Semi-Automatic Extension of Large-Scale Linguistic Knowledge Bases

Roberto Navigli

Dipartimento di Informatica
Università di Roma "La Sapienza"
Via Salaria, 113 – 00198 Roma Italy
navigli@di.uniroma1.it

Abstract

Linguistic resources are essential for the success of many AI tasks. Building a new lexical resource from scratch or combining heterogeneous resources is not only complex and time-consuming, but can also lead to knowledge inconsistency and redundancy.
In this paper, we present a novel method for the large-scale semantic enrichment of a computational linguistic resource. To this end, with the aid of a controlled vocabulary, we identified a set of *representative concepts*, i.e. a restricted, but meaningful number of concepts from WordNet, such that each of them can replace any of its descendants in the taxonomical hierarchy without a substantial loss of information in natural language sentences (e.g. *restaurant#1* is a representative for *bistro#1* or *cybercafe#1*). Then, we manually enriched these representative concepts with collocations extracted from a variety of linguistic resources. After this manual step, representative concepts are still related with *words*, rather than with concepts (e.g. for *taxi#1*: *fare*, *passenger*, *driver*, etc.). The final step is to automatically disambiguate these terms, using a word sense disambiguation algorithm named Structural Semantic Interconnections (SSI). SSI is a knowledge-based WSD algorithm that is particularly performant when words in a context are highly semantically associated. As a result, the precision of this automatic disambiguation step is very high, to a point that residual disambiguation errors could be tolerated. In any case, since SSI provides semantic patterns to justify its sense choices, manual corrections by human annotators would be considerably facilitated, achieving a significant speed-up in semantic annotation. Furthermore, SSI helps in supporting a consistency of the lexical knowledge base.

1. Introduction

Relevant tasks in the field of Artificial Intelligence have a strong bias on language (e.g. Word Sense Disambiguation, Information Retrieval, Question Answering, etc.).
The contribution of linguistic resources is then essential to the success of these tasks. Since the early Eighties, machine readable dictionaries have been developed with the aim of making explicit knowledge available for computational tasks. With the advent of computational lexicons and ontologies, the focus has been shifted on how to effectively acquire, structure, and exploit such massive quantities of knowledge. Unfortunately, large-scale linguistic resources, when available, do not seem to provide all the semantic information needed for complex AI tasks. For instance, WordNet (Fellbaum, 1998) has a rich taxonomy, but encodes few (and sometimes incomplete) conceptual relations. Other resources, like CyC (Lenat, 1995) or Microkosmos (Mahesh and Nirenburg, 1995), are only partially available and cannot be fully exploited or investigated.
The recent construction of resources built around the idea of frames (FrameNet (Baker et al., 1998)) or focused on verbs (VerbNet (Kipper et al., 2000), PropBank (Kingsbury and Palmer, 2002)) constitutes an interesting, but problematic contribution. The main obstacles for their extensive exploitation are indeed their heterogeneity, as well as their limited size (usually some thousand elements).
Combining heterogeneous resources as a larger-scale effort is a hard task and does not often lead to a coherent result, because of the different philosophies followed in constructing the resources to be integrated. Furthermore, structuring wide-coverage knowledge is *per se* a complex and time-consuming task, potentially leading to strong inconsistencies.
In this paper, we present a novel approach to the semi-automatic construction of a conceptually rich, large-scale linguistic resource, built on top of WordNet. We started from the Longman controlled vocabulary, a set of basic words used to define all the terms in the dictionary. Each defining word corresponds to a number of WordNet concepts, i.e. its semantic interpretations. As *defining words* allow to describe all the entries of the Longman dictionary, in Section 2 we will show that the corresponding WordNet concepts, that we call *representative concepts*, subsume a good portion of the WordNet sense inventory without a substantial loss of information.
Then, for each representative concept (i.e. for each sense of a defining word), we extracted collocation triples (l, w, t) from existing lexical resources (e.g. *Oxford Collocations*, *Longman Language Activator*, etc.), where l is a relationship label, w a defining word, and t is a word collocated with w. For each collocation, we manually disambiguated w, i.e. we chose the corresponding representative concept r_w.

