An Annotated Dataset for Extracting Definitions and Hypernyms from the Web

Roberto Navigli, Paola Velardi **Juana Maria Ruiz-Martínez**

Dipartimento di Informatica
Sapienza Università di Roma, Italy
navigli,velardi@di.uniroma1.it

Facultad de Informàtica
Campus de Espinardo, Murcia, Spain
jmruymar@um.es

Abstract

This paper presents and analyzes an annotated corpus of definitions, created to train an algorithm for the automatic extraction of definitions and hypernyms from Web documents. As an additional resource, we also include a corpus of non-definitions with syntactic patterns similar to those of definition sentences, e.g.: "An android is a robot" vs. "Snowcap is unmistakable". Domain and style independence is obtained thanks to the annotation of a sample of the Wikipedia corpus and to a novel pattern generalization algorithm based on word-class lattices (WCL). A lattice is a directed acyclic graph (DAG), a subclass of nondeterministic finite state automata (NFA). The lattice structure has the purpose of preserving the salient differences among distinct sequences, while eliminating redundant information. The WCL algorithm will be integrated into an improved version of the GlossExtractor Web application (Velardi et al., 2008). This paper is mostly concerned with a description of the corpus, the annotation strategy, and a linguistic analysis of the data. A summary of the WCL algorithm is also provided for the sake of completeness.

1 Introduction

This paper presents and analyzes an annotated corpus of definitions, created to train an algorithm for the automatic extraction of definitions and hypernyms from web documents. Domain and style independence is obtained thanks to the annotation of a large and domain-balanced corpus and to a novel pattern generalization algorithm based on word-class lattices (WCL). The WCL algorithm will be integrated into an improved version of the GlossExtractor (Velardi et al., 2008) web application[1]. This paper is mostly concerned with a description of the corpus, the annotation strategy, and a linguistic analysis of the data. The WCL algorithm is described in (Navigli and Velardi, forthcoming) and is summarized here for the sake of clarity.

A great deal of work is concerned with definition extraction in several languages (Gaudio and Branco, 2007; Iftene et al., 2007; Degórski et al., 2008; Przepiórkowski et al., 2007; Storrer and Wellinghoff, 2006; Westerhout and Monachesi, 2007). The majority of these approaches use symbolic methods that depend on lexico-syntactic patterns, which are manually crafted or semi-automatically learned (Zhang and Jiang, 2009; Hovy et al., 2003; Fahmi and Bouma, 2006; Westerhout, 2009). Patterns are either very simple sequences of words (e.g. "refers to", "is defined as", "is a") or more complex sequences of words, parts of speech and chunks. A fully automated method is instead proposed by Borg et al. (2009): they use genetic programming to learn simple features to distinguish between definitions and non-definitions, and they then apply a genetic algorithm to learn individual weights of features. However, rules are learned for only one category of patterns, namely "is" patterns.

Most methods suffer from both low recall and precision, because definitional sentences occur in highly variable and potentially noisy syntactic structures. Higher performance (around 60-70% F_1-measure) is obtained only for specific domains (e.g. an ICT corpus) and patterns (Borg et al., 2009).

Only few papers try to cope with the generality of patterns and domains in real-world corpora (like the web). In the GlossExtractor web-based system (Velardi et al., 2008), to improve precision while keeping pattern generality, candidates are pruned using more refined stylistic patterns and lexical filters. However in its beta-version GlossExtractor relies on a large set of machine-learned, but "fixed" lexico-syntactic patterns. Cui et al. (2007) propose the use of probabilistic patterns, called *soft patterns*, for definitional question answering in the TREC contest[2]. Soft patterns generalize over lexico-syntactic "hard" (fixed) patterns in that they allow a partial matching by calculating a generative degree of match probability between the test instance and the set of training instances. Because of its generalization power, this method is the most closely related with our word-class lattices.

