Learning Word-Class Lattices for Definition and Hypernym Extraction

Roberto Navigli and **Paola Velardi**
Dipartimento di Informatica
Sapienza Università di Roma
{navigli,velardi}@di.uniroma1.it

Abstract

Definition extraction is the task of automatically identifying definitional sentences within texts. The task has proven useful in many research areas including ontology learning, relation extraction and question answering. However, current approaches – mostly focused on lexico-syntactic patterns – suffer from both low recall and precision, as definitional sentences occur in highly variable syntactic structures. In this paper, we propose Word-Class Lattices (WCLs), a generalization of word lattices that we use to model textual definitions. Lattices are learned from a dataset of definitions from Wikipedia. Our method is applied to the task of definition and hypernym extraction and compares favorably to other pattern generalization methods proposed in the literature.

1 Introduction

Textual definitions constitute a fundamental source to look up when the meaning of a term is sought. Definitions are usually collected in dictionaries and domain glossaries for consultation purposes. However, manually constructing and updating glossaries requires the cooperative effort of a team of domain experts. Further, in the presence of new words or usages, and – even worse – new domains, such resources are of no help. Nonetheless, terms are attested in texts and some (usually few) of the sentences in which a term occurs are typically definitional, that is they provide a formal explanation for the term of interest. While it is not feasible to manually search texts for definitions, this task can be automatized by means of Machine Learning (ML) and Natural Language Processing (NLP) techniques.

Automatic definition extraction is useful not only in the construction of glossaries, but also in many other NLP tasks. In ontology learning, definitions are used to create and enrich concepts with textual information (Gangemi et al., 2003), and extract taxonomic and non-taxonomic relations (Snow et al., 2004; Navigli and Velardi, 2006; Navigli, 2009a). Definitions are also harvested in Question Answering to deal with "what is" questions (Cui et al., 2007; Saggion, 2004). In eLearning, they are used to help students assimilate knowledge (Westerhout and Monachesi, 2007), etc.

Much of the current literature focuses on the use of lexico-syntactic patterns, inspired by Hearst's (1992) seminal work. However, these methods suffer both from low recall and precision, as definitional sentences occur in highly variable syntactic structures, and because the most frequent definitional pattern – X is a Y – is inherently very noisy.

In this paper we propose a generalized form of word lattices, called Word-Class Lattices (WCLs), 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. Lattice-based methods differ in the types of nodes (words, phonemes, concepts), the interpretation of links (representing either a sequential or hierarchical ordering between nodes), their means of creation, and the scoring method used to extract the best consensus output from the lattice (Schroeder et al., 2009). 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. Lat-

Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 1318–1327,
Uppsala, Sweden, 11-16 July 2010.

tices are adopted also in Chinese word segmentation (Jiang et al., 2008), decompounding in German (Dyer, 2009), 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.

To reduce this problem, concept lattices have been proposed (Carpineto and Romano, 2005; Klein, 2008; Zhong et al., 2008). Here links represent hierarchical relations, rather than the sequential order of symbols like in word/phoneme lattices, and nodes are clusters of salient words aggregated using synonymy, similarity, or subtrees of a thesaurus. However, salient word selection and aggregation is non-obvious and furthermore it falls into word sense disambiguation, a notoriously AI-hard problem (Navigli, 2009b).

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 align patterns is based on the use of star (wildcard *) characters to facilitate sentence clustering. Each cluster of sentences is then generalized to a lattice of word classes (each class being either a frequent word or a part of speech). A key feature of our approach is its inherent ability to both identify definitions and extract hypernyms. The method is tested on an annotated corpus of Wikipedia sentences and 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.

The paper is organized as follows: Section 2 discusses related work, WCLs are introduced in Section 3 and illustrated by means of an example in Section 4, experiments are presented in Section 5. We conclude the paper in Section 6.

