Extracting glossary sentences from scholarly articles: A comparative evaluation of pattern bootstrapping and deep analysis

Melanie Reiplinger[1] **Ulrich Schäfer**[2] **Magdalena Wolska**[1*]

[1]Computational Linguistics, Saarland University, D-66041 Saarbrücken, Germany
[2]DFKI Language Technology Lab, Campus D 3 1, D-66123 Saarbrücken, Germany
{mreiplin,magda}@coli.uni-saarland.de, ulrich.schaefer@dfki.de

Abstract

The paper reports on a comparative study of two approaches to extracting definitional sentences from a corpus of scholarly discourse: one based on bootstrapping lexico-syntactic patterns and another based on deep analysis. Computational Linguistics was used as the target domain and the ACL Anthology as the corpus. Definitional sentences extracted for a set of well-defined concepts were rated by domain experts. Results show that both methods extract high-quality definition sentences intended for automated glossary construction.

1 Introduction

Specialized glossaries serve two functions: Firstly, they are *linguistic resources* summarizing the terminological basis of a specialized domain. Secondly, they are *knowledge resources*, in that they provide definitions of concepts which the terms denote. Glossaries find obvious use as sources of reference. A survey on the use of lexicographical aids in specialized translation showed that glossaries are among the top five resources used (Durán-Muñoz, 2010). Glossaries have also been shown to facilitate reception of texts and acquisition of knowledge during study (Weiten et al., 1999), while self-explanation of reasoning by referring to definitions has been shown to promote understanding (Aleven et al., 1999). From a machine-processing point of view, glossaries may be used as input for domain ontology induction; see, e.g. (Bozzato et al., 2008).

*Corresponding author

The process of glossary creation is inherently dependent on expert knowledge of the given domain, its concepts and language. In case of scientific domains, which constantly evolve, glossaries need to be regularly maintained: updated and continually extended. Manual creation of specialized glossaries is therefore costly. As an alternative, fully- and semi-automatic methods of glossary creation have been proposed (see Section 2).

This paper compares two approaches to corpus-based extraction of definitional sentences intended to serve as input for a specialized glossary of a scientific domain. The bootstrapping approach acquires lexico-syntactic patterns characteristic of definitions from a corpus of scholarly discourse. The deep approach uses syntactic and semantic processing to build structured representations of sentences based on which 'is-a'-type definitions are extracted. In the present study we used Computational Linguistics (CL) as the target domain of interest and the ACL Anthology as the corpus.

Computational Linguistics, as a specialized domain, is rich in technical terminology. As a cross-disciplinary domain at the intersection of linguistics, computer science, artificial intelligence, and mathematics, it is interesting as far as glossary creation is concerned in that its scholarly discourse ranges from descriptive informal to formal, including mathematical notation. Consider the following two descriptions of *Probabilistic Context-Free Grammar (PCFG)*:

(1) A **PCFG** is a CFG in which each production $A \rightarrow \alpha$ in the grammar's set of productions R is associated with an emission probabil-

Proceedings of the ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries, pages 55–65,
Jeju, Republic of Korea, 10 July 2012.

ity $P(A \rightarrow \alpha)$ that satisfies a normalization constraint

$$\sum_{\alpha:A\rightarrow\alpha\in R} P(A \rightarrow \alpha) = 1$$

and a consistency or tightness constraint [...]

(2) A **PCFG** defines the probability of a string of words as the sum of the probabilities of all admissible phrase structure parses (trees) for that string.

While (1) is an example of a genus-differentia definition, (2) is a valid description of PCFG which neither has the typical copula structure of an "is-a"-type definition, nor does it contain the level of detail of the former. (2) is, however, well-usable for a glossary. The bootstrapping approach extracts definitions of both types. Thus, at the subsequent glossary creation stage, alternative entries can be used to generate glossaries of different granularity and formal detail; e.g., targeting different user groups.

Outline. Section 2 gives an overview of related work. Section 3 presents the corpora and the preprocessing steps. The bootstrapping procedure is summarized in Section 4 and deep analysis in Section 5. Section 6 presents the evaluation methodology and the results. Section 7 presents an outlook.

2 Related Work

Most of the existing definition extraction methods – be it targeting definitional question answering or glossary creation – are based on mining part-of-speech (POS) and/or lexical patterns typical of definitional contexts. Patterns are then filtered heuristically or using machine learning based on features which refer to the contexts' syntax, lexical content, punctuation, layout, position in discourse, etc.