The literature on hypernym extraction offers a higher variability of methods, 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 Poznański, 1992; Ritter et al., 2009). One of the highest-coverage methods is proposed in (Snow et al., 2004). They first search sentences that contain two terms with hyponymous relations (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 classifer 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.

Except for (Snow et al., 2004; Velardi et al., 2008; Cui et al., 2007), all machine-learning methods for definition and hypernym extraction are trained and tested on relatively small domain-specific datasets, in which the variability of patterns is not so high. Annotation of data used for learning is limited to manual classification of positive versus negative examples, with the exception of (Storrer

[1] Already freely available in its beta-version at http://lcl.uniroma1.it/glossextractor.

[2] Text REtrieval Conferences: http://trec.nist.gov

and Wellinghoff, 2006). In this paper, a German corpus of 174 definitions on hypertext research is annotated by marking three fields: the DEFINIENDUM (the term to be defined), the DEFINIENS (meaning postulates for the term) and the DEFINITOR (the verb which relates the definiens component to the definiendum component). In our work, we adopted a similar, but more refined, annotation strategy on a much wider corpus of about 1,700 definitions. The annotated definitions are used to learn a generalized form of word lattices, named Word-Class Lattices (WCL), as an alternative to lexico-syntactic pattern learning. A lattice is a directed acyclic graph (DAG), a subclass of non-deterministic finite state automata (NFA). The lattice structure has the purpose of preserving the salient differences among distinct sequences, while eliminating redundant information.

In computational linguistics, lattices have been used to model in a compact way many sequences of symbols, each representing an alternative hypothesis. In speech processing, phoneme or word lattices (Campbell et al., 2007; Mathias and Byrne, 2006; Collins et al., 2004) are used as an interface between speech recognition and understanding. Lattices are adopted also in Chinese word segmentation (Jiang et al., 2008), decompounding in German (Dyer, 2009), morphologic analysis in Arabic, and to represent classes of translation models in machine translation (Dyer et al., 2008; Schroeder et al., 2009). In more complex text processing tasks, such as information retrieval, information extraction and summarization, the use of word lattices has been postulated but is considered unrealistic because of the dimension of the hypothesis space, which makes it very difficult to cluster in a lattice structure the different patterns.

In definition extraction, the variability of patterns is higher than for "traditional" applications of lattices, such as translation and speech, however not as high as in unconstrained sentences. The methodology that we propose to cluster patterns is based on the use of star (wildcard *) characters to facilitate pattern clustering. The * are then removed and replaced by the original words or by word categories during lattice generation from clusters.

A key feature of our approach is its ability to both identify definitions and extract hypernyms. The method is tested on an annotated corpus and on a large Web corpus, in order to demonstrate the independence of the method from the annotated dataset. WCLs are shown to generalize over lexico-syntactic patterns, and outperform well-known approaches to definition and hypernym extraction from Web documents.

The paper is organized as follows: Section 2 illustrates the corpus annotation strategy, the WCL algorithm is summarised in Section 3 where we also describe the experiments. Finally, a detailed linguistic analysis is performed in Section 4. We conclude the paper in Section 5.

2 Corpus annotation

In our work, we rely on a formal notion of textual definition. Specifically, we assume a definition contains the following fields (Storrer and Wellinghoff, 2006):

- The DEFINIENDUM field (DF): this part of the definition includes the *definiendum* (that is, the word being defined) and its modifiers (e.g., "In computer science, a graph");
- The DEFINITOR field (VF): it includes the verb phrase used to introduce the definition (e.g., "is");
- The DEFINIENS field (GF): it includes the genus phrase (usually including the hypernym, e.g., "a dot");
- The REST field (RF): it includes additional clauses that further specify the *differentia* of the definiendum with respect to its genus (e.g., "that is part of a computer image").