2 Related Work

Definition Extraction. A great deal of work is concerned with definition extraction in several languages (Klavans and Muresan, 2001; Storrer and Wellinghoff, 2006; Gaudio and Branco, 2007; Iftene et al., 2007; Westerhout and Monachesi, 2007; Przepiórkowski et al., 2007; Degórski et al., 2008). The majority of these approaches use symbolic methods that depend on lexico-syntactic patterns or features, 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 then they apply a genetic algorithm to learn individual weights of features. However, rules are learned for only one category of patterns, namely "is" patterns. As we already remarked, 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. Cui et al. (2007) propose the use of probabilistic lexico-semantic patterns, called *soft patterns*, for definitional question answering in the TREC contest[1]. The authors describe two soft matching models: one is based on an n-gram language model (with the Expectation Maximization algorithm used to estimate the model parameter), the other on Profile Hidden Markov Models (PHMM). Soft patterns generalize over lexico-syntactic "hard" 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. Thanks to its generalization power, this method is the most closely related to our work, however the task of definitional question answering to which it is applied is slightly different from that of definition extraction, so a direct performance comparison is not possi-

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

ble[2]. In fact, the TREC evaluation datasets cannot be considered true definitions, but rather text fragments providing some relevant fact about a target term. For example, sentences like: "Bollywood is a Bombay-based film industry" and "700 or more films produced by India with 200 or more from Bollywood" are both "vital" answers for the question "Bollywood", according to TREC classification, but the second sentence is not a definition.

Hypernym Extraction. 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 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 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.

3 Word-Class Lattices

3.1 Preliminaries

Notion of definition. In our work, we rely 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 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 closure");
- 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 first-class function");
- The REST field (RF): it includes additional clauses that further specify the *differentia* of the definiendum with respect to its genus (e.g., "with free variables that are bound in the lexical environment").

Further examples of definitional sentences annotated with the above fields are shown in Table 1. For each sentence, the definiendum (that is, the word being defined) and its hypernym are marked in bold and italic, respectively. Given the lexico-syntactic nature of the definition extraction models we experiment with, training and test sentences are part-of-speech tagged with the TreeTagger system, a part-of-speech tagger available for many languages (Schmid, 1995).

Word Classes and Generalized Sentences. We now introduce our notion of word class, on which our learning model is based. Let $\mathcal{T}$ be the set of training sentences, manually bracketed with the DF, VF, GF and RF fields. We first determine the set F of words in $\mathcal{T}$ whose frequency is above a threshold θ (e.g., *the*, *a*, *is*, *of*, *refer*, etc.). In our training sentences, we replace the term being defined with ⟨TARGET⟩, thus this frequent token is also included in F.

We 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 $s = w_1, w_2, \ldots, w_{|s|}$, where w_i is the i-th word of s, we generalize its words w_i to word classes ω_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 ($POS(w_i)$) otherwise. As a result, we obtain a generalized sentence $s' = \omega_1, \omega_2, \ldots, \omega_{|s|}$. For instance, given the first sentence in Table 1, we obtain the corresponding generalized sentence: "In NN, a ⟨TARGET⟩ is a JJ NN", where NN and JJ indicate the noun and adjective classes, respectively.

3.2 Algorithm

We now describe our learning algorithm based on Word-Class Lattices. The algorithm consists of three steps:

[2] In the paper, a 55% recall and 34% precision is achieved with the best experiment on TREC-13 data. Furthermore, the classifier of Cui et al. (2007) is based on soft patterns but also on a bag-of-word relevance heuristic. However, the relative influence of the two methods on the final performance is not discussed.

[In arts, a **chiaroscuro**]$_{\text{DF}}$ [is]$_{\text{VF}}$ [a monochrome *picture*]$_{\text{GF}}$.
[In mathematics, a **graph**]$_{\text{DF}}$ [is]$_{\text{VF}}$ [a *data structure*]$_{\text{GF}}$ [that consists of ...]$_{\text{REST}}$.
[In computer science, a **pixel**]$_{\text{DF}}$ [is]$_{\text{VF}}$ [a *dot*]$_{\text{GF}}$ [that is part of a computer image]$_{\text{REST}}$.