Finally, we applied an automatic procedure for the disambiguation of the collocation context of each representative concept r_w, i.e. the set of terms { $t : r_w$ is the appropriate sense for w in (l, w, t) }. To this end, we applied SSI, described in Section 3, a word sense disambiguation algorithm based on lexico-semantic patterns, with promising results in both precision and recall. We repeated the experiment by enriching the SSI knowledge base with a large number of manually disambiguated relation instances. The results are illustrated in Section 4. Finally, Section 5 discusses our contribution and future work.

2. Extending Large-Scale Resources

Automating the task of building large-scale linguistic resources is a necessary step, but also imposes a huge effort on the side of knowledge integration and validation.

Starting from a widespread computational lexicon such as WordNet (Fellbaum, 1998) allows concentrating on the difficult task of populating it with new semantic relations between concepts, overcoming the question of constructing a resource from scratch.

Still, relation population is a complex task for many reasons. First, assuming that each concept be connected on average to some tenth of concepts would lead to the extraction, insertion and validation of millions of semantic relation instances. Second, problems of redundancy (e.g. "*apples* have *flesh*" and "*pears* have *flesh*" both covered by "*fruits* have *flesh*"), heterogeneity and inconsistency (e.g. "*bird* related-to *flight*", but a *penguin* does not fly) would certainly emerge.

As a solution, we propose a novel approach aiming at determining and enriching a significant and manageable portion of the WordNet sense inventory.

Identification of Representative Concepts

We started from the *controlled vocabulary* (*CV*) of the *Longman Dictionary of Contemporary English* (*LDOCE*). A *CV* is a selection of the most frequent English words, called *defining words*, used to define all the words in a dictionary[1]. For each word $w \in CV$, WordNet provides one or more senses, i.e. its semantic interpretations. We denote the set of such concepts by *RC*.

The Longman *CV* includes 1,382 nouns, 354 adjectives and 314 verbs, corresponding respectively to 6,251, 1,810 and 2,310 WordNet concepts, called synsets (we did not take adverbs into account).

We name *RC*, i.e. the set of semantic interpretations of terms from the controlled vocabulary *CV*, the set of *representative concepts*. Concepts in *RC* indeed "represent" all the other concepts in the sense inventory with a certain degree of generality. A representative concept can replace most of its hyponyms (i.e. descendants in the taxonomical hierarchy) in any context without a substantial loss of information (e.g. *restaurant#1*[2] is a representative for *bistro#1* or *cybercafe#1*, *beverage#1* is a representative for *cyder#1* or *soda#2*, etc.). This statement is strongly supported by the figures in Table 1, discussed in the following paragraphs.

We say that a generic concept c is *covered* by a set of concepts S if exists $c' \in S$ such that c' is an ancestor of c in the taxonomical hierarchy. We calculated the *coverage* of the WordNet sense inventory with respect to *RC* as the number of covered concepts over the total number of WordNet concepts. Due to the WordNet structure, this measure is applicable only to nominal and verbal synsets. For adjectives, the coverage was calculated as the number of synsets containing a concept $c' \in RC$ in their adjective cluster.

The number of nominal concepts covered by *RC* is very high (98.7%). 75% of the verbs are covered, due to the large number of roots in the WordNet verb taxonomy, while around 42% of the adjectival concepts is covered, which is reasonable as the number of adjective clusters largely exceeds the amount of adjectival concepts in *RC*.

In the following, we focus on nominal concepts, but we plan to extend our work to the other major part-of-speech categories.

Table 1. Representative concepts in figures.

	Nouns	Adj.	Verbs
# of defining words (|*CV*|)	1,382	354	314
# of representative concepts (|*RC*|)	6,251	1,810	2,310
% concepts "covered"	98.7%	42.44%	75.56%
Average distance from the nearest representative	2.53	N/A	1.75
Average depth of a representative	3.64	N/A	0.75

The average depth of a representative nominal concept in the taxonomy is between 3 and 4 (at this depth we find concepts like *beverage#1*, *coffee#1*, *building#1*, *door#1*, etc.), while the distance of a nominal synset from its closest representative (in terms of number of hypernym edges connecting the two concepts) is on average between 2 and 3 edges (e.g. between *scooter#2* and *vehicle#1* or between *detective story#1* and *fiction#1*).

