Extending and Enriching WordNet with OntoLearn

Roberto Navigli[1], Paola Velardi[1], Alessandro Cucchiarelli[2], and Francesca Neri[2]

[1] Università di Roma "La Sapienza", Dipartimento di Informatica, Via Salaria 113
I-00198 Roma, Italy
{velardi,navigli}@dsi.uniroma1.it
[2] Università Politecnica delle Marche, D.I.I.G.A., Via Brecce Bianche 12, I-60131
Ancona, Italy
{cucchiarelli,neri}@diiga.univpm.it

Abstract. OntoLearn is a system for word sense disambiguation, used to automatically enrich WordNet with domain concepts and to disambiguate WordNet glosses. We summarize the WSD algorithm used by Ontolearn, called *structural semantic interconnection*, and its main applications.

1 The Structural Semantic Interconnection Algorithm

OntoLearn is a system for the automatic extraction of concepts from texts that has been developed over the past few years at the University of Roma "La Sapienza", with the contribution of several other researchers in Italy. The system has been used and is being enhanced in the context of European and national projects[1].

The key task performed by OntoLearn is semantic disambiguation, a task we applied to various problems, namely:

- associate complex domain terms (e.g. *local area networks*) with the appropriate WordNet synsets (e.g. respectively: {*local#2*} (adj.), {*area#1, country#4*}, {*network#2, communications network#1* }) in order to enrich WordNet with new domain concepts and learn domain-specific ontologies [2, 3];
- disambiguate WordNet glosses [1];
- disambiguate words in a query for sense-based web query expansion [4].

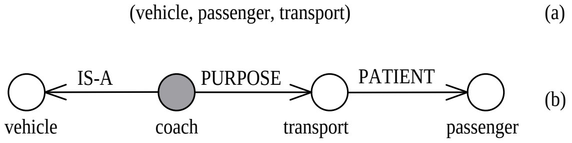
Semantic disambiguation is performed using a method we have named *structural semantic interconnection (SSI)*, a structural approach to pattern recognition, that uses graphs to describe the objects to analyze (word senses) and a context free grammar to detect common semantic patterns between graphs. Sense classification is based on the number and type of detected interconnections.

[1] Harmonise IST-13015 in the Tourism domain; WonderWeb IST-2001-33052 on ontology infrastructure for the semantic web, and the national MIUR-SP6 project on Web Learning.

In this paper we provide a high-level intuitive description of the SSI algorithm, which is rather complex. A thorough description is in [3], but a complete reformalization is in progress.

SSI is a kind of *structural pattern recognition.* Structural pattern recognition [5] has proven to be effective when the objects to be classified contain an inherent, identifiable organization, such as image data and time-series data. For these objects, a representation based on a “flat” vector of features causes a loss of information which negatively impacts on classification performances. The classification task in a structural pattern recognition system is implemented through the use of grammars which embody precise criteria to discriminate among different classes. The drawback of this approach is that grammars are by their very nature application and domain-specific. However, machine learning techniques may be adopted to learn from available examples.

Word senses clearly fall under the category of objects which are better described through a set of structured features. Compare for example the following two feature-vector (a) and graph-based representations (b) of the WordNet 1.7 definition of *coach#5* (*a vehicle carrying many passengers, used for public transport*):



The graph representation shows the semantic interrelationships among the words in the definition, in contrast with the “flat” feature vector representation.

Provided that a graph representation for alternative word senses in a context is available, *disambiguation can be seen as the task of detecting certain “meaningful” interconnecting patterns among such graphs.* We use a context free grammar to specify the type of patterns that are the best indicators of a semantic interrelationship and to select the appropriate sense configurations accordingly.

To automatically generate a graph representation of word senses, we use the information available in WordNet 1.7 augmented with other on-line lexical resources, such as semantically annotated corpora, list of domain labels, etc. Figure 1 is an example of the semantic graph generated for sense #2 of *bus.* In the figure, nodes are word senses, arcs are semantic relations. The following semantic relations are used: *hyperonymy* (car *is a kind of* vehicle, denoted with $\xrightarrow{kind-of}$), *hyponymy* (its inverse, $\xrightarrow{has-kind}$), *meronymy* (room *has-part* wall, $\xrightarrow{has-part}$), *holonymy* (its inverse, $\xrightarrow{part-of}$), *pertainymy* (dental *pertains-to tooth* $\xrightarrow{pert}$), *attribute* (dry *value-of* wetness, $\xrightarrow{att}$), *similarity* (beautiful *similar-to* pretty, $\xrightarrow{sim}$), *gloss* ($\xrightarrow{gloss}$), *topic* ($\xrightarrow{topic}$), *domain* ($\xrightarrow{dl}$). *Topic, gloss* and *domain* are extracted respectively from annotated corpora, sense definitions and domain labels. Every other relation is explicitly encoded in WordNet.