DEFINDER (Muresan and Klavans, 2002), a rule-based system, mines definitions from online medical articles in lay language by extracting sentences using cue-phrases, such as "x is the term for y", "x is defined as y", and punctuation, e.g., hyphens and brackets. The results are analyzed with a statistical parser. Fahmi and Bouma (2006) train supervised learners to classify concept definitions from medical pages of the Dutch Wikipedia using the "is a" pattern and apply a lexical filter (stopwords) to the classifier's output. Besides other features, the position of a sentence within a document is used, which, due to the encyclopaedic text character of the corpus, allows to set the baseline precision at above 75% by classifying the first sentences as definitions. Westerhout and Monachesi (2008) use a complex set of grammar rules over POS, syntactic chunks, and entire definitory contexts to extract definition sentences from an eLearning corpus. Machine learning is used to filter out incorrect candidates. Gaudio and Branco (2009) use only POS information in a supervised-learning approach, pointing out that lexical and syntactic features are domain and language dependent. Borg et al. (2009) use genetic programming to learn rules for typical linguistic forms of definition sentences in an eLearning corpus and genetic algorithms to assign weights to the rules. Velardi et al. (2008) present a fully-automatic end-to-end methodology of glossary creation. First, TermExtractor acquires domain terminology and GlossExtractor searches for definitions on the web using google queries constructed from a set of manually lexical definitional patterns. Then, the search results are filtered using POS and chunk information as well as term distribution properties of the domain of interest. Filtered results are presented to humans for manual validation upon which the system updates the glossary. The entire process is automated.

Bootstrapping as a method of linguistic pattern induction was used for learning hyponymy/is-a relations already in the early 90s by Hearst (1992). Various variants of the procedure – for instance, exploiting POS information, double pattern-anchors, semantic information – have been recently proposed (Etzioni et al., 2005; Pantel and Pennacchiotti, 2006; Girju et al., 2006; Walter, 2008; Kozareva et al., 2008; Wolska et al., 2011). The method presented here is most similar to Hearst's, however, we acquire a large set of general patterns over POS tags alone which we subsequently optimize on a small manually annotated corpus subset by lexicalizing the verb classes.

3 The Corpora and Preprocesssing

The corpora. Three corpora were used in this study. At the initial stage two development corpora were used: a digitalized early draft of the Jurafsky-

Martin textbook (Jurafsky and Martin, 2000) and the WeScience Corpus, a set of Wikipedia articles in the domain of Natural Language Processing (Ytrestøl et al., 2009).[1] The former served as a source of seed domain terms with definitions, while the latter was used for seed pattern creation.

For acquisition of definitional patterns and pattern refinement we used the *ACL Anthology*, a digital archive of scientific papers from conferences, workshops, and journals on Computational Linguistics and Language Technology (Bird et al., 2008).[2] The corpus consisted of 18,653 papers published between 1965 and 2011, with a total of 66,789,624 tokens and 3,288,073 sentences. This corpus was also used to extract sentences for the evaluation using both extraction methods.

Preprocessing. The corpora have been sentence and word-tokenized using regular expression-based sentence boundary detection and tokenization tools. Sentences have been part-of-speech tagged using the TnT tagger (Brants, 2000) trained on the Penn Treebank (Marcus et al., 1993).[3]

Next, domain terms were identified using the C-Value approach (Frantzi et al., 1998). *C-Value* is a domain-independent method of automatic multi-word term recognition that rewards high frequency and high-order n-gram candidates, but penalizes those which frequently occur as sub-strings of another candidate. 10,000 top-ranking multi-word token sequences, according to C-Value, were used.

Domain terms. The set of domain terms was compiled from the following sub-sets: 1) the 10,000 automatically identified multi-word terms, 2) the set of terms appearing on the margins of the Jurafsky-Martin textbook; the intuition being that these are domain-specific terms which are likely to be defined or explained in the text along which they appear, 3) a set of 5,000 terms obtained by expanding frequent abbreviations and acronyms retrieved from the ACL Anthology corpus using simple pattern matching. The token spans of domain terms have been marked in the corpora as these are used in the course of definition pattern acquisition (Section 4.2).

