing CHEN, Olivier KRAIF
LIDILEM
Université Stendhal Grenoble 3
{Chen, Kraif}@u-grenoble3.fr

Abstract

This paper presents an original WSD method, based on a mixture of three algorithms working on the local context of target units. The results on the task *Multilingual lexical sample* (*MLS*) of Senseval 3 were 60.3% of precision and recall for the T sub-task, and 64.1% for TS. We attempted to improve the method by constituting synonym-like classes from an English-French aligned corpus, but without any gain in the results.

1 Introduction

The CARMEL Project aims at gathering literary texts with translations in 4 languages (French, English, Spanish and Italian), and implementing Word Sense Disambiguation and Thematic Identification methods, taking advantage of the multilingual context of each unit. In this preliminary study, conducted in the framework of the MLS task of Senseval 3, we try to address two related issues. On the one hand, we would like to determine the results of a state-of-the-art supervised WSD method applied to a translation-tagged corpus, compared to a manually sense-tagged corpus. On the other hand, we try to check whether the results of the translation-disambiguation task can be improved using a complementary resource, namely a bilingual parallel corpus, that allows us to create synonym-like classes, in order to make training contexts more general. Section 2 is dedicated to the WSD algorithm description. Section 3 deals with the aligning techniques that have been implemented to constitute a word-level aligned corpus. A simple method for synonym-like class extraction is described. Section 4 gives the results and their discussion.

2 WSD Tool Description

To assign senses or translation tags, we use weighted mixtures of three algorithms: decision trees, a probabilistic method and k-nearest-neighbours. We first describe individually each algorithm, and then our mixing methods.

2.1 Semantic Classification Tree

De Mori and Kuhn (1995) have proposed to apply a Decision Tree approach to the sense disambiguation problem. The tree grown for each target word is called Semantic Classification Tree (SCT). In a first step, we construct the decision tree with the training corpus. To each node is associated a question which leads to splitting the population of examples into two sub-populations. The choice of the question is done by maximization of the entropy gain using Gini criterion:

$$E(P)=1-\sum_{s\in Senses} p_s^2$$

where $(p_s)_{s\in Senses}$ is the distribution of probabilities of senses in the population P.
For a population P split by question q into a partition of two sub-populations P_y and P_n, the entropy gain is: $Gain(q)=E(P)-\left(\frac{|P_y|}{|P|}E(P_y)+\frac{|P_n|}{|P|}E(P_n)\right)$

For our WSD system, node-questions are queries of presence (or absence) of a single word in the example context, independently of its position. If several words maximize the entropy gain, a preference is given to the longest one, assuming that it will bear more specification. In a second step, the system uses the decision tree to find a path from the root to a leaf: $(n_j)_{0\le j\le m}$, where n_0 is the root, n_m is the leaf, and each node n_j has the probabilities of senses $(p_{s,j})_s$. Since the questions are binary, unseen events may have very undesirable effect on the path selection. Therefore, the path is used to smooth the score calculation, as suggested by Breiman *et al.* (1984). So, when the path contains three or more nodes, the final score for each sense s is: $Sc_{Tree}(s)=\sum_{i=0}^{2} w_i p_{s,m-i}+w_r p_{s,0}$

and when the path length is two:

$$Sc_{Tree}(s) = w'_0 p_{s,1} + w'_1 p_{s,0} .$$

Note that weights w_i and w'_i are positive, sum to 1, and have been determined empirically.

2.2 Probabilistic Approach

As proposed by Bahl *et al.* (1988) for performing a fast match in the framework of an automatic speech recognition system, the problem of word sense disambiguation can be seen as a polling function. Each term found in the context of the target word $C=(w_i)_{-g\le i\le +d}$, will vote for each possible sense of the target word. We assume that these terms have Poisson distributions

$P(w_i / s) = \frac{e^{\lambda_{w_i,s}} \lambda_{w_i,s}^x}{x!}$, where x is number of occurrences of w_i in the current context.

After deriving some equations under the assumption of the independence hypothesis, it is easy to express the score of each sense s as

$$Sc_{Pois}(s) \approx p(s).\prod_{i=-g}^{+d} \lambda_{w_i,s}$$

$p(s)$ is estimated as the number of examples provided for this sense divided by the total number of examples. For each word w (found in the training contexts), $\lambda_{w,s}$ is estimated as the average of occurrences of this word in the examples related to sense s.

Note that if a term w does not appear in the training instances for s, $\lambda_{w,s}$ has a default value, smaller of all other λ, but not zero.

