Computers and the Humanities **34:** 121–126, 2000.

GINGER II: An Example-Driven Word Sense Disambiguator

LUCA DINI[1,*], VITTORIO DI TOMASO[1] and FRÉDÉRIQUE SEGOND[2]
[1]*Centro per l'Elaborazione del Linguaggio e dell' Informazione*
(E-mail: {dini,ditomaso}@celi.sns.it); [2]*Xerox Research Centre Europe*
(E-mail: segond@xrce.xerox.com)

1. Introduction

Ginger II performs "all word" unsupervised Word Sense disambiguation for English, exploiting information from machine-readable dictionaries in the following way. To automatically generate a large, dictionary-specific semantically tagged corpus, we extract example phrases found in the text in the dictionary entries. We attach to each headword in this text the dictionary sense numbering in which the text was found. This provides the sense label for the head word in that context. GINGER II then builds a database of semantic disambiguation rules from this labelled text by extracting functional relations between the words in these corpus sentences.

As in GINGER I (Dini et al., 1998) the acquired rules are two-level rules involving the word level and/or ambiguity class level. In contrast to the algorithm used in GINGER I which was a variant of Brill's tagging algorithm (Brill, 1997), iteratively validating adjacency rules on a tagged corpus, GINGER II is now based on a completely nonstatistical approach. GINGER II directly extracts semantic disambiguation rules from dictionary example phrases using all functional relations found there. The dictionary, providing typical usages of each sense, needs no iterative validation.

GINGER II provides the following improvements over GINGER I:

- it relies on dictionary sense numbering to semantically tag dictionary examples
- it uses syntactic parsing of dictionary examples to extract semantic disambiguation rules

* We are grateful to Gregory Grefenstette and Christopher Brewster for their comments on earlier versions of this paper. Our thanks also go to Rob Gaizauskas, Wim Peters, Mark Steventson and Yorick Wilks for fruitful discussions about the methodology. Any remaining errors are our own.

– it uses two sets of semantic information to produce semantic disambiguation rules: the dictionary numbering provided from HECTOR (Atkins, 1993) and the 45 top level categories of WordNet.

We present below the building blocks of GINGER II as well as the components and the resources it uses.

2. The GINGER II Approach to Semantic Disambiguation within the SENSEVAL Competition

GINGER II is an unsupervised rule based semantic tagger which works on all words. Semantic disambiguation rules are directly extracted from dictionary examples and their sense numberings. Because senses and examples have been defined by lexicographers, they provide a reliable linguistic source for constructing a data base of semantic disambiguation rules.

GINGER II first builds, using dictionary examples, a data-base of rules which will then be applied to a new text and return as output a semantically tagged text. To learn the semantic disambiguation rules GINGER II uses the following components:

– The HECTOR Oxford Dictionary of English (OUP),
– the Xerox Incremental Finite State Parser for English (XIFSP),
– WordNet 1.6 (English).

GINGER II uses dictionary example phrases as *a semantically tagged corpus*.

When an example *z* is listed under the sense number *x* of a dictionary entry for the word *y*, GINGER II creates a rule which stipulates that, in usages similar to *z*, the word *y* has the meaning *x*.

Using XIFSP,[1] we first parse all the OUP example phrases for the selected SENSEVAL words. XIFSP is a finite state shallow parser relying on part of speech information only to extract syntactic functions without producing complete parse trees in the traditional sense.

GINGER II makes use of the syntactic relations: subject-verb, verb-object and modifier. Subject-object relations include cases such as passives, reflexives and relative constructions. Modifier relations include prepositional and adjectival phrases as well as relative clauses. GINGER II also uses XIFSP-extracted information about appositions. Altogether GINGER II uses 6 kinds of functional relations. Although XIFSP also extracts adverbial modification, GINGER II does not use it since our semantic disambiguation also uses, as shown below, the 45 top-level WordNet categories where all adverbs are associated with the same unique semantic class.

Once all OUP examples have been parsed, each word of each functional pair is associated with semantic information.

Two sets of semantic labels are used: the HECTOR sense numbers and the 45 WordNet top-level categories. HECTOR senses numbers are used to encode the example headword, while the WordNet tags are used to encode all remaining words appearing in the examples.

We use the relatively small number of WordNet top level categories so as to obtain sufficiently general semantic disambiguation rules. If we used only HECTOR sense numbers on the assumption that they were extended to all items in a dictionary, this would result in far too many semantic rules, each with a very limited range of application.

GINGER II deduces semantic rules[2] from these functional-semantic word pairs. These rules, like those of Brill,[3] are of two kinds. There are rules at the word level and rules at the ambiguity class level.

The example below, summarizes the above steps for the example *he shook the bag violently* registered under the HECTOR sense number (sen uid = 504338) of the OUP entry for *shake*:

First XIFSP extracts the syntactic functional relations: *SUBJ(he,shake)*, *DOBJ(shake,bag)*. These functional relation are then transformed into functional pairs. For instance, OBJ(shake, bag) becomes (shake $Hasobj$, bag $Hasobj^{-1}$).

These functional pairs are then augmented with semantic information: the target word, here *shake*, is associated with HECTOR sense numbers (504338,516519, 516517, 516518, ...516388) and the other word, here *bag* for the verb-object relation, is associated with its WordNet tags sense number (6, 23, 18, 5, 4):

These pairs can be read as:

$$(\text{shake}\frac{HasObj}{504338\_516519\_516517\_516518\_\ldots\_516388}, \text{bag}\frac{HasObj^{-1}}{6\_23\_18\_5\_4})$$

From this pair we extract the two following disambiguation rules:

- bag WRIGHT bi504338_bi516519_bi516517_bi516518_..._bi516388 bi504338
- b6_b23_b18_b5_b4 WRIGHT bi504338_bi516519_bi516517_bi516518_... _bi516388 bi504338

Where *b* represents the object relation and *bi* its inverse.