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

- **Star patterns:** each sentence in the training set is pre-processed and generalized to a star pattern. For instance, "In arts, a chiaroscuro is a monochrome picture" is transformed to "In *, a ⟨TARGET⟩ is a *" (Section 3.2.1);
- **Sentence clustering:** the training sentences are then clustered based on the star patterns to which they belong (Section 3.2.2);
- **Word-Class Lattice construction:** for each sentence cluster, a WCL is created by means of a greedy alignment algorithm (Section 3.2.3).

We present two variants of our WCL model, dealing either globally with the entire sentence or separately with its definition fields (Section 3.2.4). The WCL models can then be used to classify any input sentence of interest (Section 3.2.5).

3.2.1 Star Patterns

Let $\mathcal{T}$ be the set of training sentences. In this step, we associate a star pattern $\sigma(s)$ with each sentence $s \in \mathcal{T}$. To do so, let $s \in \mathcal{T}$ be a sentence such that $s = w_1, w_2, \ldots, w_{|s|}$, where w_i is its i-th word. Given the set F of most frequent words in $\mathcal{T}$ (cf. Section 3.1), the star pattern $\sigma(s)$ associated with s is obtained by replacing with * all the words $w_i \notin F$, 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.

Note that, here and in what follows, we discard the sentence fragments tagged with the REST field, which is used only to delimit the core part of definitional sentences.

3.2.2 Sentence Clustering

In the second step, we cluster the sentences in our training set $\mathcal{T}$ based on their star patterns. Formally, let $\Sigma = (\sigma_1, \ldots, \sigma_m)$ be the set of star patterns associated with the sentences in $\mathcal{T}$. We create a clustering $\mathcal{C} = (C_1, \ldots, C_m)$ such that $C_i = \{s \in \mathcal{T} : \sigma(s) = \sigma_i\}$, that is C_i contains all the sentences whose star pattern is σ_i.

As an example, assume σ_3 = "In *, a ⟨TARGET⟩ is a *". The sentences reported in Table 1 are all grouped into cluster C_3. We note that each cluster C_i contains sentences whose degree of variability is generally much lower than for any pair of sentences in $\mathcal{T}$ belonging to two different clusters.

3.2.3 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 $s_1 = w_1^1, w_2^1, \ldots, w_{|s_1|}^1$ (w_i^j denotes the i-th token of the j-th sentence). We first produce the corresponding generalized sentence $s'_1 = \omega_1^1, \omega_2^1, \ldots, \omega_{|s_1|}^1$ (cf. Section 3.1). We then create a directed graph $G = (V, E)$ such that $V = \{\omega_1^1, \ldots, \omega_{|s_1|}^1\}$ and $E = \{(\omega_1^1, \omega_2^1), (\omega_2^1, \omega_3^1), \ldots, (\omega_{|s_1|-1}^1, \omega_{|s_1|}^1)\}$. Next, for the subsequent sentences in C_i, that is, for each $j = 2, \ldots, |C_i|$, we determine the alignment between the sentence s_j and each sentence $s_k \in C_i$ such that $k < j$ based on the following dynamic programming formulation (Cormen et al., 1990, pp. 314–319):

$$M_{a,b} = \max \{M_{a-1,b-1} + S_{a,b}, M_{a,b-1}, M_{a-1,b}\}$$

where $a \in \{1, \ldots, |s_k|\}$ and $b \in \{1, \ldots, |s_j|\}$, $S_{a,b}$ is a score of the matching between the a-th token of s_k and the b-th token of s_j, and $M_{0,0}$, $M_{0,b}$ and $M_{a,0}$ are initially set to 0 for all a and b.