For example, given the sentence:

> *In the history of science, Alchemy (from the Arabic al-kemia) refers to an early form of the investigation of nature combining elements of chemistry, metallurgy, physics, medicine, astrology, semiotics, mysticism, spiritualism, and art all as parts of one greater force*

the following fields are marked:

DF: In the history of science, [Alchemy] (from the Arabic al-kemia)
VF: refers
GF: to an early form of the [investigation] of nature
RF: combining elements of chemistry, metallurgy, physics, medicine, astrology, semiotics, mysticism, spiritualism, and art all as parts of one greater force.

Furthermore, the words *alchemy* and *investigation* are marked, respectively, as "target" and "hypernym". Finally, the part-of-speech analysis of each sentence is also reported, as generated by the TreeTagger system[3].

In annotating our corpus (and in evaluating system's performance on test sets) we followed rather strictly the above notion of definition. For example, a sentence like: "*Etching are made on zinc plates*" is not annotated as a definition, since, while it provides useful information about the term *etching*, it does not include a genus phrase. On the other side, "*noise pollution is a [problem]*", though uninformative, is considered a definition for the opposite reason. Only an analysis of the descriptive power of a detected hypernym could determine the rejection of uninformative definitions: however, one such analysis is mostly context-dependent, since for example "*rucksack is a [problem] in combinatorial optimization*" is indeed an informative definition (and hypernym) for "*rucksack*" in applied mathematics.

Overall, 1,717 definitions have been annotated like in the above example. Terms belong to different Wikipedia categories (en.wikipedia.org/wiki/Wikipedia:Categories) and on-line glossary domains, in order to capture a representative and domain-independent sample of lexical and syntactic patterns for definitions. The associated corpus of negative examples ("syntactically plausible" false definitions) was obtained by automatically extracting sentences from Wikipedia and by manually selecting only false definitions (2,847). An example of false definition in this corpus is:

[3] http://www.ims.uni-stuttgart.de/projekte/.../TreeTagger/

A Pacific Northwest tribe's saga refers to a young woman who put all the self-hatred she felt for her own secretions.

The corpus was annotated by one of the authors, based on a previously agreed annotation policy, and reviewed by the other two authors. When about a 40% of the corpus was annotated, sentences in the remaining 60%, with the same part-of-speech sequence as in the annotated set, were automatically annotated and then reviewed. Finally, survived inconsistencies in annotations have been automatically detected and manually corrected in a third pass. The corpus is available at `http://lcl.uniroma1.it/deco`.

3 Summary of Lattice Learning Algorithm

The annotated corpus was used to learn a variety of definition patterns and compact them into word-class lattices (WCL). The classifier was then tested on a fragment of the annotated corpus and on a subset of the ukWaC Web Corpus (Ferraresi et al., 2008). In this section, we summarize the lattice learning algorithm and analyze its performance.

3.1 Algorithm

The algorithm consists of three steps, described in the next subsections.

3.1.1 Star Patterns

The first step consists of associating a star pattern with each sentence in the training set. Let s be a sentence such that $s = t_1, t_2, \ldots, t_n$, where t_i is its i-th token. Given the set F of most frequent words in the training set, the star pattern $\sigma(s)$ associated with s is obtained by replacing with * all the tokens $t_i \notin F$, that is all the tokens that are non-frequent words. For instance, given the sentence "In arts, a chiaroscuro is a monochrome picture", the corresponding star pattern is "In *, a ⟨TARGET⟩ is a *", where ⟨TARGET⟩ is the defined term.

3.1.2 Sentence Clustering

In the second step, we cluster the sentences in our training set $\mathcal{T}$ based on their star pattern. Formally, let $\Sigma = (\sigma_1, \ldots, \sigma_m)$ be the set of star patterns associated with the sentences in the training set. We create a clustering $\mathcal{C} = (C_1, \ldots, C_m)$ such that $C_i = \{s : \sigma(s) = \sigma_i\}$, that is C_i contains all the sentences whose star pattern is σ_i. We note that each cluster C_i contains sentences whose degree of variability is generally much lower than for any pair of sentences in the training set belonging to two different clusters.

