Error Driven Word Sense Disambiguation

Luca Dini and **Vittorio Di Tomaso**
CELI
{dini,ditomaso}@celi.sns.it

Frédérique Segond
Xerox Research Centre Europe
Frederique.Segond@xrce.xerox.com

Abstract

In this paper we describe a method for performing word sense disambiguation (WSD). The method relies on unsupervised learning and exploits functional relations among words as produced by a shallow parser. By exploiting an error driven rule learning algorithm (Brill 1997), the system is able to produce *rules* for WSD, which can be optionally edited by humans in order to increase the performance of the system.

1 Introduction

Although automatic word sense disambiguation (WSD) remains a much more difficult task than part of speech (POS) disambiguation, resources and automatic systems are starting to appear. Some of these systems are even mature enough to be evaluated. This paper presents an overview of a system for English WSD which will be evaluated in the context of the SENSEVAL project[1]. We report on performing automatic WSD using a specially-adapted version of Brill's error driven unsupervised learning program (Brill, 1997), originally developed for POS tagging. In our experiment, like in Resnik (1997), we used both functional and semantic information in order to improve the learning capabilities of the system. Indeed, by having access to a syntactic and functional sketch of sentences, and by being able to stipulate which relations are important for sentence meaning, we overcame some of the traditional problems found in continuous bigram models, such as the occurrence of interpolated clauses and passive constructions.

Consider, for example, temporal expressions like *Tuesday* in *The stock market Tuesday staged a technical recovery.* Such expressions are quite frequent in newspaper text, often appearing near verbs. Without any functional information, the semantic rules produced by the algorithm will stipulate a strong semantic relation between the semantic class of words like *Tuesday* and the semantic class of verbs like *stage.* On the contrary, if we use information from a shallow parser, we know that *Tuesday* is an adverbial expression, probably part of the verb phrase, and that the really important relation to learn is the one between the subject and the verb.

In the following sections we describe (i) the resources we used (Penn Tree Bank, 45 upper level WordNet tags); (ii) the experiment we ran using rule induction techniques on functional relations (functional relation extraction, tag merging, corpus preparation and learning); (iii) the evaluation we performed on the semantically hand-tagged part of the Brown corpus and, finally, we sketch out the general architecture we are in the process of implementing.

2 The Resources

We decided to take advantage of the syntactic structures already contained in the Penn Tree Bank (PTB) (Mitchell et al., 1995) in order to build a large set of functional relation pairs (much as in Resnik (1997)). These relations are then used to learn how to perform semantic disambiguation. To distinguish word meanings we use the top 45 semantic tags included in WordNet (Miller, 1990). The non-supervised Brill algorithm is used to learn and then to apply semantic disambiguation rules. The semantically hand-tagged Brown Corpus is used to evaluate the performance of automatically acquired rules.

2.1 Obtaining Functional Structures.

We consider as crucial for semantic disambiguation the following functional relations: SUBJ/VERB, VERB/OBJ, VERB/PREP/PREP-

[1] http://www.itri.bton.ac.uk/events/senseval

OBJ, NOUN/PREP/PREP-OBJ.

In order to extract them, we parsed the PTB structures using *Zebu* (Laubusch, 1994), a LARLR(1) parser implemented in LISP. The parser scans the trees, collecting information about relevant functional relations and writing them out in an explicit format. For instance, the fragment *you do something to the economy*, after some intermediate steps which are described in Dini et al. (1998a) and Dini et al. (1998b), is transformed into:

```
HASOBJ do something
HASSBJ do you
PREPMOD do TO economy
```

2.2 Adding Lexical Semantics.

The WordNet team has developed a general semantic tagging scheme where every set of synonymous senses, *synsets*, is tagged with one of 45 tags as in WordNet version 1.5. We use these tags to label all the content words contained in extracted functional relations. We associate each word with all its possible senses ordered in a canonical way. The semantically tagged version of the sample sentence given above is:

HASOBJ do/stative_social_motion_creation_body something/top
HASSBJ do/stative_social_motion_creation_body you/person
PREPMOD do/stative_social_motion_creation_body TO
economy/group_cognition_attribute_act