2.3 K-Nearest Neighbours

The KNN algorithm is a dynamic disambiguation method. Into the set of learning examples, the system selects a subset of the k most similar with the ambiguous example. If $C=(w_i)_{-g\le i\le +d}$ and $C'=(w'_i)_{-g'\le i\le +d'}$ are two contexts of an ambiguous lemma (where w_0 and w'_0 are the ambiguous word), the similarity between C and C' is the square of the number of identical words at the same relative position:

$Simil(C,C') = \left(\sum_{i=-G}^{+D} (w_i \equiv w'_i) \right)^2$, where $G=min(g,g')$ and $D=min(d,d')$. The motivation for adding a square to the simple similarity measure was to enlarge the domain variation of the scores obtained by the different senses, to make it comparable to the one estimated through the SCT scheme.

To cope with the problem of similarity score tie (more than k examples have a similarity greater or equal to this of the k^{th} most similar example), the constraint ($k=4$ in our case) is relaxed so that the vote relies on more than k neighbours.

In all cases, each neighbour votes proportionally with its similarity. After normalization, the score of each sense is:

$$Sc_{KNN}(s) = \frac{\sum_{C\in KNN \cap Train(s)} Simil(C,C_q)}{\sum_{C\in KNN} Simil(C,C_q)}$$

where C_q is the ambiguous context, KNN the set of neighbours, and $Train(s)$ the set of training examples for the sense s.

2.4 Two different mixtures

To increase the performances, we merge the individual result of the three algorithms. We have first used a natural mixing method, which calculates the weighted sum of the individual scores:

$$Sc_{Mix0}(s) = W_{Tree}.Sc_{Tree}(s) + W_{Pois}.Sc_{Pois}(s) + W_{KNN}.Sc_{KNN}(s)$$

We don't actually have any heuristic for determining the weights and therefore, we consider that each algorithm has the same weight (i.e. W= 1/3). This basic mixture will be denoted mix-0.

Undesirable behaviour of this kind of mixture has been observed, as, for example, the SCT dominates the raw mixture. Despite the smoothing, the top choice of the SCT is given an exaggerated weight. So, we have developed another merging strategy mix-1, which takes into account the rank of a sense in the different results of an algorithm:

$$Rk_{Algo}(s) = \sum_{s'\in Senses} \left(Sc_{Algo}(s') \ge Sc_{Algo}(s) \right).$$

Then, $Sc_{Mix1}(s) = \frac{W_{Tree}}{Rk_{Tree}(s)} + \frac{W_{Pois}}{Rk_{Pois}(s)} + \frac{W_{KNN}}{Rk_{KNN}(s)}$.

Last, we filter the results to keep only a few senses (generally 1 or 2): a sense is conserved if its score is greater of *95%* of the best score.

3 Use of a bilingual parallel corpus

Many methods have been proposed to improve WSD using multilingual material. Ide *et al.* (2001) showed that a multilingual corpus allows to discriminate senses as well as human annotators, and Diab & Resnik (2002) have imple-

mented an unsupervised WSD method using an artificial machine translated multilingual corpus.

We decided to test the following hypotheses: 1/ using an English-French aligned corpus, it is possible to identify a kind of synonymic relationship between English units. 2/ from this relationship, we may create more general classes among English words, and train our WSD models on these classes, in order to give more consistency to the semantic content of the training contexts: for instance, if the word *huge* does not appear in the training contexts, it cannot bring any information for the WSD of a new example; but if *large* does occur in the training, and if *huge* is known as synonym of *large*, the occurrence of *huge* may be taken into account and improve WSD.

As in Diab & Resnik (2002), hypothesis 1 states that if two units e_1 and e_2 are often translated by the same French unit *f*, they may share some common sense. *f* can be polysemous and e_1 and e_2 may correspond to various senses, but these senses must be somehow related. To reinforce this hypothesis, and to adapt the classes to the task, specific word classes are made from each target-word sentence set.

The implementation of hypothesis 1 may face two sources of noise: wrong word-to-word alignments and homographs. We tried to filter out this noise using two criteria:

- Word pairs frequencies: every aligned word-pair that occurs less than 5 times through the aligned corpus is discarded.
- Class stability: The classes are constructed iteratively: from an English word *e*, it is possible to get the set $F^1(e)$ of every associated words in French. From this set, a new extraction yields the set $E^1(e)$ of the English corresponding words. Stability is reached when $E^n=E^{n+1}$. The classes finally constitute a partition of the English word set. For instance, *huge* and *large* fall in the same class after 3 iterations: E1(*huge*)={*huge*, *important*,*vast*}, E2(*huge*)={*huge*,*important*,*vast*, *large*}=E3(*huge*) and E1(*large*)={*large*, *vast*} E2(*large*)={*huge*, *important*, *vast*, *large*}= E3(*large*). Noisy word alignments generally result in the cascading fusion of non related word sets: they yield large classes that become stable only after a lot of iterations. Thus, we decided to discard every non stable class after three iterations, each word being considered as a singleton.

For sentence alignment, we have used a combination of clues (cognates, sentence length) with an algorithm that yielded results similar to the best system of the Arcade Campaign (Langlais & Véronis, 2000). For lexical alignment, we implemented an original method based on probabilistic models for co-occurrence, word position and POS correspondence. Evaluated on a manually aligned corpus (149 aligned sentences extracted from Flaubert's *Madame Bovary*), this method yielded precision and recall over 90% for the content words.