The matching score $S_{a,b}$ is calculated on the generalized sentences s'_k of s_k and s'_j of s_j as follows:

$$S_{a,b} = \begin{cases} 1 & \text{if } \omega_a^k = \omega_b^j \\ 0 & \text{otherwise} \end{cases}$$

where ω_a^k and ω_b^j are the a-th and b-th word classes of s'_k and s'_j, respectively. In other words, the matching score equals 1 if the a-th and the b-th tokens of the two original sentences have the same word class.

Finally, the alignment score between s_k and s_j is given by $M_{|s_k|,|s_j|}$, which calculates the mini-

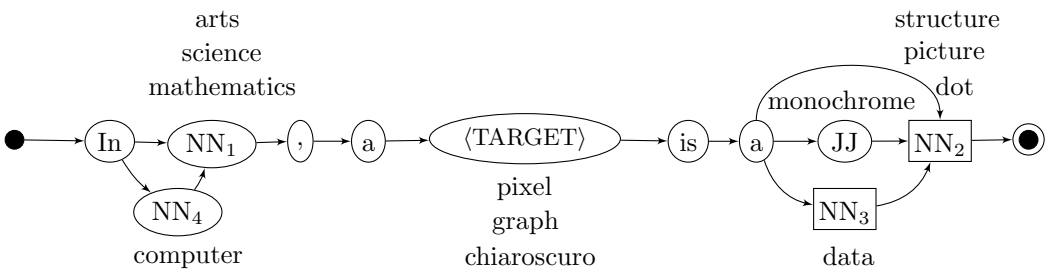

Figure 1: The Word-Class Lattice for the sentences in Table 1. The support of each word class is reported beside the corresponding node.

mal number of misalignments between the two token sequences. 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. Such alignment is obtained by means of backtracking from $M_{|s_k|,|s_j|}$ to $M_{0,0}$. We add to the set of vertices V the tokens of the generalized sentence s'_j for which there is no alignment to s'_k and we add to E the edges (ω^j_1, ω^j_2), $\ldots, (\omega^j_{|s_j|-1}, \omega^j_{|s_j|})$. Furthermore, in the final lattice, nodes associated with the hypernym words in the learning sentences are marked as hypernyms in order to be able to determine the hypernym of a test sentence at classification time.

3.2.4 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). We now propose a second model that learns separate WCLs for each field of the definition, namely: the DEFINIENDUM (DF), DEFINITOR (VF) and DEFINIENS (GF) fields (see Section 3.1). 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.

3.2.5 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. 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 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:

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

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 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 three lattices.

Finally, when a sentence is classified as a definition, its hypernym is extracted by selecting the words in the input sentence that are marked as "hypernyms" in the WCL-1 lattice (or in the WCL-3 GF lattice).

4 Example

As an example, consider the definitions in Table 1. As illustrated in Section 3.2.2, their star pattern is "In *, a ⟨TARGET⟩ is a *". The corresponding WCL is built as follows: the first part-of-speech tagged sentence, "In/IN arts/NN , a/DT ⟨TARGET⟩/NN is/VBZ a/DT monochrome/JJ *picture*/NN", is considered. The corresponding generalized sentence is "In NN , a ⟨TARGET⟩ is a JJ NN". The initially empty graph is thus populated with one node for each word class and one edge for each pair of consecutive tokens, as shown in Figure 1 (the central sequence of nodes in the graph). Note that we draw the hypernym token $\boxed{\text{NN}_2}$ with a rectangle shape. We also add to the

graph a start node • and an end node ⊙, and connect them to the corresponding initial and final sentence tokens. Next, the second sentence, "In mathematics, a graph is a data structure that consists of...", is aligned to the first sentence. The alignment of the generalized sentence is perfect, apart from the $\boxed{\text{NN}_3}$ node corresponding to "data". The node is added to the graph together with the edges a$\rightarrow\boxed{\text{NN}_3}$ and $\boxed{\text{NN}_3}\rightarrow\boxed{\text{NN}_2}$. Finally, the third sentence in Table 1, "In computer science, a pixel is a dot that is part of a computer image", is generalized as "In NN NN , a ⟨TARGET⟩ is a NN". Thus, a new node NN_4 is added, corresponding to "computer" and new edges are added: In$\rightarrow\text{NN}_4$ and $\text{NN}_4\rightarrow\text{NN}_1$. Figure 1 shows the resulting WCL-1 lattice.