2.3 Preparing the input.

As a result of adding lexical semantics we get a triple <functional relation, word$_i$/tagset$_i$, word$_j$/tagset$_j$>, but in its current formulation, the unsupervised learning algorithm is only able to learn relations holding among bigrams. Thus, it can learn either relations between a functional relation name (e.g. "HASOBJ") and a tagset or between tagsets, without considering the relation between them. In both cases we report a loss of information which is fatal for the learning of proper rules for semantic disambiguation. There is an intuitive solution to this problem: most of the relations we are interested in are diadic in nature. For example, adjectival modification is a relation holding between two heads (MOD(h1,h2)). Also relations concerning verbal arguments can be split, in a neo-davidsonian perspective, into more atomic relations such as "SUBJ(h1,h2)" "OBJ(h1,h2)". These relations can be translated into a "bigram format" by assuming that the relation itself is incorporated among the properties of the involved words (e.g. **w1/IS-OBJ w2/IS-HEAD**). Learnable properties of words are standardly expressed through *tags*. Thus, we can merge functional and semantic tags into a single tag (e.g. **w1/IS-OBJ w2/IS-HEAD + w1/2_3 w2/4 ⇒ w1/IS-OBJ2_IS-OBJ3 w2/IS-HEAD4**). The learner acquires constraints which relate functional *and* semantic information, as planned in this experiment. We obtain the following format where every line of the input text represents what we label an FS-pair (*Functional Semantic pair*):

$$\text{do}/\frac{HASOBJ}{42\_41\_38\_36\_29} \quad \text{something}/\frac{HASOBJ^{-1}}{3}$$

$$\text{do}/\frac{HASSBJ}{42\_41\_38\_36\_29} \quad \text{you}/\frac{HASSBJ^{-1}}{18}$$

where relations labelled with $^{-1}$ are just inverse relations (e.g. HAS-SUBJ^{-1} ≡ IS-SUBJ-OF). Functional relation involving modification through prepositional phrases is ternary as it involves the preposition, the governing head and the governed head. Crucially, however, only substantive heads receive semantic tags, which allows us to condense the preposition form in the FS tags as well. The representation of the modification structure of the phrase *do to the economy* becomes:

$$\text{do}/\frac{MOD\text{-}TO}{42\_41\_38\_36\_29} \quad \text{economy}/\frac{MOD\text{-}TO^{-1}}{14\_9\_7\_4}$$

3 Unsupervised Learning for WSD

Sufficiently large texts should contain good cues to learn rules for WSD in terms of *selectional preferences*.[2] The crucial assumption in using functional relations for WSD is that, when compositionality holds, selectional preferences can be checked through an intersection operation between the semantic features of the syntactically related lexical items. By looking at functional relations that contain at least one non-ambiguously tagged word, we can learn evidence for disambiguating ambiguous words appearing in the same context. So, if we know that in the sentence *John went to Milan* the word *Milan* is

[2] By selectional preferences we mean both the selection of semantic features of a dependent given a certain head and its inverse (i.e. selection of a head's semantic features by a dependent constituent).

unambiguously tagged as `place`, we learn that in a structure *GO to X*, where *GO* is a verb of the same semantic class as the word *go* and *X* is a word containing `place` among its possible senses, then *X* is disambiguated as `place`.

The Brill algorithm[3] is based on *rule patterns* which describe rules that can be learned, as well as on a lexicon where words are associated with ambiguity classes. The learning algorithm is recursively applied to an ambiguously tagged corpus, producing a set of rules. The set of learnable rules includes the rules for which there is corpus evidence in terms of unambiguous configurations. In other words, the learning algorithm extensively relies on bigrams where one of the words is unambiguously tagged. The preferred rules, the ones with the highest score, are those that best minimize the entropy of the untagged corpus. For instance, a rule which resolves ambiguity for 1000 occurences of a given ambiguity class is preferred to one which resolves the same ambiguity only 100 times.

Consider the following rule pattern: *Change tagSet* (X_1, X_2 ....X_n) *into tag* X_i *if the left context is associated with the tagSet* (Y_1, Y_2 ... Y_m). This pattern generates rules such as:[4]

```
bi18_bi4 bi18 LEFT b42_b32 1209.64
```

which is paraphrased as: *If a noun is ambiguous between* `person` *and* `act` *and it appears as the subject of a verb which is ambiguous between* `stative` *and* `communication`, *then disambiguate it as* `person`. This instantiation relies on the fact that the untagged corpus contains a significant number of cases where a noun unambiguously tagged as `person` appears as subject of a verb ambiguous between `stative` and `communication`. The rule is then applied to the corpus in order to further reduce its ambiguity, and the new corpus is passed again as an input to the learner, and the next most preferred rule is learned.

Three different scoring methods have been used[5] as criteria to select the best rule. They are referred to in the program documentation, and in Dini et al. (1998a), as "paper", "original" and "goodlog". Here we will describe only "original" and "goodlog", because "paper" differs from "original" only for some implementation details.

