Online Word Sense Disambiguation with Structural Semantic Interconnections

Roberto Navigli
Dipartimento di Informatica
Università di Roma "La Sapienza"
Roma, Italy
navigli@di.uniroma1.it

Abstract

In this paper we present an online implementation of a knowledge-based Word Sense Disambiguation algorithm called Structural Semantic Interconnections (SSI). We describe the system implementation and the user interface, and discuss the strengths and weaknesses of our approach.

1 Introduction

Word Sense Disambiguation (WSD) is the task of formalizing the intended meaning of a word in context by selecting an appropriate sense from a computational lexicon in an automatic manner.

The availability on the web of WSD algorithms can certainly contribute to tasks like the semantic indexing of online summaries, short news, web page titles, as well as other online applications.

Unfortunately, most of the WSD algorithms are not available online, or they are only available for download, requiring some training before the user can employ them. An important effort in this direction has been carried out with the release of the *WordNet::Similarity* package (Pedersen et al., 2004), which is also available online through a web interface. The package provides a variety of relatedness measures to determine the similarity between word pairs. Based on it, Patwardhan et al. (2005) provide a Word Sense Disambiguation package, called *SenseRelate*. The package is not available through a graphical interface, but the authors plan to develop it.

Among the state-of-the-art WSD disambiguation systems, i.e. the best performing algorithms at the Senseval-3 disambiguation competition, *Gambl* (Decadt et al., 2004), a memory-based algorithm, is available online[1], while *SenseLearner* (Mihalcea and Faruque, 2004), the second best-performing system, can be downloaded and run offline[2].

However, due to their trained nature, the application of these systems to open-domain sentences is not guaranteed to have the same performances as those obtained during the Senseval disambiguation exercises.

In this paper, we present the online implementation of *Structural Semantic Interconnections* (SSI), a state-of-the-art, knowledge-based WSD algorithm. Thanks to its untrained nature, SSI obviates the problems affecting the best-performing supervised systems.

First, we introduce the algorithm (Section 2). Then, we describe its architecture and implementation details (Section 3), as well as its user interface (Section 4). Finally, we discuss the performances of SSI (Section 5), and conclude (Section 6).

2 Structural Semantic Interconnections

The *Structural Semantic Interconnections* algorithm (SSI) is a WSD algorithm based on structural pattern recognition (Navigli and Velardi, 2004; Navigli and Velardi, 2005).

Given a word context $\sigma = w_1, w_2, \ldots, w_n$ and a lexical knowledge base, obtained by integrating WordNet (Fellbaum, 1998) with other resources, SSI selects that configuration of senses $\hat{s}_{w_1}, \hat{s}_{w_2}, \ldots, \hat{s}_{w_n}$ that maximizes the degree of mutual interconnection according to a measure of connectivity, that is, for each $w \in \sigma$:

$$\hat{s}_w = \arg\max_{s_w \in Senses(w)} f(s_w, \sigma),$$

[1] http://www.cnts.ua.ac.be/∼decadt/?section=wsd_demo
[2] http://mira.csci.unt.edu/∼senselearner

where f is a function of the semantic interconnections linking s_w to the senses of the words in σ, and $Senses(w)$ is the set of senses of w in the WordNet inventory.

Semantic interconnection patterns are relevant sequences of edges selected according to a context-free grammar, i.e. paths connecting pairs of word senses (dark nodes in Figure 1), possibly including a number of intermediate concepts (light nodes in Figure 1). This notion was inspired by the idea of lexical chains (Morris and Hirst, 1991).

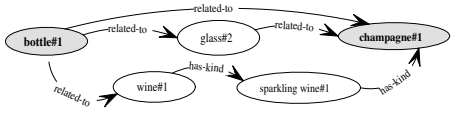

Figure 1: Examples of semantic interconnections.

For example, given the word context [*bottle-n*, *champagne-n*], the senses chosen by SSI with respect to WordNet are: [*bottle-n#1*, *champagne-n#1*][3], supported – among the others – by the pattern *bottle-n*#1 $\xrightarrow{related-to}$ *wine-n#1* $\xrightarrow{has-kind}$ *sparkling wine-n#1* $\xrightarrow{has-kind}$ *champagne-n#1*.