5 Experiments

5.1 Experimental Setup

Datasets. We conducted experiments on two different datasets:

- A corpus of 4,619 Wikipedia sentences, that contains 1,908 definitional and 2,711 non-definitional sentences. The former were obtained from a random selection of the first sentences of Wikipedia articles[3]. The defined terms belong to different Wikipedia domain categories[4], so as to capture a representative and cross-domain sample of lexical and syntactic patterns for definitions. These sentences were manually annotated with DEFINIENDUM, DEFINITOR, DEFINIENS and REST fields by an expert annotator, who also marked the hypernyms. The associated set of negative examples ("syntactically plausible" false definitions) was obtained by extracting from the same Wikipedia articles sentences in which the page title occurs.
- 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 over 300,000 sentences in which occur any of 239 terms 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. For example, there are sentences that should be classified as definitional according to Section 3.1 but are rather uninformative, like "**dynamic programming** was the *brainchild* of an american mathematician", as well as informative sentences that are not definitional (e.g., they do not have a hypernym), like "**cubism** was characterised by muted colours and fragmented images". Even more frequently, the dataset includes sentences which are not definitions but have a definitional pattern ("A Pacific Northwest tribe's **saga** refers to a young woman who [..]"), or sentences with very complex definitional patterns ("**white body cells** are the body's clean up *squad*" and "**joule** is also an *expression of electric energy*"). These cases can be correctly handled only with fine-grained patterns. Additional details on the corpus and a more thorough linguistic analysis of complex cases can be found in Navigli et al. (2010).

Systems. For definition extraction, we experiment 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 3.2.4, the best combination is selected).
- **Star patterns**: a simple classifier based on the patterns learned as a result of step 1 of our WCL learning algorithm (cf. Section 3.2.1): 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

[3] The first sentence of Wikipedia entries is, in the large majority of cases, a definition of the page title.

[4] `en.wikipedia.org/wiki/Wikipedia:Categories`

Algorithm	P	R	F_1	A
WCL-1	**99.88**	42.09	59.22	76.06
WCL-3	**98.81**	60.74	**75.23**	**83.48**
Star patterns	86.74	66.14	**75.05**	81.84
Bigrams	66.70	**82.70**	73.84	75.80
Random BL	50.00	50.00	50.00	50.00

Table 2: Performance on the Wikipedia dataset.

Algorithm	P	R†
WCL-1	**98.33**	39.39
WCL-3	**94.87**	56.57
Star patterns	44.01	**63.63**
Bigrams	46.60	45.45
Random BL	50.00	50.00

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

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 (cf. Section 2), their results do not show significant improvements over the bigram language model.

For hypernym extraction, we compared WCL-1 and WCL-3 with **Hearst's patterns**, a system that extracts hypernyms from sentences based on the lexico-syntactic patterns specified in Hearst's seminal work (1992). These include (hypernym in italic): "such *NP* as {NP ,} {(or | and)} NP", "NP {, NP} {,} or other *NP*", "*NP* {,} including { NP ,} {or | and} NP", "*NP* {,} especially { NP ,} {or | and} NP", and variants thereof. However, it should be noted that hypernym extraction methods in the literature do not extract hypernyms from definitional sentences, like we do, but rather from specific patterns like "X such as Y". Therefore a direct comparison with these methods is not possible. Nonetheless, we decided to implement Hearst's patterns for the sake of completeness. We could not replicate the more refined approach by Snow et al. (2004) because it requires the annotation of a possibly very large dataset of sentence fragments. In any case Snow et al. (2004) reported the following performance figures on a corpus of dimension and complexity comparable with ukWaC: the recall-precision graph indicates precision 85% at recall 10% and precision 25% at recall of 30% for the hypernym classifier. A variant of the classifier that includes evidence from coordinate terms (terms with a common ancestor in a taxonomy) obtains an increased precision of 35% at recall 30%. We see no reasons why these figures should vary dramatically on the ukWaC.