3.1.3 Word-Class Lattice Construction

Finally, the third step consists of the construction of a Word-Class Lattice for each sentence cluster. Given such a cluster $C_i \in \mathcal{C}$, we apply a greedy algorithm that iteratively constructs the WCL.

Let $C_i = \{s_1, s_2, \ldots, s_{|C_i|}\}$ and consider its first sentence $s_1 = t_1, t_2, \ldots, t_n$. Initially, we create a directed graph $G = (V, E)$ such that $V = \{t_1, \ldots, t_n\}$ and $E = \{(t_1, t_2), (t_2, t_3), \ldots, (t_{n-1}, t_n)\}$. We consider the subsequent sentences in C_i one by one and iteratively add them to the graph. To do so, we determine the best alignment between sentence s_j and each sentence $s_k \in C_i$ such that $k < j$. The alignment is calculated based the pairwise similarity of the tokens and parts of speech of the respective sentences. We repeat this calculation for each sentence s_k ($k = 1, \ldots, j-1$) and choose the one that maximizes its alignment score with s_j. We then use the best alignment to add s_j to the graph G: we add to the set of vertices V the tokens of s_j for which there is no alignment to s_k and we add to E the edges $(t_1, t_2), \ldots, (t_{|s_j|-1}, t_{|s_j|})$.

3.1.4 Variants of the WCL Model

So far, we have assumed that our WCL model learns lattices from the training sentences in their entirety (we call this model WCL-1). Note that our model does not take into account the REST field, so this fragment of the training sentences is discarded. We now propose a second model that learns separate WCLs for each field of the definition, namely: DEFINIENDUM (DF), DEFINITOR (VF) and DEFINIENS (GF, cf. Section 2). We refer to this latter model as WCL-3. Rather than applying the WCL algorithm to the entire sentence, the very same method is applied to the sentence fragments tagged with one of the three definition fields. The reason for introducing the WCL-3 model is that, while definitional patterns are highly variable, DF, VF and GF individually exhibit a lower variability, thus WCL-3 should improve the generalization power.

3.1.5 Classification

Once the learning process is over, a set of WCLs is produced. Given a test sentence s, the classification phase for the WCL-1 model consists of determining whether it exists a lattice that matches s. In the case of WCL-3, we consider any combination of DEFINIENDUM, DEFINITOR and DEFINIENS lattices. Given that different combinations might match, for each combination of three WCLs we calculate a confidence score as follows:

$$score(s, l_{\text{DF}}, l_{\text{VF}}, l_{\text{GF}}) = coverage \cdot log(support)$$

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 tokens 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 three lattices.

While WCL-1 is applied as a yes-no classifier as there is a single WCL that can possibly match the input sentence, WCL-3 selects, if any, the combination of the three WCLs that best fits the sentence in terms of coverage and support from the training set.

3.2 Performance evaluation

We conducted experiments on two different datasets:

- Our corpus of 4,564 Wikipedia sentences, that contains 1,717 definitional and 2,847 non-definitional sentences (see Section 2).
- A subset of the ukWaC Web corpus (Ferraresi et al., 2008), a large corpus of the English language constructed by crawling the `.uk` domain of the Web. The subset includes all 313,095 sentences in which occur

Algorithm	P	R	F_1	A
WCL-1	**99.88**	41.82	58.99	75.95
WCL-3	**99.30**	59.85	74.69	**83.24**
Star patterns	94.23	65.15	**77.04**	**83.96**
Bigrams	66.12	**78.77**	71.89	74.56
Random BL	50.00	50.00	50.00	50.00

Table 1: Performance on the Wikipedia dataset.

any of 239 terms randomly selected from the terminology of four different domains (COMPUTER SCIENCE, ASTRONOMY, CARDIOLOGY, AVIATION).

The reason for using the ukWaC corpus is that, unlike the "clean" Wikipedia dataset, in which relatively simple patterns can achieve good results, ukWaC represents a real-world test, with many complex cases (discussed in Section 4).