The outcome of the SSI algorithm is therefore not only a set of sense choices, but also a semantic graph encoding the interconnections that structurally justify those choices.

An excerpt of the manually written context-free grammar encoding valid semantic interconnection patterns for the WordNet lexicon is reported in Table 1. For further details the reader can refer to the literature (Navigli and Velardi, 2005).

3 Implementation of Online SSI

Two basic features made it possible to put online a fully-engineered version of the SSI algorithm: the construction of a large, optimized lexical knowledge base, and the implementation of the connectivity measure f in terms of the outcome of HITS (Kleinberg, 1998), a well-known page ranking algorithm.

3.1 The Lexical Knowledge Base

First, we enriched the WordNet lexicon with a number of *relatedness* relations, connecting pairs of related word senses. The enrichment is based on the acquisition of collocations from existing resources (like the Oxford Collocations, the Longman Language Activator, collocation web sites, etc.). Each collocation is mapped to the WordNet sense inventory in a semi-automatic manner (Navigli, 2005) and transformed into a *relatedness* edge.

[3]We indicate a word sense with the convention *w-p#i*, where w is a word, p its part of speech, and i its sense number in the WordNet inventory.

Table 1: An excerpt of the context-free grammar for the recognition of semantic interconnections.

$S \rightarrow S'S_1 \vert S'S_2 \vert S'S_3$
(start rule)
$S' \rightarrow e_{nominalization} \vert e_{pertainymy} \vert \epsilon$
(part-of-speech jump)
$S_1 \rightarrow e_{kind-of}S_1 \vert e_{part-of}S_1 \vert e_{kind-of} \vert e_{part-of}$
(hyperonymy/meronymy)
$S_2 \rightarrow e_{kind-of}S_2 \vert e_{relatedness}S_2 \vert e_{kind-of} \vert e_{relatedness}$
(hypernymy/relatedness)
$S_3 \rightarrow e_{similarity}S_3 \vert e_{antonymy}S_3 \vert e_{similarity} \vert e_{antonymy}$
(adjectives)

For each word sense s in the WordNet inventory, semantic interconnection patterns are exhaustively retrieved by exploring the lexicon, according to the predefined context-free grammar of valid patterns mentioned in Section 2. The resulting lexical knowledge base, stored as an optimized database, associates with each pair of word senses (s, s') the set of valid interconnection patterns between s and s'. Each pattern is assigned a weight based on its length (i.e. the contribution of a single interconnection $s \rightarrow^* s'$ is $\frac{1}{length(s \rightarrow^* s')}$). Given a configuration of n word senses, the associated semantic graph is therefore obtained in terms of $\binom{n}{2}$ queries, one for each possible pair combination. Each query takes $O(log(n))$ time, assuming the database is implemented as a B-tree.

3.2 Concept ranking

Previous implementations of the SSI algorithm (exhaustive and iterative) were not sufficiently fast to be accessible through a web interface, so we decided to reimplement the connectivity function f in terms of the HITS algorithm. HITS (Kleinberg, 1998) is a page ranking algorithm that calculates for each graph node v the degree of connectivity conveyed towards v (*authority* degree) and from v to the other nodes in the graph (*hub* degree).

Given a word context $\sigma = w_1, w_2, \ldots, w_n$ we define a graph $G = (V, E)$ such that $V = \bigcup_{w \in \sigma} Senses(w)$, and $(s, s') \in E$ if there exists at least one semantic interconnection between the senses s and s'. A weighted adjacency matrix L is

associated with G such that $L_{s,s'}$ is the sum of the weights of the interconnection patterns between s and s'. HITS is then applied to L to obtain the authority vector $\underline{a}$. This vector provides a degree of relevance for each node in V, that we call "confidence factor". For each word $w \in \sigma$, SSI selects $\arg\max_{s \in Senses(w)} a_s$ as the most appropriate sense of w in context σ, that is the sense s of w with the highest degree of confidence. If the confidence factor is below a fixed threshold, SSI does not choose any sense for w.