In the method called "original", at every iteration step the best scored disambiguation rule is learned, and the score of a rule is computed, according to Brill, in the following way: assume that *Change the tag of a word from* ξ *to* Y *in context* C is a rule ($Y \in \xi$). Call R the tag Z which maximizes the following function (where Z ranges over all the tags in ξ except Y, $freq(Y)$ is the number of occurences of words unambiguously tagged with Y, $freq(Z)$ is the number of occurences of words unambiguously tagged with Z, and $incontext(Z, C)$ is the number of times a word unambiguously tagged with Z occurs in context C):

$R = argmax_Z \frac{freq(Y)*incontext(Z,C)}{freq(Z)}$

The score assigned to the rule would then be:

$S = incontext(Y, C) - \frac{freq(Y)*incontext(R,C)}{freq(R)}$

In short, a good transformation from ξ to Y is one for which alternative tags in ξ have either very low frequency in the corpus or they seldom appear in context C. At every iteration cycle, the algorithm simply computes the best scoring transformation.

The method "goodlog" uses a probabilistic measure which minimizes the effects of tag frequenc, adopting this is the formula for giving a score to the rule that selects the best tag Y in a context C (Y and Z belong to the ambiguous tagset):

$S = argmax_Y(Z)\text{abs}(\log(\frac{incontext(Y,C)}{freq(Y)} * \frac{freq(Z)}{incontext(Z,C)}))$

The differences in results between the different scoring methods are reported and commented on in section 4 in table 1.

4 Evaluation

For the evaluation we used as test corpus the subset of the Brown corpus manually tagged with the 45 top-level WordNet tags. We started with the Penn Tree Bank representation and went through all the necessary steps to build FS-pairs

[3]For the sake of clarity, we just present here the general lines of Brill's algorithm. For a detailed version of the algorithm see Brill's original paper (Brill, 1997).

[4]Letters are abbreviation for functional relation and numbers are abbreviations for semantic tags.

[5]The search space of the algorithm is parametrised setting two different thresholds governing the possibility for a tag or a word to appear in a rule: i) the minimal frequency of a tag; ii) the minimal frequency of a word in the corpus. We set the first parameter to 400 (that is, we asked the learner to consider only the 400 most frequent TagSets) and we ignored the second one (that is we asked the learner to consider all words in the corpus).

used by the applier. These FS pairs were then labelled according to the manual codification and used as a standard for evaluation. We also produced, from the same source, a randomly tagged corpus for measuring the improvements of our system with respect to random choice.

The results of comparing the randomly tagged corpus and the corpus tagged by our system using the methods "original" and "goodlog" are shown in table 1. As usual, Precision is

	Precision	Recall	F-measure	Adjusted
Random	0.45	0.44	0.44	0.28
500 Goodlog	0.97	0.25	0.40	0.91
500 Original	0.78	0.30	0.44	0.50

Table 1: Precision and recall figures

the number of correctly tagged words divided by the total number of tagged words; Recall is the number of correctly tagged words divided by the number of words in the test corpus (about 40000). F-measure is (2*Precision*Recall)/(Precison+Recall). The column labelled "Adjusted" reports the Precision taking into account non-ambiguous words. The adjusted precision is computed in the following way: (Correct - unambiguous words) / ((Correct + Uncorrect) - unambiguous words). On an absolute basis, our results improve on those of Resnik (1997), who used an information-theory model of *selectional strength preference* rather than an error-driven learning algorithm. Indeed, if we compare the "Adjusted" measure we obtained with a set of about 500 rules (50% precision), with the average reported by Resnik (1997) (41% precision), we obtain an advantage of 10 points, which, for a task such as WSD, is noteworthy. For comparison with other experiments, refer to Resnik (1997).

It is interesting to compare the figures provided by "goodlog" and "original". Since "goodlog" smooths the influence of absolute tag frequency, the learned rules achieve much higher precision, even though they are less efficient in terms of the number of words they can disambiguate. This is due to the fact that the most frequent words also tend to be the most ambiguous ones, thus the ones for which the task of WSD is most difficult (cf. Dini et al. (1998a)).

5 Towards SENSEVAL

As mentioned above, the present system will be adopted in the context of the SENSEVAL project, where we will adopt the Xerox Incremental Finite State Parser, which is completely based on finite state technology. Thus, in the present pilot experiment, we are only interested in relations which could reasonably be captured by a shallow parser, and complex informative relations present in the Penn Tree Bank are simply disregarded during the parsing step described in section 2.1. Also, structures which are traditionally difficult to parse through Finite State Automata, such as incidental and parenthetic clauses or coordinate structures, are discarded from the learning corpus. This might have caused a slight decrease in the performance of the system.