We experimented with the following systems:

- **WCL-1** and **WCL-3**: these two classifiers are based on our Word-Class Lattice model. WCL-1 learns from the training set a lattice for each cluster of sentences, whereas WCL-3 identifies clusters (and lattices) separately for each sentence field (DEFINIENDUM, DEFINITOR and DEFINIENS) and classifies a sentence as a definition if any combination from the three sets of lattices matches (cf. Section 3.1.4).
- **Star patterns**: a simple classifier based on the patterns learned as a result of step 1 of our WCL learning algorithm (cf. Section 3.1.1): a sentence is classified as a definition if it matches any of the star patterns in the model.
- **Bigrams**: an implementation of the bigram classifier for soft pattern matching proposed by Cui et al. (2007). The classifier selects as definitions all the sentences whose probability is above a specific threshold. The probability is calculated as a mixture of bigram and unigram probabilities, with Laplace smoothing on the latter. We use the very same settings of Cui et al. (2007), including threshold values. While the authors propose a second soft-pattern approach based on Profile HMM, their results do not show significant improvements over the bigram language model.

We calculated precision (the number of definitional sentences correctly retrieved by the system over the number of sentences marked by the system as definitional), recall (the number of definitional sentences correctly retrieved by the system over the number of definitional sentences in the dataset), the F_1-measure (a harmonic mean of precision (P) and recall (R) given by $\frac{2PR}{P+R}$) and accuracy (the number of correctly classified sentences over the total number of sentences in the dataset).

3.3 Results

In Table 1 we report the results of definition extraction systems on the Wikipedia dataset. Given this dataset is also used for training, experiments are performed with 10-fold cross validation. The results show very high precision for WCL-1, WCL-3 (above 99%) and star patterns (94%). As expected, star patterns exhibit a higher recall (65%). The lower recall of WCL-1 is due to its limited ability to generalize compared to WCL-3 and the other methods. In terms of F_1-measure, star patterns achieve 77.04%, and are thus the best system, with WCL-3 ranking second (74.69%). However, if we also account for negative sentences – that is we calculate accuracy – the two systems perform the same (the difference is not statistically significant at $p < 0.01$). The Bigram method achieves the best recall, but precision and F-measure are lower. All the systems perform significantly better than the random baseline.

DF	VF	GF
A ⟨TARGET⟩	is	a *
⟨TARGET⟩	was	of *
The ⟨TARGET⟩	are	an *
A ⟨TARGET⟩ (from *)	were	the *
⟨TARGET⟩ (also known as *)	refers	a * of *
In * , a ⟨TARGET⟩	is a type	the * of *

Table 2: Most frequent patterns for DF, VF and GF.

Algorithm	P	R†
WCL-1	**98.33**	39.39
WCL-3	**94.87**	68.69
Star patterns	44.01	58.59
Bigrams	46.60	45.45
Random BL	50.00	50.00

Table 3: Performance on the ukWaC dataset († Recall is estimated).

From our Wikipedia corpus, we learned 981 lattices (and star patterns). Using WCL-3, we learned 353 DF, 230 VF and 363 GF lattices, that then we used to extract definitions from the ukWaC dataset. Table 2 shows the most frequent patterns for the three fields: DF, VF and GF. An example of combination of different DF, VF and GF lattices is shown in Figure 1.

To calculate precision on the ukWaC dataset, we manually validated the definitions output by each system. However, given the large size of this dataset, recall could only be estimated. To this end, we manually analyzed 50,000 sentences and identified 99 definitions, against which recall was calculated. The results are shown in Table 3. On the ukWaC dataset, WCL-3 performs best, obtaining 94.87% precision and 68.69% recall (we did not calculate F_1, as recall is estimated). Interestingly, star patterns obtain only 44% precision and 58.59% recall. The Bigram method also obtains rather low perfromance. The reason for such bad performance on ukWaC is due to the very different nature of the two datasets: for example, in Wikipedia most "is a" sentences are definitional, whereas this property is not verified in the real world (that is, on the Web, of which ukWaC is a sample).

