

# A Java Framework for Multilingual Definition and Hypernym Extraction

Stefano Faralli and Roberto Navigli

Dipartimento di Informatica

Sapienza Università di Roma

{faralli, navigli}@di.uniroma1.it

## Abstract

In this paper we present a demonstration of a multilingual generalization of Word-Class Lattices (WCLs), a supervised lattice-based model used to identify textual definitions and extract hypernyms from them. Lattices are learned from a dataset of automatically-annotated definitions from Wikipedia. We release a Java API for the programmatic use of multilingual WCLs in three languages (English, French and Italian), as well as a Web application for definition and hypernym extraction from user-provided sentences.

## 1 Introduction

Electronic dictionaries and domain glossaries are definition repositories which prove very useful not only for lookup purposes, but also for automatic tasks such as Question Answering (Cui et al., 2007; Saggion, 2004), taxonomy learning (Navigli et al., 2011; Velardi et al., 2013), domain Word Sense Disambiguation (Duan and Yates, 2010; Faralli and Navigli, 2012), automatic acquisition of semantic predicates (Flati and Navigli, 2013), relation extraction (Yap and Baldwin, 2009) and, more in general, knowledge acquisition (Hovy et al., 2013). Unfortunately, constructing and updating such resources requires the effort of a team of experts. Moreover, they are of no help when dealing with new words or usages, or, even worse, new domains. Nonetheless, raw text often contains several definitional sentences, that is, it provides within itself formal explanations for terms of interest. Whilst it is not feasible to search texts manually for definitions in several languages, the task of extracting definitional information can be automated by means of Machine Learning (ML) and Natural Language Processing (NLP) techniques.

Many approaches (Snow et al., 2004; Kozareva and Hovy, 2010, *inter alia*) build upon lexico-syntactic patterns, inspired by the seminal work of Hearst (1992). However, these methods suffer from two significant drawbacks: on the one hand, low recall (as definitional sentences occur in highly variable syntactic structures), and, on the

other hand, noise (because the most frequent definitional pattern – *X is a Y* – is inherently very noisy). A recent approach to definition and hypernym extraction, called Word-Class Lattices (Navigli and Velardi, 2010, WCLs), overcomes these issues by addressing the variability of definitional sentences and providing a flexible way of automatically extracting hypernyms from them. To do so, lattice-based classifiers are learned from a training set of textual definitions. Training sentences are automatically clustered by similarity and, for each such cluster, a lattice classifier is learned which models the variants of the definition template detected. A lattice is a directed acyclic graph, a subclass of non-deterministic finite state automata. The purpose of the lattice structure is to preserve (in a compact form) the salient differences among distinct sequences.

In this paper we present a demonstration of Word-Class Lattices by providing a Java API and a Web application for online usage. Since multilinguality is a key need in today’s information society, and because WCLs have been tested overwhelmingly only with the English language, we provide experiments for three different languages, namely English, French and Italian. To do so, in contrast to Navigli and Velardi (2010), who created a manually annotated training set of definitions, we provide a heuristic method for the automatic acquisition of reliable training sets from Wikipedia, and use them to determine the robustness and generalization power of WCLs. We show high performance in definition and hypernym extraction for our three languages.

## 2 Word-Class Lattices

In this section we briefly summarize Word-Class Lattices, originally introduced by Navigli and Velardi (2010).

### 2.1 Definitional Sentence Generalization

WCL relies on a formal notion of textual definition. Specifically, given a definition, e.g.: “In computer science, a closure is a first-class function with free variables that are bound in the lexical environment”, we assume that it contains the

[In geography, a **country**]<sub>DF</sub> [is]<sub>VF</sub> [a political *division*]<sub>GF</sub> .  
 [In finance, a **bond**]<sub>DF</sub> [is]<sub>VF</sub> [a negotiable *certificate*]<sub>GF</sub> [that that acknowledges...]<sub>REST</sub> .  
 [In poetry, a **foot**]<sub>DF</sub> [is]<sub>VF</sub> [a *measure*]<sub>GF</sub> [, consisting...]<sub>REST</sub> .

Table 1: Example definitions (defined terms are marked in bold face, their hypernyms in italics).