The identification of a complete set of representative concepts allows on one side to reduce the mass of work needed to populate a large-scale resource with semantic relations, on the other side to specify relations at a medium level of abstraction, so that inferences can be made for more specific concepts. For instance, in the sentence "the murderer used a stiletto to kill the victim", good

[1] A similar effort has been carried out in the Oxford Advanced Learners' Dictionary.

[2] By *w#i*, *w-a#i*, *w-v#i* we denote, respectively, the *i*-th sense of the noun, adjective and verb *w* in WordNet.

representatives for *murderer* and *stiletto* can be, respectively, *criminal* and *knife*.

Extraction of Collocations

The next step is to extract collocations from a variety of existing lexical resources connecting words in CV with other terms. A collocation is represented as a triple:

$$(l, w, t) \in L \times CV \times V$$

where L is the set of relation labels (i.e., $L = \{$ *relatedness*, *meronymy*, *attribute*, … $\}$), V is the set of all terms in the WordNet dictionary, and $CV \subseteq V$. For instance, (*relatedness*, *fruit*, *confiture*) and (*meronymy*, *wall*, *brick*) are collocation triples. Notice that L includes both lexical and semantic relations, although terms in CV and V are still ambiguous (i.e. not yet disambiguated). Furthermore, relations in L are either new (e.g. *relatedness*) or extend the existing ones with new instances (e.g. the missing *meronymy* link in WordNet between *brick* and *wall*).
Collocation triples are collected from the following resources (through either automatic extraction or manual editing, depending on the availability in electronic format):

- *WordNet glosses*: glosses (i.e. word definitions) and usage examples contain a number of words that are related to the defined concept. For instance, *coffee#1* is defined as *a* ***beverage*** *consisting of an* ***infusion*** *of* ***ground coffee beans****; "he ordered a* ***cup*** *of coffee"*.
- *Collocation web sites*: *Lexical Freenet* (*www.lexfn.com*), *OneLook* (*www.onelook.com*) and Sharp's *JustTheWord* provide collocations for words and multi-word expressions (e.g. collocations for *engine* are *aircraft*, *petrol*, *combustion*, *car*, etc.).
- *Oxford collocations* (Lea, 2002): this resource provides linguistic interrelations, like noun attributes (e.g. ***powerful*** as an attribute for *engine*), phrases (***roar*** of the *engine*), compounds (*engine* ***speed***, *engine* ***room***, etc.), collocated verbs (e.g. ***start*** the *engine*), etc.
- *The Longman Language Activator* (Longman, 2003): the words in the *Activator* are organized into groups, expressing basic ideas (e.g. *music*, *meal*, *medical treatment*, etc.). Compounds, collocations, attributes, etc. are provided for each word belonging to a group.

Each collocation crafted from these heterogeneous resources is converted to its triple representation $(l, w, t) \in L \times CV \times V$, where the set L includes the following relations:

- *Relatedness*: a generic semantic relation, to be further specialized (e.g. between *milk* and *coffee*, *car* and *driver*, etc.).
- *Meronymy*: this is an extension of the relation defined in WordNet (e.g. between *brick* and *wall*, *smoke* and *particle*, etc.).
- *Attribute*: a value t for a property w (e.g. *hot* is an attribute of *temperature*).
- *Property*: a property t holding for w (e.g. *temperature* is a property of *liquid*, *breed* is a property of *animal*, etc.); a property can assume different attributes.
- *Quantity*: t indicates a quantity of w (e.g. *bunch* expresses a quantity for *flowers*, *bag* for *potatoes*, etc.).
- *Predicate*: t is a predicate for w (as *to cook* for *food*, *to grow* for *flower*, etc.).

Relations like *quantity* and *predicate* concern syntactic constraints or semantic preferences, while *relatedness*, *property* and *attribute* express semantic interconnections.