4 Linguistic Analysis

This section is dedicated to the linguistic analysis of frequent and relevant phenomena emerging from the Wikipedia and ukWaC datasets. This analysis has ben facilitated by the learned lattices, that allow an identification of recurrent as well as idiosyncratic patterns. In previous work on definition analysis, definition classification was mostly manual and led to the identification of few recurrent structures (e.g. 5 types in (Westerhout and Monachesi, 2007)). In the following of this section we report relevant phenomena observed as a result of the analysis of sample definitions and non-definitions.

Ambiguity. In large domains, such as the Web, the level of term ambiguity is larger than what one could imagine. Consider for example the following definitions of "mosaic" and "renaissance", most of which are (commonsensically) unexpected:

Mosaic is a ten week online [course].

Mosaic was the first example of a [web browser].

Mosaic is a representation of medieval [bedford].

Mosaic is the third annual [film festival] sponsored by...

Mosaic is a community [organisation] of black families.

Mosaic is the fourth biggest [retailer] in the uk.

Renaissance is the first central [government investment programme].

Renaissance is an animated cyberpunk/science fiction detective [film].

Renaissance was a humanistic [revival] of classical art.

Renaissance is a very impressive [hotel].

Notice that the method described in section 3 is aimed at extracting definition patterns, not at solving ambiguity[4], a problem known as Word Sense Disambiguation (Navigli, 2009).

Multiple hypernyms. By grouping definitions belonging to the same sense, another phenomenon emerges, which is critical when considering the application of hypernym extraction to taxonomy building: the multiplicity of hypernyms. Consider the following examples:

Renaissance is doubtless the most popular [historical era] portrayed theatrically today.

Renaissance was a humanistic [revival] of classical art, architecture, literature, and learning...

Renaissance was a [period] in European history during which...

The term Renaissance also refers to a [group of artists] who work together...

While the first three hypernyms could be recognized as referring to the notion of *period of time*, it is not easy for an automated system to handle multiple hypernyms in a correct way. This problem has never been discussed in recent literature on taxonomy building based on automatically extracted hypernyms, e.g. (Snow et al., 2006; Yang and Callan, 2009), though we found that it is a very common case.

Uninformative definitions. As we already remarked, there are many sentences which, syntactically, are to be considered definitions (according to the notion of definition stated in Section 2), but are uninformative, e.g.

Artificial Intelligence is a killer [application]...

The mosaic is the main [feature] of the halls large outside patio...

Lithography is an ideal [process] for artists who...

Dada is the [world soul], Dada is the [pawnshop].

A black hole is the ultimate [triumph] of gravity over matter.

This is indeed another frequent problem that is not dealt with by state-of-the-art taxonomy building methods.

Informative non-definitions. Symmetrically, there are sentences correctly rejected on the basis of our notion of definition, which are however informative, though they express other types of relations, e.g.:

*(**author**) Mosaic was developed at the National Center for Supercomputing...*

*(**made of**) Mosaics are made of broken tiles.*

*(**characteristics**) Mosaics are multi-coloured or polychrome...*

*(**characteristics**) A black hole is absolutely invisible.*

*(**made of**) The black holes are formed in the death-throes of a star which...*

Concerning the treatment of these cases, there are different attitudes in the literature, since some of the papers on definition extraction accept as definitions sentences of the above type, e.g. "eLearning comprises resources and application..." in (Westerhout and Monachesi, 2007).

Complex structures. Finally, both during annotation and experiments we identified some particularly complex structure for which the evidence was not strong enough to learn patterns, e.g.:

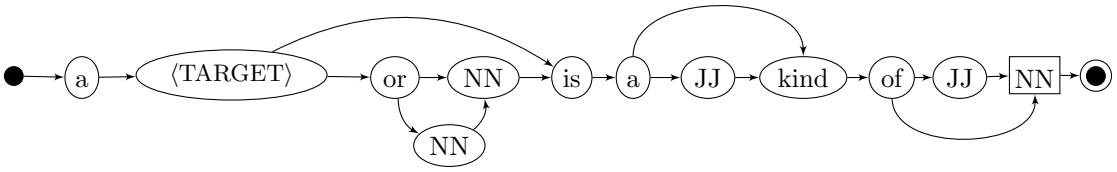
1. **Structures where the information is contained in a relative clause that modifies a noun:**

   A diuretic is something [that makes you urinate more often] e. g. alcohol.