Finally, we compare all systems with the **random baseline**, that classifies a sentence as a definition with probability $\frac{1}{2}$.

Measures. To assess the performance of our systems, we calculated the following measures:

- **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 $\frac{2PR}{P+R}$.
- **accuracy** – the number of correctly classified sentences (either as definitional or non-definitional) over the total number of sentences in the dataset.

5.2 Results and Discussion

Definition Extraction. In Table 2 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 (around 99%) and star patterns (86%). As expected, bigrams and star patterns exhibit a higher recall (82% and 66%, respectively). 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 and WCL-3 achieve 75%, and are thus the best systems. Similar performance is observed when we also account for negative sentences – that is we calculate accuracy (with WCL-3 performing better). All the systems perform significantly better than the random baseline.

From our Wikipedia corpus, we learned over 1,000 lattices (and star patterns). Using WCL-3, we learned 381 DF, 252 VF and 395 GF lattices, that then we used to extract definitions from

Algorithm	Full	Substring
WCL-1	**42.75**	77.00
WCL-3	40.73	**78.58**

Table 4: Precision in hypernym extraction on the Wikipedia dataset

Algorithm	Full		Substring	
WCL-1	86.19	(206)	**96.23**	(230)
WCL-3	**89.27**	(383)	**96.27**	(413)
Hearst	65.26	(62)	88.42	(84)

Table 5: Precision in hypernym extraction on the ukWaC dataset (number of hypernyms in parentheses).

the ukWaC dataset. To calculate precision on this dataset, we manually validated the definitions output by each system. However, given the large size of the test set, 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 56.57% recall (we did not calculate F_1, as recall is estimated). Interestingly, star patterns obtain only 44% precision and around 63% recall. Bigrams achieve even lower performance, namely 46.60% precision, 45.45% recall. 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). Also, while WCL does not need any parameter tuning[5], the same does not hold for bigrams[6], whose probability threshold and mixture weights need to be best tuned on the task at hand.

Hypernym Extraction. For hypernym extraction, we tested WCL-1, WCL-3 and Hearst's patterns. Precision results are reported in Tables 4 and 5 for the two datasets, respectively. The Substring column refers to the case in which the captured hypernym is a substring of what the annotator considered to be the correct hypernym. Notice that this is a complex matter, because often the selection of a hypernym depends on semantic and contextual issues. For example, "**Fluoroscopy** is an *imaging method*" and "the **Mosaic** was an interesting *project*" have precisely the same genus pattern, but (probably depending on the vagueness of the noun in the first sentence, and of the adjective in the second) the annotator selected respectively *imaging method* and *project* as hypernyms. For the above reasons it is difficult to achieve high performance in capturing the correct hypernym (e.g. 40.73% with WCL-3 on Wikipedia). However, our performance of identifying a substring of the correct hypernym is much higher (around 78.58%). In Table 4 we do not report the precision of Hearst's patterns, as only one hypernym was found, due to the inherently low coverage of the method.

On the ukWaC dataset, the hypernyms returned by the three systems were manually validated and precision was calculated. Both WCL-1 and WCL-3 obtained a very high precision (86-89% and 96% in identifying the exact hypernym and a substring of it, respectively). Both WCL models are thus equally robust in identifying hypernyms, whereas WCL-1 suffers from a lack of generalization in definition extraction (cf. Tables 2 and 3). Also, given that the ukWaC dataset contains sentences in which any of 239 domain terms occur, WCL-3 extracts on average 1.6 and 1.7 full and substring hypernyms per term, respectively. Hearst's patterns also obtain high precision, especially when substrings are taken into account. However, the number of hypernyms returned by this method is much lower, due to the specificity of the patterns (62 vs. 383 hypernyms returned by WCL-3).