Semantic Disambiguation of Collocations

At this stage, relations are still between collocated terms, and not between concepts. For interpreting terms in collocation triples (l, w, t) as concepts, we proceeded as follows: for each triple we manually disambiguated the defining word w from the controlled vocabulary by choosing the appropriate representative concept $r_w \in RC$. Notice that the mapping from w to r_w is trivial for collocations extracted from WordNet glosses (a gloss provides collocations about the synset it defines). In other cases, e.g. the Oxford Collocations and the Longman Activator, collocations are provided for each sense of w in the respective sense inventory: senses of w are defined in natural language, so that they can be manually mapped to the corresponding WordNet synsets.
As a result, each triple $(l, w, t) \in L \times CV \times V$ is mapped to the corresponding triple $(l, r_w, t) \in L \times RC \times V$. Each representative concept $r \in RC$ is now linked to a number of terms (given by the set $\{ t \in V \mid (l, r, t)$ is a valid triple $\}$), to be still disambiguated with respect to WordNet.
Due to the strong semantic, possibly structural, correlation between a term t and a representative concept r, we chose to apply the Structural Semantic Interconnections (SSI) algorithm (Navigli and Velardi, 2004) for the disambiguation of these terms. A description of the algorithm is provided in the next section.
After the disambiguation step, relation triples are finally encoded as $(l, r, s) \in L \times RC \times C$, where C is the complete set of WordNet concepts and $RC \subseteq C$ (as representative concepts are still concepts). This allows the definition of our extended lexical resource as $O = (C, R)$, where C is again the set of concepts, $R = R_{WN} \cup \{$ *relatedness*, *meronymy*, *attribute*, *property*, *quantity*, *predicate* $\}$, and R_{WN} is the set of WordNet relations. Notice that, while relations in R_{WN} are defined over $C \times C$, relations in $R \setminus R_{WN}$ are defined over $RC \times C$.

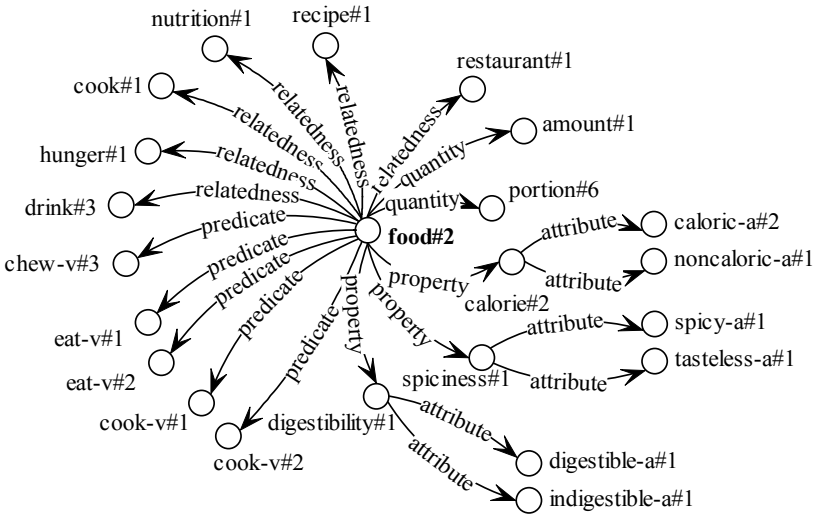

Figure 1. Semantic relations connecting *food#2* to other concepts.

As a fragment of the extended resource, consider the semantic relations manually crafted for *food#2* illustrated in Figure 1.

3. Structural Semantic Interconnections

SSI (Structural Semantic Interconnections (Navigli and Velardi, 2004)) is a word sense disambiguation algorithm based on structural pattern matching (Bunke and Sanfeliu, 1990). It disambiguates words in contexts using a "core" semantic knowledge base including WordNet and other lexical resources. The algorithm consists of an initialisation and an iterative step.

In a generic iteration of the algorithm the input is a list of co-occurring terms $T = [\ t_1, \ldots, t_n\ ]$ and a list of associated senses $I = [S^{t_1}, \ldots, S^{t_n}]$, i.e. the semantic interpretation of T, where S^{t_i} is either the chosen sense for t_i (i.e., the result of a previous disambiguation step) or the *null* element (i.e., the term is not yet disambiguated).