   Artificial Intelligence is a fascinating subject [in which you build intelligent machines and study the nature of mind].

2. **Structures where the hypernym is separated from the verb by a circumlocution:**

[4] In the GlossExtractor system, we describe a domain heuristics (Velardi et al., 2008) to prune non-domain definitions, when given an initial terminology T for which a glossary must be contructed.



A *dial telephone* is a special kind of [telephone]...
A *monomial* or mononomial is a particular kind of [polynomial].
A *bahuvrihi* or bahuvrihi compound is a kind of compound [word].

Figure 1: An example of combined lattice and matching sentences.

Dada is best understood not as a single entity but as a [collection of strands of literature and art] that occurred in zurich, and later paris and berlin, from approximately 1916 to 1924.

The big bang theory is widely considered to be a successful [theory of cosmology], but the theory is incomplete.

We did not add these patterns to the WCL lattices, since first, definitional patterns are open-ended, second, adding some rare pattern might improve recall but reduce precision.

5 Conclusions

In this paper, we have presented a lattice-based approach to definition and hypernym extraction. The novelty of the WCL approach is in the use of a lattice structure to generalize over lexico-syntactic definitional patterns and the ability of the system to jointly identify definitions and extract hypernyms. WCLs achieve high performance as compared with the best-known methods for definition extraction, particularly where the task is more complex, as in real-world documents (i.e. the ukWaC corpus)[5].

The automated clustering of definition structures and the manual identification of the system's hits and errors in the ukWaC corpus allowed a detailed analysis of frequent and relevant phenomena, especially for what concerns hypernym extraction, that have been so far ignored in current literature on automated taxonomy building. In the near future, we plan to apply the results of our methodology to the task of automated taxonomy building, and to test the WCL approach on other information extraction tasks, like hypernym extraction from generic sentence fragments, as in (Snow et al., 2004). We also aim to integrate Hearst's patterns as well as other patterns so as to increase the system's recall in extracting hypernyms not only from definitions, but also from other sentences.

6 References

Eneko Agirre, Ansa Olatz, Xabier Arregi, Xabier Artola, Arantza Daz de Ilarraza Snchez, Mikel Lersundi, David Martnez, Kepa Sarasola, and Ruben Urizar. 2000. Extraction of semantic relations from a basque monolingual dictionary using constraint grammar. In *Proceedings of Euralex*.

Claudia Borg, Mike Rosner, and Gordon Pace. 2009. Evolutionary algorithms for definition extraction. In *Proceedings of the 1st Workshop on Definition Extraction 2009 (wDE'09)*.

William M. Campbell, M. F. Richardson, and D. A. Reynolds. 2007. Language recognition with word lattices and support vector machines. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007)*, Honolulu, HI.

Sharon A. Caraballo. 1999. Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the 37^{th} Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 120–126, Maryland, USA.

Christopher Collins, Bob Carpenter, and Gerald Penn. 2004. Head-driven parsing for word lattices. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 231–238, Barcelona, Spain, July.

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):8.

Łukasz Degórski, Michał Marcinczuk, and Adam Przepiórkowski. 2008. Definition extraction using a sequential combination of baseline grammars and machine learning classifiers. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco.

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.

Christopher Dyer, Smaranda Muresan, and Philip Resnik. 2008. Generalizing word lattice translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2008)*, pages 1012–1020, Columbus, Ohio, USA.