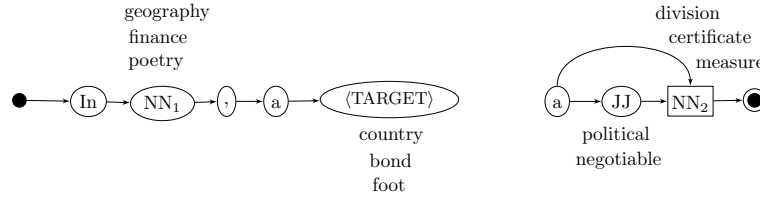


Figure 1: The DF and GF Word-Class Lattices for the sentences in Table 1.

following fields (Storrer and Wellinghoff, 2006): *definiendum* (DF), *definitior* (VF), *definiens* (GF) and *rest* (REST), where DF is the part of the definition including the word being defined (e.g., “In computer science, a closure”), VF is the verb phrase used to introduce the definition (e.g., “is”), GF usually includes the hypernym (e.g., “a first-class *function*”, hypernym marked in italics) and RF includes additional clauses (e.g., “with free variables that are bound in the lexical environment”).

Consider a set of training sentences  $T$ , each of which is automatically part-of-speech tagged and manually bracketed with the DF, VF, GF and REST fields (examples are shown in Table 1). We first identify the set of most frequent words  $F$  (e.g., *the, a, is, of, refer*, etc.). Then we add the symbol  $\langle \text{TARGET} \rangle$  to  $F$  and replace in  $T$  the terms being defined with  $\langle \text{TARGET} \rangle$ . We then use the set of frequent words  $F$  to generalize words to “word classes”.

We define a word class as either a word itself or its part of speech. Given a sentence  $s = w_1, w_2, \dots, w_{|s|}$ , where  $w_i$  is the  $i$ -th word of  $s$ , we generalize its words  $w_i$  to word classes  $\omega_i$  as follows:

$$\omega_i = \begin{cases} w_i & \text{if } w_i \in F \\ POS(w_i) & \text{otherwise} \end{cases}$$

that is, a word  $w_i$  is left unchanged if it occurs frequently in the training corpus (i.e.,  $w_i \in F$ ) or is transformed to its part of speech tag ( $POS(w_i)$ ) otherwise. As a result, we obtain a generalized sentence  $s' = \omega_1, \omega_2, \dots, \omega_{|s|}$ . For instance, given the first sentence in Table 1, we obtain the corresponding generalized sentence: “In NN, a  $\langle \text{TARGET} \rangle$  is a JJ NN”, where NN and JJ indicate the noun and adjective classes, respectively.

## 2.2 Learning

The WCL learning algorithm consists of 3 steps:

- **Star patterns:** each sentence in the training set is preprocessed and generalized to a star

pattern by replacing with  $*$  all the words  $w_i \notin F$ , i.e., non-frequent words. For instance, “In geography, a country is a political division” is transformed to “In  $*$ , a  $\langle \text{TARGET} \rangle$  is a  $*$ ”;

- **Sentence clustering:** the training sentences are then clustered based on the star patterns they belong to;

- **Word-Class Lattice construction:** for each sentence cluster, a WCL is created separately for each DF, VF and GF field by means of a greedy alignment algorithm. In Figure 1 we show the resulting lattices for the DF and GF fields built for the cluster of sentences of Table 1. Note that during the construction of the lattice the nodes associated with the hypernym words in the learning sentences (i.e., *division, certificate* and *measure*) are marked as hypernyms in order to determine the hypernym of a test sentence at classification time (see (Navigli and Velardi, 2010) for details).

## 2.3 Classification

Once the learning process is over, a set of WCLs is produced for the DF, VF and GF fields. Given a test sentence  $s$ , we consider all possible combinations of *definiendum*, *definitior* and *definiens* lattices and select the combination of the three WCLs that best fits the sentence, if such a combination exists. In fact, choosing the most appropriate combination of lattices impacts the performance of hypernym extraction. The best combination of WCLs is selected by maximizing the following confidence score:  $score(s, l_{DF}, l_{VF}, l_{GF}) = coverage \cdot \log(support + 1)$  where  $s$  is the candidate sentence,  $l_{DF}$ ,  $l_{VF}$  and  $l_{GF}$  are three lattices one for each definition field, coverage is the fraction of words of the input sentence covered by the three lattices, and support is the sum of the number of sentences in the star patterns corresponding to the GF lattice. Finally, when a sentence is classified as a definition, its hypernym is extracted by

	# Wikipedia pages	# definitions extracted
English (EN)	3,904,360	1,552,493
French (FR)	1,617,359	447,772
Italian (IT)	1,008,044	291,259

Table 2: The number of Wikipedia pages and the resulting automatically annotated definitions.

selecting the words in the input sentence that are marked as hypernyms in the WCL selected for GF.