A set of pending terms is also maintained, $P = \{t_i \mid S^{t_i} = null\}$. I is named the semantic context of T and is used, at each step, to disambiguate new terms in P.

The algorithm works in an iterative way, so that at each stage either at least one term is removed from P (i.e., at least a pending term is disambiguated) or the procedure stops because no more terms can be disambiguated. The output is the updated list I of senses associated with the input terms T.

Initially, the list I includes the senses of monosemous terms in T. If no monosemous terms are found, the algorithm makes an initial guess based on the most probable sense of the less ambiguous term. The initialisation policy is further adjusted depending upon the specific WSD task considered.

During a generic iteration, the algorithm selects those terms t in P showing an interconnection between at least one sense S of t and one or more senses in I. Relevant interconnections are encoded in a context-free grammar describing meaningful lexico-semantic patterns. The likelihood for a sense S of being the correct interpretation of t is given by a function of the weights of patterns connecting S to other synsets in I.

In the case of collocation disambiguation, for each representative concept $r_w \in RC$, its disambiguation context T can be populated with w (the defining word in the CV whose semantic interpretation is given by r_w) and the set of terms $t_1, t_2, \ldots, t_n$ related to r_w by the respective relation triples $(label_1, r_w, t_1)$, $(label_2, r_w, t_2)$, …, $(label_n, r_w, t_n)$ extracted in the previous section.

Then SSI can be applied to the context $T = [w, t_1, t_2, \ldots, t_n]$, by fixing r_w as the correct sense of w in I.

As an example, consider the representative concept *restaurant#1*, exposing, among the others, the following collocations: *menu*, *chain*, *waiter*, *cafeteria*.

The initial context T is given by [*restaurant*, *menu*, *chain*, *waiter*, *cafeteria*]. I is initialised to [*restaurant#1*, -, -, -, *cafeteria#1*] (*cafeteria* is monosemous) and P = { *menu*, *chain*, *waiter* }.

During the first iteration, SSI detects, among the others, the following interconnections between senses in I and the first sense of the word *menu* ("a list of dishes available at a restaurant"):

$$\text{menu\#1} \xrightarrow{gloss} \text{restaurant\#1} \quad (\textit{gloss pattern})$$

$$\text{menu\#1} \xrightarrow{gloss} \text{restaurant\#1} \xleftarrow{gloss} \text{cafeteria\#1} \quad (\textit{gloss+gloss pattern})$$

$$\text{menu\#1} \xrightarrow{gloss} \text{restaurant\#1} \xleftarrow{kind-of} \text{cafeteria\#1} \quad (\textit{gloss+hypernymy})$$

As a result, I and P are updated as follows:

I = [*restaurant#1*, *menu#1*, -, -, *cafeteria#1*]
P = { *chain*, *waiter* }

During the second iteration, the fourth sense of *chain* is selected and added to *I*, thanks to the following semantic patterns:

$$\text{chain\#4} \xrightarrow{has-kind} \text{restaurant} \quad \text{chain\#1} \xrightarrow{\text{has-part}} \text{restaurant\#1}$$
(*hypernymy+meronymy pattern*)

$$\text{chain\#4} \xrightarrow{gloss} \text{restaurant\#1}$$ (*gloss pattern*)

$$\text{chain\#4} \xrightarrow{gloss} \text{restaurant\#1} \xleftarrow{\text{gloss}} \text{menu\#1}$$ (*gloss+gloss pattern*)

After this step, *I* = [*restaurant#1*, *menu#1*, *chain#4*, -, *cafeteria#1*] and *P* = { *waiter* }. Finally, *waiter* is also disambiguated. *P* is now the empty set, so SSI stops and its outcome is the list *I* = [*restaurant#1*, *menu#1*, *chain#4*, *waiter#1*, *cafeteria#1*]. For the sake of space we do not describe patterns supporting alternative sense choices (discarded by the algorithm because of their smaller weight). The semantic patterns identified by SSI are shown in Figure 2. For a description of the relation set and pattern grammar, and for an extensive running example, the interested reader can refer to the bibliography.