Christopher Dyer. 2009. Using a maximum entropy model to build segmentation lattices for mt. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Associa-*

[5]For more details on comparative performance evaluation, see (Navigli and Velardi, forthcoming).

tion for Computational Linguistics (HLT-NAACL 2009), pages 406–414, Boulder, Colorado, USA.

Ismail Fahmi and Gosse Bouma. 2006. Learning to identify definitions using syntactic features. In *Proceedings of the EACL 2006 workshop on Learning Structured Information in Natural Language Applications*, Trento, Italy.

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)*, Marrakech, Morocco.

Rosa Del Gaudio and António Branco. 2007. Automatic extraction of definitions in portuguese: A rule-based approach. In *Proceedings of the TeMa Workshop*.

Marti Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the* 14^{th} *International Conference on Computational Linguistics (COLING)*, pages 539–545, Nantes, France.

Eduard Hovy, Andrew Philpot, Judith Klavans, Ulrich Germann, and Peter T. Davis. 2003. Extending metadata definitions by automatically extracting and organizing glossary definitions. In *Proceedings of the 2003 Annual National Conference on Digital Government Research*, pages 1–6. Digital Government Society of North America.

Adrian Iftene, Diana Trandabă, and Ionut Pistol. 2007. Natural language processing and knowledge representation for elearning environments. In *Proc. of Applications for Romanian. Proceedings of RANLP workshop*, pages 19–25.

Wenbin Jiang, Haitao Mi, and Qun Liu. 2008. Word lattice reranking for chineseword segmentation and part-of-speech tagging. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008)*, pages 385–392, Manchester, UK.

Lambert Mathias and William Byrne. 2006. Statistical phrase-based speech translation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006)*, Toulouse, France.

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. forthcoming. Learning word-class lattices for definition and hypernym extraction.

Roberto Navigli. 2009. Word sense disambiguation: a survey. *ACM Computing Surveys*, 41(2):1–69.

Michael P. Oakes. 2005. Using hearst's rules for the automatic acquisition of hyponyms for mining a pharmaceutical corpus. In *Proceedings of the Workshop Text Mining Research*.

Adam Przepiórkowski, Lukasz Degórski, Beata Wójtowicz, Miroslav Spousta, Vladislav Kuboň, Kiril Simov, Petya Osenova, and Lothar Lemnitzer. 2007. Towards the automatic extraction of definitions in slavic. In *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing (in ACL '07)*, pages 43–50, Prague, Czech Republic. Association for Computational Linguistics.

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*.

Antonio Sanfilippo and Victor Poznański. 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.

Josh Schroeder, Trevor Cohn, and Philipp Koehn. 2009. Word lattices for multi-source translation. In *Proceedings of the European Chapter of the Association for Computation Linguistics (EACL 2009)*, Athens, Greece.

Rion Snow, Dan Jurafsky, and Andrew Y. Ng. 2004. Learning syntactic patterns for automatic hypernym discovery. In *Proceedings of Advances in Neural Information Processing Systems*.

Rion Snow, Dan Jurafsky, and Andrew Y. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of* 44^{th} *Annual Meeting of the Association for Computational Linguistics joint with* 21^{th} *Conference on Computational Linguistics (COLING-ACL)*, Sydney, Australia.

Angelika Storrer and Sandra Wellinghoff. 2006. Automated detection and annotation of term definitions in german text corpora. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*, Genova, Italy.

Paola Velardi, Roberto Navigli, and Pierluigi D'Amadio. 2008. Mining the web to create specialized glossaries. *IEEE Intelligent Systems*, 23(5):18–25.

Eline Westerhout and Paola Monachesi. 2007. Extraction of dutch definitory contexts for e-learning purposes. In *Proceedings of CLIN*.

Eline Westerhout. 2009. Definition extraction using linguistic and structural features. In *Proceedings of the RANLP 2009 Workshop on Definition Extraction*, pages 61–67.

H. Yang and J. Callan. 2009. A metric-based framework for automatic taxonomy induction. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing*, pages 271–279, Singapore.

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.