[1] `http://moin.delph-in.net/WeScience`

[2] `http://aclweb.org/anthology/`

[3] The accuracy of tokenization and tagging was not verified.

Seed terms	machine translation		language model
	neural network		reference resolution
	finite(-|)state automaton		hidden markov model
	speech synthesis		semantic role label(l)?ing
	context(-|)free grammar		ontology
	generative grammar		dynamic programming
	mutual information		
Seed patterns		T	.* (is|are|can be) used
		T	.* called
		T	.* (is|are) composed
		T	.* involv(es|ed|e|ing)
		T	.* perform(s|ed|ing)?
		T	\(or .*? \)
	task of .*	T	.*? is
	term	T	.*? refer(s|red|ring)?

Table 1: Seed domain terms (top) and seed patterns (bottom) used for bootstrapping; T stands for a domain term.

4 Bootstrapping Definition Patterns

Bootstrapping-based extraction of definitional sentences proceeds in two stages: First, aiming at recall, a large set of *lexico-syntactic patterns* is acquired: Starting with a small set of seed terms and patterns, term and pattern "pools" are iteratively augmented by searching for matching sentences from the ACL Anthology while acquiring candidates for definition terms and patterns. Second, aiming at precision, general patterns acquired at the first stage are systematically optimized on set of annotated extracted definitions.

4.1 Seed Terms and Seed Patterns

As seed terms to initialize pattern acquisition, we chose terms which are likely to have definitions. Specifically, from the top-ranked multi-word terms, ordered by C-value, we selected those which were also in either the Jurafsky-Martin term list or the list of expanded frequent abbreviations. The resulting 13 seed terms are shown in the top part of Table 1.

Seed definition patterns were created by inspecting definitional contexts in the Jurafsky-Martin and WeScience corpora. First, 12 terms from Jurafsky-Martin and WeScience were selected to find domain terms with which they co-occurred in simple "is-a" patterns. Next, the corpora were searched again to find sentences in which the term pairs in "is-a" relation occur. Non-definition sentences were discarded.

Finally, based on the resulting definition sentences, 22 seed patterns were constructed by transforming the definition phrasings into regular expressions. A subset of the seed phrases extracted in this way is shown in the bottom part of Table 1.[4]

4.2 Acquiring Patterns

Pattern acquisition proceeds in two stages: First, based on seed sets, candidate defining terms are found and ranked. Then, new patterns are acquired by instantiating existing patterns with pairs of likely co-occurring domain terms, searching for sentences in which the term pairs co-occur, and creating POS-based patterns. These steps are summarized below.

Finding definiens candidates. Starting with a set of seed terms and a set of definition phrases, the first stage finds sentences with the seed terms in the `T`-placeholder position of the seed phrases. For each term, the set of extracted sentences is searched for candidate defining terms (other domain terms in the sentence) to form term-term pairs which, at the second stage, will be used to acquire new patterns.

Two situations can occur: a sentence may contain more than one domain term (other than one of the seed terms) or the same domain terms may be found to co-occur with multiple seed terms. Therefore, term-term pairs are ranked.

Ranking. Term-term pairs are ranked using four standard measures of association strength: 1) *co-occurrence* count, 2) *pointwise mutual information (PMI)*, 3) *refined PMI*; compensates bias toward low-frequency events by multiplying PMI with co-occurrence count (Manning and Schütze, 1999), and 4) *mutual dependency (MD)*; compensates bias toward rare events by subtracting co-occurrence's self-information (entropy) from its PMI (Thanopoulos et al., 2002). The measures are calculated based on the corpus for co-occurrences within a 15-word window.

Based on experimentation, mutual dependency was found to produce the best results and therefore it was used in ranking definiens candidates in the final evaluation of patterns. The top-k candidates make up the set of defining terms to be used in the pattern acquisition stage. Table 2 shows the top-20 candidate defining terms for the term "Lexical Functional Grammar", according to mutual dependency.

Domain term	Candidate defining terms
lexical functional grammar (LFG)	phrase structure grammar formal system functional unification grammar grammatical representation phrase structure generalized phrase functional unification binding theory syntactic theories functional structure grammar formalism(s) grammars linguistic theor(y|ies)

Table 2: Candidate defining phrases of the term "Lexical Functional Grammar (LFG)".