### 3 Multilingual Word-Class Lattices

In order to enable multilinguality, thereby extracting definitions and hypernyms in many languages, we provide here a heuristic method for the creation of multilingual training datasets from Wikipedia, that we apply to three languages: English, French and Italian. As a result, we are able to fully automatize the definition and hypernym extraction by utilizing collaboratively-curated encyclopedia content.

#### 3.1 Automatic Learning of Multilingual WCLs

The method consists of four steps:

1. **candidate definition extraction:** we iterate through the collection of Wikipedia pages for the language of interest. For each article we extract the first paragraph, which usually, but not always, contains a definitional sentence for the concept expressed by the page title. We discard all those pages for which the title corresponds to a special page (i.e., title in the form “List of [...]”, “Index of [...]”, “[...] (disambiguation)” etc.).
2. **part-of-speech tagging and phrase chunking:** for each candidate definition we perform part-of-speech tagging and chunking, thus automatically identifying noun, verb, and prepositional phrases (we use TreeTagger (Schmid, 1997)).
3. **automatic annotation:** we replace all the occurrences in the candidate definition of the target term (i.e., the title of the page) with the marker  $\langle \text{TARGET} \rangle$ , we then tag as hypernym the words associated with the first hyperlink occurring to the right of  $\langle \text{TARGET} \rangle$ . Then we tag as VF (i.e., definitor field, see Section 2.1) the verb phrase found between  $\langle \text{TARGET} \rangle$  and the hypernym, if such a phrase exists. Next we tag as GF (i.e., definiens field) the phrase which contains the hypernym and as DF (i.e., definiendum field) the phrase which starts at the beginning of the sentence and ends right before the start of the VP tag. Finally we mark as REST the

remaining phrases after the phrase already tagged as GF. For example, given the sentence “Albert Einstein was a German-born theoretical physicist.”, we produce the following sentence annotation: “[**Albert Einstein**]<sub>DF</sub> [was]<sub>VF</sub> [a German-born theoretical *physicist*]<sub>GF</sub> .” (target term marked in bold and hypernym in italics).

4. **filtering:** we finally discard all the candidate definitions for which not all fields could be found during the previous step (i.e., either the  $\langle \text{TARGET} \rangle$ , hypernym or any DF, VF, GF, REST tag is missing).

We applied the above four steps to the English, French and Italian dumps of Wikipedia<sup>1</sup>. The numbers are shown in Table 2: starting with 3,904,360 Wikipedia pages for English, 1,617,359 for French and 1,008,044 for Italian (first column), we obtained 1,552,493, 447,772, and 291,259 automatically tagged sentences, respectively, for the three languages (second column in the Table). Since we next had to use these sentences for training our WCLs, we took out a random sample of 1000 sentences for each language which we used for testing purposes. We manually annotated each of these sentences as definitional or non-definitional<sup>2</sup> and, in the case of the former, also with the correct hypernym.

#### 3.2 Evaluation

We tested the newly acquired training dataset against two test datasets. The first dataset was our random sampling of 1000 Wikipedia test sentences which we had set aside for each language (no intersection with the training set, see previous section). The second dataset was the same one used in Navigli and Velardi (2010), made up of sentences from the ukWaC Web corpus (Ferraresi et al., 2008) and used to estimate the definition and hypernym extraction performance on an open text corpus.

#### 3.3 Results

Table 3 shows the results obtained on definition (column 2-4) and hypernym extraction (column 5-7) in terms of precision (P), recall (R) and accuracy (A) on our first dataset. Note that accuracy also takes into account candidate definitions in the test set which were tagged as non-definitional (see Section 3.1). In the Table we compare the performance of our English WCL trained from Wikipedia sentences using our automatic procedure against the original performance of WCL

<sup>1</sup>We used the 21-09-2012 (EN), 17-09-2012 (FR), 21-09-2012 (IT) dumps.

<sup>2</sup>Note that the first sentence of a Wikipedia page might seldom be non-definitional, such as “Basmo fortress is located in the north-western part ...”.

	Definition Extraction			Hypernym Extraction		
	P	R	A	P	R	A
EN	98.5	78.3	81.0	98.5	77.4	80.0
FR	98.7	83.3	84.0	98.6	78.0	79.0
IT	98.8	87.3	87.0	98.7	83.2	83.0
EN (2010)	100.0	59.0	66.0	100.0	58.3	65.0

Table 3: Precision (P), recall (R) and accuracy (A) of definition and hypernym extraction when testing on our dataset of 1000 randomly sampled Wikipedia first-paragraph sentences. EN (2010) refers to the WCL learned from the original manually-curated training set from Navigli and Velardi (2010), while EN, FR and IT refer to WCL trained, respectively, with one of the three training sets automatically acquired from Wikipedia.