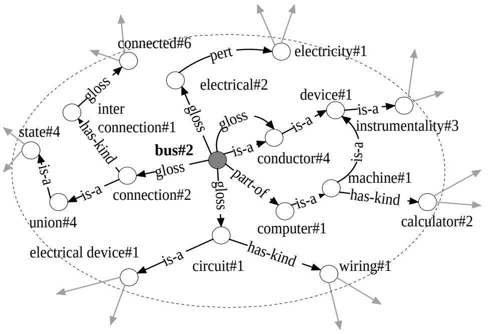


Fig. 1. Example of derived semantic graph for sense #2 of *bus* in WordNet

The basic *semantic disambiguation step* of the SSI algorithm is described hereafter. Let $C = \{w_0, w_1, \ldots, w_{n-1}\}$ be a list of co-occurring words. In a generic step i of the algorithm, let $D = \{S^a_j, S^b_i, \ldots, S^c_m\}$ be a list of semantic graphs, one for each of the words $W_D = \{w_a, w_b, \ldots, w_c\}$, $W_D \subseteq C$ already disambiguated in steps $1, 2, \ldots, i-1$. Let further $P = \{w_p, w_q, \ldots, w_z\}$ be the list of words in C that are still ambiguous, where $W_D \cup P = C$ and $W_D \cap P = \emptyset$. D is called the *semantic context* of P.

Until all words $w_r \in P$ have been analyzed, do:

- Let $S_{w_r} = \{S^r_1, S^r_2, \ldots, S^r_k\}$ be the set of senses of w_r, each represented by a semantic graph.
- Find the best sense $S^r_l \in S_{w_r}$, according to a classification criterion $\Im$. If $\Im$ is not met, skip to a subsequent word in P.
- Add S^r_l to D, delete w_r from P.

Repeat until either P is empty, or no new words are found that meet the classification criterion $\Im$. We now describe the classification criterion $\Im$.

Classification is based on searching specific interconnection patterns between some of the semantic graphs in D and the semantic graphs associated to senses of a word w_r. Each matching pattern increases the weight $w(S^r_k)$ of the correspondent word sense. The classification criterion assigns sense S^r_l to word w_r if $w(S^r_l) = argmax_k(w(S^r_k))$ and $w(S^r_l) \geq \beta$, where β is a fixed threshold.

Interconnection patterns are described by a context free grammar. For the sake of space we are unable to give here an account of the grammar. An intuitive example of an elementary pattern between two semantic graphs S^i_j S^h_k is informally described by the following sentence: "The graph S^i_j is connected to the graph of S^h_k through a holonymy path". For example: $window\#7 \xrightarrow{part-of}$

computer screen#1. The grammar includes several complex patterns made of elementary ones, e.g. *holonymy-hyperonymy* sequences. We are now left with the problem of how to initialize the list D. Initialization depends upon the specific disambiguation task being considered. In OntoLearn, we experimented the SSI algorithm for three disambiguation tasks:

1. Disambiguation of the words in a WordNet gloss (e.g. *retrospective#1: "an exhibition of a representative selection of an artist's life work"*).
2. Disambiguation of words in a query (e.g queries from TREC web retrieval tasks: *"how we use statistics to aid our decision making?"*).
3. Disambiguation of complex terms (e.g. *connected bus network*).

In task 1, D is initialized with the sense described by the gloss under consideration, possibly augmented with the senses of all unambiguous words in the gloss, e.g. for the *retrospective* example, we have: D={*retrospective#1, statue#1, artist#1*} and P={*work, exhibition, life, selection, representative, art*}.

In task 1, we are sure that D in step 1 includes at least one semantic graph, that of the synset whose gloss we are disambiguating. In the other two tasks, either one of the words at least in set C is monosemous, or the algorithm begins with an initial guess, selecting the most probable sense of the less ambiguous word. If the total score is below a given threshold, the algorithm is then repeated with a different initial guess.

We now consider a complete example of the SSI algorithm for the complex term disambiguation task: *connected bus network*. As no word is monosemous, the algorithm makes a guess about the sense of the less ambiguous word, namely *network*. The only sense of *network* passing the threshold is #3, "an intersected or intersecting configuration or system of components". Initially we have $D = \{network\#3\}$ and $P = \{connected, bus\}$. At the first step, the following pattern involving the domain label relation is matched: $network\#3 \xrightarrow{dl} connected\#6$ (i.e. the two concepts have the same domain label "computer_science"). So, $D = \{network\#3, connected\#6\}$ and $P = \{bus\}$. Finally, linguistic parallelism (i.e. the two concepts have a common ancestor) and domain label patterns provide the correct indication for the choice of the second sense of bus, "an electrical conductor that makes a common connection between several circuits". The final configuration is thus $D = \{network\#3, connected\#6, bus\#2\}$ and $P = \emptyset$.