4 The SSI Interface

The system interface consists of three pages[4]:

Query page: In the first page, the user can type either a bag of words or a full sentence and apply the SSI algorithm by clicking on the `Disambiguate` button.

Part-of-speech specification: If there are words in the original query belonging to more than one part of speech, the user is asked to specify the appropriate part of speech for those words. Automated part-of-speech tagging is not performed in that the user is free to type a bag of words, rather than a fully grammatical sentence[5].

Result page: The outcome of the SSI algorithm is visualized. For each word w, this page shows the sense number possibly assigned to w, its WordNet definition and the degree of confidence for that sense choice. The page also shows a semantic graph encoding the semantic interconnections providing a justification of the sense choices.

The three steps are summarized in Figure 2. As interesting feature of the algorithm, compared with the other systems, is the visualization of the semantic graph encoding the interconnection patterns between the chosen senses. The user can click on a word sense and highlight the patterns connecting that sense to the other senses selected by the algorithm. In case the graph is too large, an automated, iterative pruning is applied until the overall number of vertices and patterns does not fall below a certain threshold. A screenshot of the result page is shown in Figure 3.

5 Evaluation

The SSI algorithm has been extensively evaluated in several tasks, including open-text Word Sense Disambiguation, gloss disambiguation, ontology learning, relation learning, etc.

Here we do not aim at reporting all these results (the interested reader can refer to Navigli and Velardi (2005)), instead we focus on the English all-words Word Sense Disambiguation tasks at Senseval-3 (Snyder and Palmer, 2004).

The performances are reported in Table 2. SSI performs better than the best unsupervised system, developed at IRST (Villarejo et al., 2004), and is some points below the two best-ranked supervised systems (Gambl and SenseLearner).

The untrained nature of SSI is indeed one of its major strengths, allowing it to be applied to any word context irrespective of its specificity, unlike most of the state-of-the-art trained algorithms.

The implementation of SSI in terms of a page ranking algorithm (SSI-HITS in the Table) only slightly affects the performances compared to our

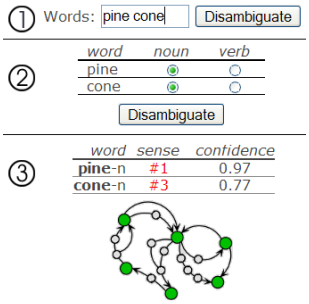

Figure 2: Online WSD in 3 steps: type the query, specify parts of speech, get the result.

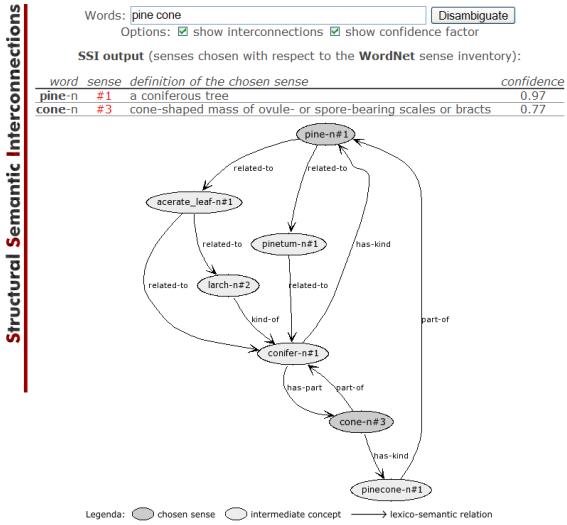

Figure 3: A screenshot of the outcome of SSI on the context [*pine-n*, *cone-n*].

[4] SSI is available online at http://lcl.di.uniroma1.it/ssi.

[5] We plan to include automatic tagging as an additional option of the query page.

Table 2: Performances of SSI compared to state-of-the-arts WSD systems in the all-words task at Senseval-3.