	P	R
EN	98.9	57.6
EN (2010)	94.8	56.5

Table 4: Estimated WCL definition extraction precision (P) and recall (R), testing a sample of ukWaC sentences.

trained on 1,908 manually-selected training sentences<sup>3</sup>. It can be seen that the automatically acquired training set considerably improves the performance, as it covers higher variability. We note that the recall in both definition and hypernym extraction is higher for French and Italian. We attribute this behavior to the higher complexity and, again, variability of English Wikipedia pages, and specifically first-sentence definitions. We remark that the presented results were obtained without any human effort, except for the independent collaborative editing and hyperlinking of Wikipedia pages, and that the overall performances can be improved by manually checking the automatically annotated training datasets.

We also replicated the experiment carried out by Navigli and Velardi (2010), testing WCLs with a subset (over 300,000 sentences) of the ukWaC Web corpus. As can be seen in Table 4, the estimated precision and recall for WCL definition extraction with the 2010 training set were 94.8% and 56.5%, respectively, while with our automatically acquired English training set we obtained a higher precision of 98.9% and a recall of 57.6%. This second experiment shows that learning WCLs from hundreds of thousands of definition candidates does not overfit to Wikipedia-style definitional sentences.

After looking at the automatically acquired training datasets, we noted some erroneous annotations mainly due to the following factors: i) some Wikipedia pages do not start with defini-

<sup>3</sup>Available from <http://lcl.uniroma1.it/wcl>

```

1 // select the language of interest
2 Language targetLanguage = Language.EN;
3 // open the training set
4 Dataset ts = new AnnotatedDataset(
5     trainingDatasetFile,
6     targetLanguage);
7 // obtain an instance of the WCL classifier
8 WCLClassifier c = new WCLClassifier(targetLanguage);
9 c.train(ts);
10 // create a sentence to be tested
11 Sentence sentence = Sentence.createFromString(
12     "WCL",
13     "WCL_is_a_kind_of_classifier.",
14     targetLanguage);
15 // test the sentence
16 SentenceAnnotation sa = c.test(sentence);
17 // print the hypernym
18 if (sa.isDefinition())
19     System.out.println(sa.getHyper());

```

Figure 2: An example of WCL API usage.

tional sentences; ii) they may contain more than one verbal phrase between the defined term and the hypernym; iii) the first link after the verbal phrase does not cover, or partially covers, the correct hypernym. The elimination of the above wrongly acquired definitional patterns can be implemented with some language-dependent heuristics or can be done by human annotators. In any case, given the presence of a high number of correct annotated sentences, these wrong definitional patterns have a very low impact on the definition and hypernym extraction precision as shown in the above experiments (see Table 3 and Table 4).

## 4 Multilingual WCL API

Together with the training and test sets of the above experiments, we also release here our implementation of Word-Class Lattices, available as a Java API. As a result the WCL classifier can easily be used programmatically in any Java project. In Figure 2 we show an example of the API usage. After the selection of the target language (line 2), we load the training dataset for the target language (line 4). Then an instance of `WCLClassifier` is created (line 8) and the training phase is launched on the input training corpora (line 9). Now the classifier is ready to be tested on any given sentence in the target language (lines 11-16). If the classifier output is positive (line 18) we can print the extracted hypernym (line 19). The output of the presented code is the string “classifier” which corresponds to the hypernym extracted by WCL for the input sentence “WCL is a kind of classifier”.

### 4.1 Web user interface

We also release a Web interface to enable online usage of our WCLs for the English, French and Italian languages. In Figure 3 we show a screenshot of our Web interface. The user can type the

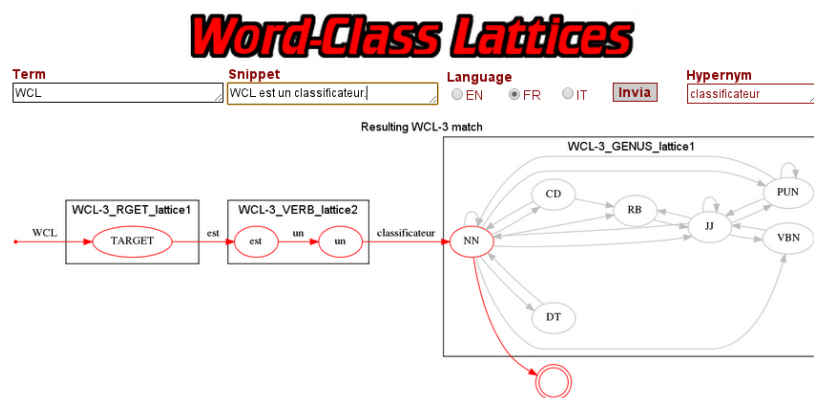


Figure 3: A screenshot of the WCL Web interface.