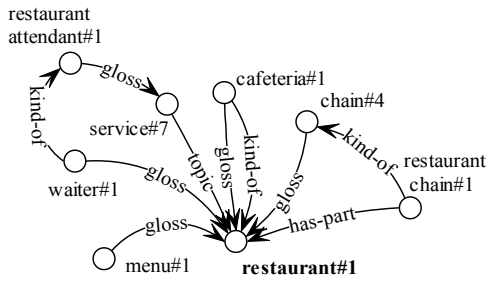

Figure 2. Semantic patterns connecting *restaurant#1* with related concepts.

4. Experiments

For our experiments, we focused on relatedness relations between nominal concepts, leaving the work on other kinds of relations (attribute, quantity, predicate, etc.) and different parts of speech to future experiments.

A disambiguation context is then a set of terms t_1, t_2, …, t_n related to the same concept *r*, i.e. the set of terms contained in the triples (*relatedness, r*, t_1), (*relatedness*, *r*, t_2), …, (*relatedness*, *r*, t_n).

We identified 70 disambiguation contexts of different sizes (one for each selected representative concept), containing a total number of 815 terms to be disambiguated. These terms were manually disambiguated by two annotators, with adjudication in case of disagreement. The application of SSI to such contexts led to a precision result of 85.23% and a recall of 76.44%. The results, reported in Table 2(a), show that both recall and precision measures tend to grow with the context size |*T*|. The intuition for this behaviour is that larger contexts provide richer (and more expressive) semantic interconnections.

Notice that, with respect to other tasks like general-purpose WSD, ontology learning, query expansion, etc., these disambiguation contexts contain terms with stronger interconnections, because collocations express a form of tight semantic relatedness. This explains the high precision results obtained with medium-size or large contexts (around 86.9% on average when $20 \leq |T| \leq 40$).

Table 2. Performances of simple (a) and enriched (b) SSI on different context sizes (|T|).

	|*T*|=5	|*T*|=10	|*T*|=20	|*T*|=30	|*T*|=40
Tot # terms:	175	170	160	150	160
(a) Recall	66.86%	75.29%	78.75%	80.00%	82.52%
Prec.	82.98%	82.58%	86.90%	86.96%	86.84%
(b) Recall	75.43%	82.94%	83.13%	84.00%	88.13%
Prec.	84.08%	83.95%	90.48%	86.30%	89.81%

Then, we enriched the SSI lexical knowledge base with about 10,000 manually disambiguated relatedness relations, and extended the SSI grammar in order to match patterns including relatedness edges (some examples of patterns are shown in Figure 3).

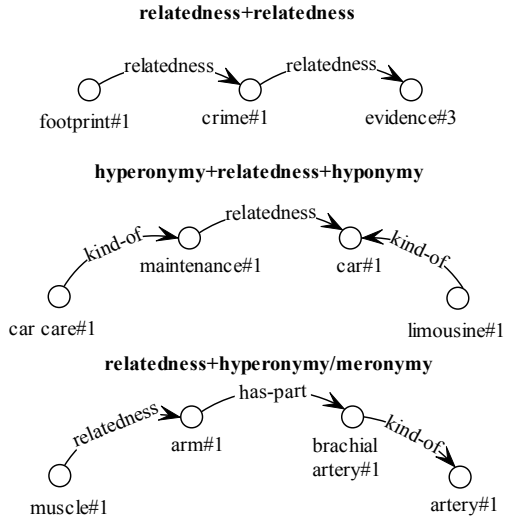

Figure 3. Some examples of relatedness patterns added to the SSI grammar.

In order to measure the improvement obtained on the same task as in Table 2(a), relations connecting concepts in the test set of 70 sets of collocations were excluded (35% over a total number of 10,000 relation instances, about 11 relations per representative concept on average). The total number of "survived" relations actually used in the experiment was then 7,000. Such relations concerned 883 representative concepts.

The second experiment resulted in a significant improvement in terms of recall (82.58% in the average) with respect to the first run, while the increase in precision (86.84%, i.e. about +1.6%) is not striking. Table 2(b) shows that both measures tend to increase with respect to

the first experiment for all context sizes $|T|$ (with a minor, but still significant increase for larger contexts).
The improvement in recall is chiefly motivated by the enrichment of the SSI knowledge base with relatedness relations, thus enabling semantic interconnections between previously unrelated concepts. Such interconnections are transversal in that they overcome the original structure of WordNet, founded on its taxonomical hierarchy and meronymy relations.