Then, we applied these algorithms to the literary multilingual corpus that we are developing for the Carmel Project (travel stories from the late 19th century). The corpus has previously been tokenized and POS tagged. It included around 850,000 words in each language, and 359,123 content word pairs have been extracted.

4 Experiment

4.1 Default Treatment

In order to improve WSD, it is important for different reasons (mainly a better coverage) to tag the words at a POS and morphological level. Thanks to the POS tags, we can eliminate determinants and punctuations in the target word context. Moreover, some groups of words are substituted by a pseudo-lemma for instance: numbers by CD, all pronouns by PRONOUN, days of week by DAY, months by MONTH.

Afterwards, the context is reduced to a short window[2]: 3 words before and 3 words after the ambiguous word, within the same sentence.

Finally, the context words are lemmatized (or class-reduced), except the ambiguous word that keeps its form (including its case). Our assumption is that it may have an impact on the sense determination (for example *bank* and *Bank*)

In the case of the *Translation and Sense* test, we append the English sense's identifier to the ambiguous word. In this way, the WSD program and the context windowing don't need any change. But this choice has the drawback of linking together the form and the English sense. See the two last lines of the example given in Table 1.

[12] The SCT is subject to capture noise when the context is too large, as shown in Crestan and El-Bèze (2001)

Initial context	**Even before** the **huge new projects began**, the **Strip's recent expansion squeezed smaller competitors.**
Filtered context	Even before **huge new projects began Strip 's recent** expansion squeezed smaller competitors
FL context	huge new project began strip 's recent
FL context with sense	huge new project **began_19** strip 's recent
FC context with sense	**important** new project began_19 strip 's recent

Table 1: examples from the *begin* contexts

4.2 Results

Our results for MLS task are displayed on table 2 (only mix-0 results have been submitted).
One can observe that using the sense-tag as additional information, the results are significantly improved (we only obtain 0.593 with mix-0, and 0.603 with mix-1, when removing sense's identifier in sub-task TS).

	Sub-task T		Sub-task TS	
P and R	Using lemmas	Using classes	Using lemmas	Using classes
mix-0	0.603	0.603	0.641	0.641
mix-1	0.607	0.607	0.645	0.645

Table 2: results for MLS task

We note that the global results without and with classes are the same, and this is quite disappointing. Indeed, for the T sub-task, the results were a bit higher for certain words and lower for others, and the exact identity of the global precision is fortuitous. For the TS sub-task, the results were identical. The gain that may be brought by semantically more consistent contexts (the classes) may be compensated by the noise inherited from the aligning and clustering methods. In addition, the lack of improvement may be due to the discrepancy between our English-French corpus and the task: classes built from an English-French corpus may not suit the English-Hindi examples; moreover, classes were extracted from a 19th Century literary corpus, quite heterogeneous with the Training and the Test corpora.

5 Conclusion

The results of our WSD method are correct, while a bit inferior to the expected results for a traditional WSD task. Translation disambiguation may require specific methods, involving more contrastive data.
The use of a bilingual corpus to improve our WSD method has not been very conclusive. The substitution of context words with more general synonym-like word classes may be worth further experiments, using an appropriate bilingual material.

6 Acknowledgements

Thanks to RIAM which funds the Carmel Project, and to our partners, ACCE and SINEQUA.

References

E. Crestan & M. El-Bèze. 2001. Improving Supervised WSD by Including Rough Semantic Features in a Multi-Level View of the Context. *Sempro* 2001.

M. Diab and P. Resnik. 2002. An Unsupervised Method for Word Sense Tagging using Parallel Corpora, in *Proc. of ACL-02*, Philadelphia.

C. de Loupy, M. El-Bèze, P.-F. Marteau. 1998. WSD based on three short context methods, *Senseval Workshop*, Herstmontceux.

N. Ide, T. Erjavec and D. Tufis. 2001. Automatic Sense Tagging Using Parallel Corpora. In *Proc. of the Sixth Natural Language Processing Pacific Rim Symposium*, Tokyo, 83-9.

J. Véronis and P. Langlais. 2000. Evaluation of parallel text alignment systems – The ARCADE project. In *Parallel Text Processing*, J. Véronis, ed., Dordrecht, Nederlands: Kluwer Academic Publishers, pp. 49-68.

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. 1984. *Classification and Regression Trees*. Wadsworth Inc.

L. Bahl, R. Bakis, P. de Souza and R. Mercer. 1988. Obtaining candidate words by polling in a large vocabulary speech recognition system. In *Proc. ICASP, vol. 1*, pp. 489-92.

R. De Mori, R. Kuhn. 1995. The Application of Semantic Classification Trees to Natural Language Understanding, *IEEE Transactions on PAMI*, vol. 17, no. 5, pp. 449-460.