System	Precision	Recall
Gambl	65.2%	65.2%
SenseLearner	64.6%	64.6%
SSI	60.4%	60.4%
SSI-HITS	59.5%	59.3%
IRST-DDD	58.3%	58.2%

previous experiments with exhaustive and iterative implementations (Navigli and Velardi, 2005). On the other side, the speed up is impressive (from several minutes to one or two seconds per context).

Thanks to its knowledge-based nature, SSI produces a semantic graph as a justification for the sense choices assigned to a word context. An interesting application of this feature is in the validation of sense annotations, where validators need evidences for adjudicating a sense choice when annotators disagree (Navigli, 2006).

A weakness of our approach is in its dependency on the availability of general-purpose knowledge. If the SSI lexical knowledge base encodes poor knowledge for certain words, the algorithm is not able to find interconnections enabling the selection of the appropriate word senses. Moreover, as SSI treats a sentence as a bag of word, the complexity of a sentence affects the performance of the algorithm. In fact, a large, complex sentence is likely to perform worse than the average, due to the fact that many interconnections can be found between syntactically unrelated words.

6 Conclusions

We have presented the online implementation of SSI, a knowledge-based Word Sense Disambiguation algorithm. We described two important features that allow the algorithm to be available online: its large, optimized lexical knowledge base and its fast implementation with the HITS ranking algorithm.

We reported the performances of the SSI algorithm on the Senseval-3 all-words task and discussed the major strengths and weaknesses of our approach. In the near future, we plan to take syntax into account in order to overcome the bag-of-words effect on open-text disambiguation.

References

Decadt Bart, Hoste Véronique, Daelemans Walter, and van den Bosch Antal. 2004. GAMBL, Genetic Algorithm Optimization of Memory-Based WSD. *Proc. of ACL/SIGLEX Senseval-3*, Barcelona, Spain.

Fellbaum Christiane (ed.). 1998. *WordNet: an Electronic Lexical Database*, Cambridge, MIT press.

Kleinberg M. Jon. 1998. Authoritative sources in a hyperlinked environment. *Proc. of ACM-SIAM Symposium on Discrete Algorithms*.

Mihalcea Rada and Faruque Ehsanul. 2004. SenseLearner: Minimally Supervised Word Sense Disambiguation for All Words in Open Text. *Proc. of ACL/SIGLEX Senseval-3*, Barcelona, Spain.

Morris Jane and Hirst Graeme. 1991. Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text. *Computational Linguistics*, 17(1).

Navigli Roberto. 2006. Experiments on the Validation of Sense Annotations Assisted by Lexical Chains. *Proc. of* 11^{th} *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Trento, Italy.

Navigli Roberto and Velardi Paola. 2004. Learning Domain Ontologies from Document Warehouses and Dedicated Websites. In *Computational Linguistics*, 30(2). MIT Press.

Navigli Roberto and Velardi Paola. 2005. Structural Semantic Interconnections: a Knowledge-Based Approach to Word Sense Disambiguation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(7).

Navigli Roberto. 2005. Semi-Automatic Extension of Large-Scale Linguistic Knowledge Bases. In Proc. of 18^{th} FLAIRS Conference. Clearwater Beach, Florida.

Patwardhan Siddhart, Banerjee Satanjeev and Pedersen Ted. 2005. SenseRelate::TargetWord - A Generalized Framework for Word Sense Disambiguation. *Proc. of* 20^{th} *National Conference on Artificial Intelligence*, Pittsburgh, PA.

Pedersen Ted, Patwardhan Siddhart, and Michelizzi Jason. 2004. WordNet::Similarity - Measuring the Relatedness of Concepts. *Proc. of* 19^{th} *National Conference on Artificial Intelligence*, San Jose, CA.

Snyder Benjamin and Palmer Martha. 2004. The English all-words task. In *Proc. of ACL 2004 SENSEVAL-3 Workshop*, Barcelona, Spain.

Villarejo Luís, Màrquez Lluis, Agirre Eneko, Martnez David, Magnini Bernardo, Strapparava Carlo, McCarthy Diana, Montoyo Andrés and Suérez Armando 2004. The Meaning system on the English all-words task. *Proc. of ACL/SIGLEX Senseval-3*, Barcelona, Spain.