term of interest, the candidate definition, select the language of interest and, after submission, in the case of positive response from WCL, obtain the corresponding hypernym and a graphical representation of the lattices matching the given sentence, as shown in the bottom part of the Figure.

The graphical representation shows the concatenation of the learned lattices which match the DF, VF, GF parts of the given sentence (see Section 2). We also allow the user not to provide the term of interest: in this case all the nouns in the sentence are considered as candidate defined terms. The Web user interface is part of a client-server application, created with the JavaServer Pages technology. The server side produces an HTML page (like the one shown in Figure 3), using the WCL API (see Section 4) to process and test the submitted definition candidate.

## 5 Related Work

A great deal of work is concerned with the language independent extraction of definitions. Much recent work uses symbolic methods that depend on lexico-syntactic patterns or features, which are manually created or semi-automatically learned as recently done in (Zhang and Jiang, 2009; Westerhout, 2009). A fully automated method is, instead, proposed by Borg et al. (2009), where higher performance (around 60-70% F1-measure) is obtained only for specific domains and patterns. Velardi et al. (2008), in order to improve precision while keeping pattern generality, prune candidates using more refined stylistic patterns and lexical filters. Cui et al. (2007) propose the use of probabilistic lexico-semantic patterns, for definitional question answering in the TREC contest<sup>4</sup>. However, the TREC evaluation datasets cannot be considered true definitions, but rather text fragments providing some relevant fact about a target term.

<sup>4</sup>Text REtrieval Conferences: <http://trec.nist.gov>

Hypernym extraction methods vary from simple lexical patterns (Hearst, 1992; Oakes, 2005) to statistical and machine learning techniques (Agirre et al., 2000; Caraballo, 1999; Dolan et al., 1993; Sanfilippo and Poznanski, 1992; Ritter et al., 2009). Extraction heuristics can be adopted in many languages (De Benedictis et al., 2013), where given a definitional sentence the hypernym is identified as the first occurring noun after the defined term. One of the highest-coverage methods is proposed by Snow et al. (2004). They first search sentences that contain two terms which are known to be in a taxonomic relation (term pairs are taken from WordNet (Miller et al., 1990)); then they parse the sentences, and automatically extract patterns from the parse trees. Finally, they train a hypernym classifier based on these features. Lexico-syntactic patterns are generated for each sentence relating a term to its hypernym, and a dependency parser is used to represent them.

## 6 Conclusion

In this demonstration we provide three main contributions: 1) a general method for obtaining large training sets of annotated definitional sentences for many languages from Wikipedia, thanks to which we can release three new training sets for English, French and Italian; 2) an API to programmatically use WCLs in Java projects; 3) a Web application which enables online use of multilingual WCLs: <http://lcl.uniroma1.it/wcl/>.

## Acknowledgments

The authors gratefully acknowledge the support of the ERC Starting Grant MultiJEDI No. 259234.