Rule (1) can be read as: the ambiguity class (504338, 516519, 516517, 516518, ...504388) disambiguates as *504338* when it has as object the word *bag*.

Rule (2) can be read as: the ambiguity class (504338, 516519, 516518, ...504388) disambiguates as *504338* when it has as object the WordNet ambiguity class (6, 23, 18, 5, 4).

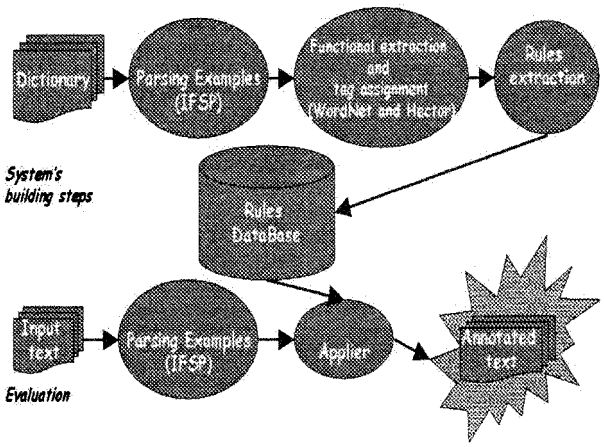

Figure 1. GINGER II: general architecture.

All dictionary example phrases are translated into semantic disambiguation rules and form a rule data-base.

GINGER II then applies these rules to any new input text and gives as output a semantically tagged text.

The applier, designed at CELI, uses several heuristics in order to drive the application of the disambiguating rules. In particular it exploits the notion of *tagset distance* in order to determine the best matching rule. The tagset distance is a metric which calculates the distance between two semantic classes within WordNet. The metric for computing the distance can be set by the user and can vary across several applications.

The applier first parses the new text and extract the functional dependencies. Then it extracts the potential matching rules. In case of conflict between rules, priority is given to word-level rules. If no word-level rules can apply then priority is given to the rule with the lowest or the highest (depending on the way user set the metrics) distance. The system is now complete and can run on all words of any text.

The general architecture of GINGER II is summarized in Figure 1.

3. Evaluation and Future Perspectives

For the overall SENSEVAL task GINGER II obtained a precision of 0.46 and a recall of 0.37 which is among the upper band of the unsupervised systems and among the average band of the supervised systems. But contrary to many systems in this range, GINGER II is a general system which works on all words and, regarding the SENSEVAL exercise, it did not take any advantage of knowing the word's part of speech in advance. Besides, because it directly uses HECTOR senses it did not have the disadvantage of the "mapping senses" phase.

We expect these results would improve since a new English tagger is now integrated in XIFSP which performs better than the one we used. Future versions of GINGER will include more functional relations and richer dictionary information. We are also interested in testing possible improvement in system performance using, for instance, triples rather than pairs, for example, using subject-verb-object relations rather than subject-verb, verb-object relations.

Encouraged by GINGER's robustness we are now integrating such a WSD component into XeLDA (Xerox Linguistic Development Architecture) making use of additional dictionary information such as collocates and subcategorization. All this information gives birth to a rule database attached to a particular dictionary leading to a dictionary based semantic tagger.[4]

Other areas of investigation concern deciding which semantic tags would be best to use, and associating weights with the semantic rules of the database.

The results of GINGER II indicates that even if dictionaries, seen as hand-tagged corpora, are reliable sources of information to extract semantic disambiguation rules from, they can be improved. We believe that one important way of creating better linguistic resources for many Natural Language processing tasks, is to enrich dictionaries with prototypical example phrases.

Because it is unsupervised, the method used within GINGER II can be applied to any language for which on-line dictionaries exist but for which significantly large semantically pre-tagged copora are not available.

Notes

[1] See Ait-Mokhtar and Chanod (1997).
[2] The rule extractor has been implemented as a Java program which parses dictionary entries in order to gather all the relevant information.
[3] See Brill (1995, 1997).
[4] See Segond et al. (1999).

References

Ait-Mokhtar, S. and J-P. Chanod. "Subject and Object Dependency Extraction Using Finite-State Transducers". In *Proceedings of Workshop on Information Extraction and the Building of Lexical Semantic Resources for NLP Applications, ACL*, Madrid, Spain, 1997.

Atkins, S. "Tools for Corpus-aided Lexicography: The HECTOR Project". In *Acta Linguistica Hungarica 41, 1992–1993.* Budapest, 1993 pp. 5–72.

Brill, E. "Transformation-based Error-Driven Learning and Natural Language Processing: A Case Study in Part of Speech Tagging". In *Computational Linguistics*, 1995.

Brill, E. "Unsupervised Learning of Disambiguation Rules for Part of Speech Tagging". In *Natural Language Processing Using Very Large Corpora*. Kluwer Academic Press, 1997.

Dini, L., V. Di Tomaso and F. Segond. "Error Driven Word Sense Disambiguation". In *Proceedings of COLING/ACL*, Montreal, Canada, 1998.

Miller, G. "Wordnet: An On-line Lexical Database'. *International Journal of Lexicography*, 1990.

Resnik, P. and D. Yarowsky. "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, 1997.

Segond, F., E. Aimelet, V. Lux and C. Jean. "Dictionary-driven Semantic Look-up". In *Computer and the Humanities*, this volume.

Yarowsky, D. "Unsupervised Word Sense Disambiguation Method Rivalizing Supervised Methods". In *Proceedings of the ACL*, 1995.

Wilks, Y and M. Stevenson. "Word Sense Disambiguation Using Optimised Combinations of Knowledge Sources". In *Proceedings of COLING/ACL*, Montreal, Canada, 1998.