5. Conclusions

In this paper we presented a method for the semi-automatic enrichment on top of an existing, large-scale linguistic resource (we adopted WordNet). Building such a richer resource is still a complex task, as a huge number of relations should be extracted and validated. The introduction of representative concepts allows to focus on a restricted, though adequate, number of senses, covering the vast majority of WordNet synsets in terms of taxonomical subsumption.
The application of the SSI WSD algorithm to the disambiguation of collocations is a crucial choice, in that collocation contexts (i.e. the set of terms related to a representative concept) reveal strong semantic interconnections. Furthermore, the enrichment of the pattern grammar with relatedness paths, as well as the extension of the SSI knowledge base with relatedness instances, showed major improvements in disambiguating such contexts.
The definition of a bootstrapping method for extending our experiments to the complete set of representative concepts is ongoing. Following (Yarowsky, 1995) and subsequent works, such a method would select the best seeds to be fed back to the algorithm in order to iteratively increase its initial knowledge. The precision of SSI being very high, residual disambiguation errors could be possibly tolerated.
SSI produces a justification of its sense choices in terms of the detected semantic patterns and their weights, since not all the patterns equally contribute to the choice of a specific sense. This constitutes a significant help for the subsequent manual validation of senses chosen by the automatic procedure. A tool for supporting non-expert annotators in a distributed environment in the complex process of sense selection and validation on a large scale is being developed in our laboratory (Navigli, 2005). Further support is provided by the tool in the form of semantic patterns for the detection of inconsistencies in the knowledge base being extended.
Additional semantic and lexical patterns will also be included to the SSI grammar in order to take into account other kinds of crafted relations (e.g. *attribute*, *property*, *predicate*, etc.).
Finally we plan to apply the SSI algorithm with the improved knowledge base to tasks like *Senseval-3 all words* and *gloss* WSD, ontology learning and query expansion, expecting better performances than those obtained in the original runs.

Acknowledgements

This work is partially funded by the Interop Network of Excellence (508011), 6th European Union FP.

References

Baker, C. F., Fillmore, C. J., and Lowe, J. B. 1998. The Berkeley FrameNet project. In *Proceedings of the COLING-ACL*, Montreal, Canada.
Bunke, H. and Sanfeliu, A. eds. 1990. *Syntactic and Structural pattern Recognition: Theory and Applications* World Scientific, Series in Computer Science vol. 7.
Fellbaum, C. ed. 1998. *WordNet: an Electronic Lexical Database*. MIT Press.
Lea, D. ed. 2002. *Oxford Collocations*. Oxford University Press.
Lenat, D. B. 1995. CYC: A Large-Scale Investment in Knowledge Infrastructure. *Communications of the ACM*, 38(11).
Longman. Ed. 2003 *Longman Language Activator*. Pearson Education.
Kingsbury, P. and Palmer, M. 2002. From Treebank to PropBank. In: *Proceedings of LREC-02*, Las Palmas, Spain.
Kipper, K., Dang, H.T., Palmer, M. 2000. Class-based construction of a verb lexicon. In *Proceedings of AAAI-2000, 17th National Conference on Artificial Intelligence*, Austin, TX.
Mahesh, K. and Nirenburg, S. 1995. A Situated Ontology for Practical NLP. In *Proceedings of IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing*, Montreal, Canada.
Navigli, R. 2005. Supporting Large-Scale Knowledge Acquisition with Structural Semantic Interconnections. *Proceedings of AAAI Spring Symposium 2005 on Knowledge Collection from Volunteer Contributors (KCVC05)*, Stanford University, Palo Alto, California.
Navigli, R. and Velardi, P. 2004. Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites. *Computational Linguistics*, MIT press, (30)2: 151-179.
Yarowsky, D. 1995. Unsupervised Word Sense Disambiguation Rivalling Supervised Methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, 189-196, Cambridge, MA.