6 Conclusions

In this paper, we have presented a lattice-based approach to definition and hypernym extraction. The novelty of our approach is:

1. The use of a lattice structure to generalize over lexico-syntactic definitional patterns;
2. The ability of the system to jointly identify definitions and extract hypernyms;
3. The generality of the method, which applies to generic Web documents in any domain and style, and needs no parameter tuning;

[5]WCL has only one threshold value θ to be set for determining frequent words (cf. Section 3.1). However, no tuning was made for choosing the best value of θ.

[6]We had to re-tune the system parameters on ukWaC, since with the original settings of Cui et al. (2007) performance was much lower.

4. The high performance as compared with the best-known methods for both definition and hypernym extraction. Our approach outperforms the other systems particularly where the task is more complex, as in real-world documents (i.e., the ukWaC corpus).

Even though definitional patterns are learned from a manually annotated dataset, the dimension and heterogeneity of the training dataset ensures that training needs not to be repeated for specific domains[7], as demonstrated by the cross-domain evaluation on the ukWaC corpus.

The datasets used in our experiments are available from `http://lcl.uniroma1.it/wcl`. We also plan to release our system to the research community. In the near future, we aim to apply the output of our classifiers 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).

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)*, pages 989–992, 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.

Claudio Carpineto and Giovanni Romano. 2005. Using concept lattices for text retrieval and mining. In B. Ganter, G. Stumme, and R. Wille, editors, *Formal Concept Analysis*, pages 161–179.

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.

Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. 1990. *Introduction to algorithms*. the MIT Electrical Engineering and Computer Science Series. MIT Press, Cambridge, MA.

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 Association 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*, pages 64–71, 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.

Aldo Gangemi, Roberto Navigli, and Paola Velardi. 2003. The OntoWordNet project: Extension and axiomatization of conceptual relations in WordNet. In *Proceedings of the International Conference on Ontologies, Databases and Applications of SEmantics (ODBASE 2003)*, pages 820–838, Catania, Italy.

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.

[7] Of course, it would need some additional work if applied to languages other than English. However, the approach does not need to be adapted to the language of interest.

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.

Judith Klavans and Smaranda Muresan. 2001. Evaluation of the DEFINDER system for fully automatic glossary construction. In *Proc. of the American Medical Informatics Association (AMIA) Symposium*.

Michael Tully Klein. 2008. *Understanding English with Lattice-Learning, Master thesis*. MIT, Cambridge, MA, USA.

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. 2006. Ontology enrichment through automatic semantic annotation of on-line glossaries. In *Proceedings of the 15th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2006)*, pages 126–140, Podebrady, Czech Republic.

Roberto Navigli, Paola Velardi, and Juana María Ruiz-Martínez. 2010. An annotated dataset for extracting definitions and hypernyms from the Web. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, Valletta, Malta.

Roberto Navigli. 2009a. Using cycles and quasi-cycles to disambiguate dictionary glosses. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2009)*, pages 594–602, Athens, Greece.

Roberto Navigli. 2009b. 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*, pages 88–93.

Horacio Saggion. 2004. Identifying denitions in text collections for question answering. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*, Lisbon, Portugal.

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.

Helmut Schmid. 1995. Improvements in part-of-speech tagging with an application to german. In *Proceedings of the ACL SIGDAT-Workshop*, pages 47–50.

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)*, pages 719–727, 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*, pages 1297–1304.

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 eLearning 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.

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.

Zhao-man Zhong, Zong-tian Liu, and Yan Guan. 2008. Precise information extraction from text based on two-level concept lattice. In *Proceedings of the 2008 International Symposiums on Information Processing (ISIP '08)*, pages 275–279, Washington, DC, USA.