2 Evaluation of SSI algorithm

Each of the three tasks described in previous sections have been evaluated using standard (when available) and ad-hoc test bed. A summary evaluation for each task is shown in the three tables below. Details are provided in previously referenced papers. The baseline in Tables 1 and 3 is computed selecting the first WordNet sense (the most probable according to authors). In Table 3, in order to obtain a 100% recall, sense #1 is selected when no interconnections are found for appropriate sense selection. Furthermore, to increase the set D at step 1,

Table 1. Summary of experiments on gloss disambiguation

Domains	#Glosses	#Words	#Disamb. words	#Disamb. words ok	Recall	Precision	Baseline Precision
Tourism	305	1345	636	591	47.28%	92.92%	82.55%
Generic	100	421	173	166	41.09%	95.95%	67.05%

Table 2. Summary of experiments on sense-based query expansion

First 20 TREC 2002 web track queries	Without sense expansion (baseline)	With sense expansion (best expansion strategy)
Avg. n. of correct retrieved GOOGLE pages over first 10	5.12	6.29
% of increase over baseline	-	22.76%

we jointly disambiguate many terms having word strings in common (e.g. *public transport service, bus service, coach service,* etc.).

As shown in Table 3 and in other papers, the performance of the SSI algorithm in the WordNet extension task is between 84% and 89% depending upon domains. Furthermore, the extended WordNet may include other types of errors (e.g. inappropriate terminology), therefore it needs to be inspected by domain experts for refinements. To facilitate the human task of evaluating new proposed concepts, we defined a grammar for each semantic relation type to compositionally create a gloss for new complex concepts in an automatic fashion.

Let $cc(h,k) = S_j^k \overset{sem\_rel}{\longrightarrow} S_l^h$ be the complex concept associated to a complex term $w_h w_k$ (e.g. *coach service, or board of directors*), and let:

<**GNC**> be the gloss of the new complex concept $cc(h,k)$;
<**HYP**> the direct hyperonym of $cc(h,k)$ (e.g. respectively, *service#1* and *board#1*);
<**GHYP**> the gloss of HYP;
<**FPGM**> the main sentence of the correct gloss of the complex term modifier (e.g respectively: *coach, director*).

We provide here two examples of rules for generating GNC:

1. if sem_rel=*attribute*, <GNC>::=**a kind of** <HYP>, <GHYP>, <FPGM>

Table 3. Summary of experiments on complex term disambiguation

# of complex terms (tourism domain)	Average words per term	Precision	Baseline Precision
650	2.2	84.56%	79.00%

2. if sem_rel=*purpose*, <GNC>::=**a kind of** <HYP>, <GHYP>, **for**<FPGM>

The following are examples of generated definitions for rules 1 and 2.

COMPLEX TERM: Traditional garment *(tourism)*

<HYP>::=garment#1
<GHYP>::=an article of clothing
<FPGM>::=consisting of or derived from tradition
<GNC>::=**a kind of** garment, an article of clothing, consisting of or derived from tradition

COMPLEX TERM: Classification rule *(computer science)*

<HYP>::=rule#11
<GHYP>::=a standard procedure for solving a class of problems
<FPGM>::= the basic cognitive process of arranging into classes or categories
<GNC>::=**a kind of** rule, a standard procedure for solving a class of problems, **for** the basic cognitive process of arranging into classes or categories

3 Conclusion

Current research on OntoLearn follows two directions: on the theoretical side, we are trying to obtain a better formalization of the structural semantic interconnection methodology through the use of graph grammars. On the application side, we are extending the type of semantic information that is extracted by Ontolearn. Furthermore, we are augmenting the information represented in semantic graphs, using other semantic resources, such as FrameNet.

References

[1] Gangemi, A., Navigli, R., Velardi, P.: Axiomatizing WordNet: a Hybrid Methodology. Workshop on Human Language Technology for the Semantic Web and Web Services at the 2003 International Semantic Web Conference, Sanibel Island, Florida, USA (2003)

[2] Missikoff, M., Navigli, R., Velardi, P.: Integrated Approach for Web Ontology Learning and Engineering. IEEE Computer, November 2002

[3] Navigli, R., Velardi, P., Gangemi, A.: Corpus Driven Ontology Learning: a Method and its Application to Automated Terminology Translation. IEEE Intelligent Systems **18** (2003) 22–31

[4] Navigli, R., Velardi, P.: An Analysis of Ontology-based Query Expansion Strategies. Workshop on Adaptive Text Extraction and Mining at the 14th European Conference on Machine Learning, Cavtat-Dubrovnik, CROATIA (2003)

[5] Olszewski, R.T.: Generalized Feature Extraction for Structural Pattern Recognition. In Time-Series Data, PhD dissertation, Carnagie Mellon University CMU-CS-01-108 (2001)