## References

- Eneko Agirre, Olatz Ansa, Eduard H. Hovy, and David Martínez. 2000. Enriching very large ontologies using the WWW. In *ECAI Workshop on Ontology Learning*, Berlin, Germany.
- Claudia Borg, Mike Rosner, and Gordon Pace. 2009. Evolutionary algorithms for definition extraction. In *Proceedings of the 1st Workshop on Definition Extraction*, pages 26–32, Borovets, Bulgaria.
- Sharon A. Carballo. 1999. Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the 37<sup>th</sup> Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference*, pages 120–126, Maryland, USA.
- Hang Cui, Min-Yen Kan, and Tat-Seng Chua. 2007. Soft pattern matching models for definitional question answering. *ACM Transactions on Information Systems*, 25(2):1–30.
- Flavio De Benedictis, Stefano Faralli, and Roberto Navigli. 2013. GlossBoot: Bootstrapping Multilingual Domain Glossaries from the Web. In *Proceedings of 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria.
- William Dolan, Lucy Vanderwende, and Stephen D. Richardson. 1993. Automatically deriving structured knowledge bases from on-line dictionaries. In *Proceedings of the First Conference of the Pacific Association for Computational Linguistics*, pages 5–14, Vancouver, Canada.
- Weisi Duan and Alexander Yates. 2010. Extracting glosses to disambiguate word senses. In *Proceedings of Human Language Technologies: The 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 627–635, Los Angeles, CA, USA.
- Stefano Faralli and Roberto Navigli. 2012. A new minimally-supervised framework for Domain Word Sense Disambiguation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1411–1422, Jeju, Korea.
- Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. 2008. Introducing and evaluating ukWaC, a very large web-derived corpus of English. In *Proceedings of the 4th Web as Corpus Workshop (WAC-4)*, pages 47–54, Marrakech, Morocco.
- Tiziano Flati and Roberto Navigli. 2013. SPred: Large-scale Harvesting of Semantic Predicates. In *Proceedings of 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria.
- Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 15th International Conference on Computational Linguistics*, pages 539–545, Nantes, France.
- Eduard Hovy, Roberto Navigli, and Simone Paolo Ponzetto. 2013. Collaboratively built semi-structured content and artificial intelligence: The story so far. *Artificial Intelligence*, 194:2–27.
- Zornitsa Kozareva and Eduard Hovy. 2010. Learning arguments and supertypes of semantic relations using recursive patterns. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL), Uppsala, Sweden*, pages 1482–1491, Uppsala, Sweden.
- George A. Miller, R.T. Beckwith, Christiane D. Fellbaum, D. Gross, and K. Miller. 1990. WordNet: an online lexical database. *International Journal of Lexicography*, 3(4):235–244.
- Roberto Navigli and Paola Velardi. 2010. Learning Word-Class Lattices for definition and hypernym extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1318–1327, Uppsala, Sweden.
- Roberto Navigli, Paola Velardi, and Stefano Faralli. 2011. A graph-based algorithm for inducing lexical taxonomies from scratch. In *Proceedings of the 22th International Joint Conference on Artificial Intelligence*, pages 1872–1877, Barcelona, Spain.
- Michael P. Oakes. 2005. Using Hearst’s rules for the automatic acquisition of hyponyms for mining a pharmaceutical corpus. In *RANLP Text Mining Workshop ’05*, pages 63–67, Borovets, Bulgaria.
- Alan Ritter, Stephen Soderland, and Oren Etzioni. 2009. What is this, anyway: Automatic hypernym discovery. In *Proceedings of the 2009 AAAI Spring Symposium on Learning by Reading and Learning to Read*, pages 88–93, Palo Alto, California.
- Horacio Saggion. 2004. Identifying definitions in text collections for question answering. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation*, pages 1927–1930, Lisbon, Portugal.
- Antonio Sanfilippo and Victor Poznanski. 1992. The acquisition of lexical knowledge from combined machine-readable dictionary sources. In *Proceedings of the third Conference on Applied Natural Language Processing*, pages 80–87, Trento, Italy.
- Helmut Schmid. 1997. Probabilistic part-of-speech tagging using decision trees. In Daniel Jones and Harold Somers, editors, *New Methods in Language Processing*, Studies in Computational Linguistics, pages 154–164. UCL Press, London, GB.
- Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2004. Learning syntactic patterns for automatic hypernym discovery. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Proc. of NIPS 2004*, pages 1297–1304, Cambridge, Mass. MIT Press.
- Angelika Storrer and Sandra Wellinghoff. 2006. Automated detection and annotation of term definitions in German text corpora. In *LREC 2006*, pages 275–295, Genoa, Italy.
- Paola Velardi, Roberto Navigli, and Pierluigi D’Amadio. 2008. Mining the Web to create specialized glossaries. *IEEE Intelligent Systems*, 23(5):18–25.
- Paola Velardi, Stefano Faralli, and Roberto Navigli. 2013. OntoLearn Reloaded: A graph-based algorithm for taxonomy induction. *Computational Linguistics*, 39(3).
- Eline Westerhout. 2009. Definition extraction using linguistic and structural features. In *Proceedings of the RANLP 2009 Workshop on Definition Extraction*, page 61–67, Borovets, Bulgaria.
- Willy Yap and Timothy Baldwin. 2009. Experiments on pattern-based relation learning. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM 2009)*, pages 1657–1660, Hong Kong, China, 2009.
- Chunxia Zhang and Peng Jiang. 2009. Automatic extraction of definitions. In *Proceedings of 2nd IEEE International Conference on Computer Science and Information Technology*, pages 364–368, Beijing, China.