Some additional decrease might have been caused by noise introduced by incorrect assignment of senses in context during the learning phase (see Schuetze et al. (1995)). In particular, the system has to face the problem of sense assignment to named entities such as person or industry names. Since we didn't use any text preprocessor, we simply made the assumption that any word having no semantic tag in WordNet, and which is not a pronoun, is assigned the label **human**. This assumption is certainly questionable and we adopted it only as a working hypothesis. In the following rounds of this experiment we will plug in a module for named entity recognition in order to improve the performance of the system.

Another issue that will be tackled in the SENSEVAL project concerns word independence. In this experiment we duplicated lexical heads when they were in a functional relation with different items. This permitted an easy adaptation to the input specification of the Brill learner, but it has drawbacks both in the learning and the application phase. During the learning phase the inability to capture the identity of the same lexical head subtracts evidence for the learning of new rules. For instance, assume that at an iteration cycle n the algorithm has learned that verbal information is enough to disambiguate the word *cat* as **animal** in *the wild cat mewed.* Since the FS-pairs cat/mew and wild/cat are autonomous, at cycle $n+1$ the learner will have no evidence to learn that the adjective *wild* tends to associate

with nouns of type **animal**. On the contrary, *cat*, as appearing in *wild cat*, will still be ambiguous.

The consequences of assuming independence of lexical heads are even worse in the rule application phase. First, certain words are disambiguated only in some of the instances in which they appear, thus producing a decrease in terms of recall. Second, there might be a case where the same word is tagged differently according to the relations into which it enters, thus causing a decrease in terms of precision. Both problems will be overcome by the new Java-based versions of the Brill learner and applier which have been developed at CELI.

When considering the particular WSD task, it is evident that the information conveyed by adjectives and pre-nominal modifiers is at least as important as that conveyed by verbs, and it is statistically more prominent. In the corpus obtained from parsing the PTB, approximately $\frac{2}{7}$ of FS-pairs are represented by pre-nominal modification (roughly analogous to the subject-verb FS-pairs and more frequent than the object-verb pairs, which amount to $\frac{1}{7}$ of the whole corpus). But adjectives receive very poor lexical-semantic information from WordNet. This forced us to exclude them both from the training and test corpora. This situation will again improve in the SENSEVAL experiment with the adoption of a different semantic lexicon.

6 Conclusion

We presented a WSD system with reasonable results as well as suggestions for improving it. We will implement these improvements in the context of the SENSEVAL experiment and we plan to extend the system to other languages, with special attention to French and Italian.[6] Indeed, the availability of lexical resources providing a word sense classification with roughly the same granularity of the 45 top classes of Wordnet makes our method applicable also to languages for which no sense tagged corpora has been produced. In the long run, these extensions will lead, we hope, to better systems for foreign language understanding and machine translation.

Acknowledgements We are grateful to Ken Beesley, Andrea Bolioli, Gregory Grefenstette, David Hull, Hinrich Schuetze and Annie Zaenen for their comments and discussion on earlier versions of this paper. Our gratitude also goes to Vincent Nainemoutou and Herve Poirier for providing us with technical support. Any remaining errors are our own fault.

[6]The system will be used in the MIETTA project (LE4-8343) for enhancing the performance of the information extraction and information retrieval module.

References

E. Brill and P. Resnik. 1994. A rule-based approach to prepositional phrase attachment disambiguation. In *Proceedings of COLING.*

E. Brill. 1997. Unsupervised learning of disambiguation rules for part of speech tagging. In *Natural Language Processing Using Very Large Corpora.* Kluwer Academic Press.

Dini, L., V. Di Tomaso, F. Segond 1998. Error Driven Unsupervised Semantic Disambiguation. In *Proceedings of TANLPS ECML-98.* Chemnitz, Germany.

Dini, L., V. Di Tomaso, F. Segond 1998. Word Sense Disambiguation with Functional Relation. In *Proceedings of LREC-98.* Granada, Spain.

J. Laubusch. 1994. Zebu: A tool for specifying reversible LARL(1) parsers.

G. Miller. 1990. Wordnet: An on-line lexical database. *Int. Journal of Lexicography.*

M. Mitchell, B. Santorini, and M.A. Marcinkiewicz. 1995. Building a large annotated corpus of English : the Penn Treebank. *Computational Linguistics*, (19):313–330.

P. Resnik and D. Yarowsky. 1997. A perspective on word sense disambiguation methods and their evaluation. In *Proceedings of ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, Washington, D.C., USA.

P. Resnik. 1997. Selectional preference and sense disambiguation. In *Proceedings of ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, Washington, D.C., USA.

H. Schuetze, , and J. Pedersen. 1995. Information retrieval based on word senses. In *Proceedings 4th Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, USA.

D. Yarowsky. 1995. Unsupervised word sense disambiguation method rivaling supervised methods. In *Proceedings of the ACL.*