Pattern and domain term acquisition. At the pattern acquisition stage, definition patterns are retrieved by 1) coupling terms with their definiens candidates, 2) extracting sentences that contain the pair within the specified window of words, and finally 3) creating POS-based patterns corresponding to the extracted sentences. All co-occurrences of each term together with each of its defining terms within the fixed window size are extracted from the POS-tagged corpus. At each iteration also new definiendum and definiens terms are found by applying the new abstracted patterns to the corpus and retrieving the matching domain terms.

The newly extracted sentences and terms are filtered (see "Filtering" below). The remaining data constitute new instances for further iterations. The linguistic material between the two terms in the extracted sentences is taken to be an instantiation of a potential definition pattern for which its POS pattern is created as follows:

- The defined and defining terms are replaced by placeholders, `T` and `DEF`,
- All the material outside the `T` and `DEF` anchors is removed; i.e. the resulting patterns have the form `'T ... DEF'` or `'DEF ... T'`
- Assuming that the fundamental elements of a definition pattern, are verbs and noun phrases,

[4] Here and further in the paper, regular expressions are presented in Perl notation.

all tags except verb, noun, modal and the infinitive marker "to" are replaced with by placeholders denoting any string; punctuation is preserved, as it has been observed to be informative in detecting definitions (Westerhout and Monachesi, 2008; Fahmi and Bouma, 2006),

- Sequences of singular and plural nouns and proper nouns are replaced with noun phrase placeholder, `NP`; it is expanded to match complex noun phrases when applying the patterns to extract definition sentences.

The new patterns and terms are then fed as input to the acquisition process to extract more sentences and again abstract new patterns.

Filtering. In the course of pattern acquisition information on term-pattern co-occurrence frequencies is stored and relative frequencies are calculated: 1) for each term, the percentage of seed patterns it occurs with, and 2) for each pattern, the percentage of seed terms it occurs with. These are used in the bootstrapping cycles to filter out terms which do not occur as part of a sufficient number of patterns (possibly false positive definiendum candidates) and patterns which do not occur with sufficient number of terms (insufficient generalizing behavior).

Moreover, the following filtering rules are applied: Abstracted POS-pattern sequences of the form `'T .+ DEF'`[5] and `'DEF T'` are discarded; the former because it is not informative, the latter because it is rather an indicator of compound nouns than of definitions. From the extracted sentences, those containing negation are filtered out; negation is contra-indicative of definition (Pearson, 1996). For the same reason, auxiliary constructions with "do" and "have" are excluded unless, in case of the latter, "have" is followed by a two past participle tags as in, e.g., "has been/VBN defined/VBN (as)."

4.3 Manual Refinement

While the goal of the bootstrapping stage was to find as many candidate patterns for good definition terms as possible, the purpose of the refinement stage is to aim at precision. Since the automatically extracted patterns consist only of verb and noun phrase tags between the definiendum and its defining term anchors, they are too general.

[5] '.+' stands for at least one arbitrary character.

#	Definitions	#	Non-definitions
25	is/VBZ	24	is/VBZ
8	represents/VBZ	14	contains/VBZ
6	provides/VBZ	9	employed/VBD
6	contains/VBZ	6	includes/VBZ
6	consists/VBZ	4	reflects/VBZ
3	serves/VBZ	3	uses/VBZ
3	describes/VBZ	3	typed/VBN
3	constitutes/VBZ	3	provides/VBZ
3	are/VBP	3	learning/VBG

Table 3: Subset of verbs occurring in sentences matched by the most frequently extracted patterns.

In order to create more precise patterns, we tuned the pattern sequences based on a small development sub-corpus of the extracted sentences which we annotated. The development corpus was created by extracting sentences using the most frequent patterns instantiated with the term which occurred with the highest percentage of seed patterns. The term "ontology" appeared with more than 80% of the patterns and was used for this purpose. The sentences were then manually annotated as to whether they are true-positive or false examples of definitions (101 and 163 sentences, respectively).

Pattern tuning was done by investigating which verbs are and which are not indicative of definitions based on the positive and negative example sentences. Table 3 shows the frequency distribution of verbs (or verb sequences) in the annotated corpus which occurred more than twice. Abstracting over POS sequences of the sentences containing definition-indicative verbs, we created 13 patterns, extending the automatically found patterns, that yielded 65% precision on the development set, matching 51% of the definition sentences, and reducing noise to 17% non-definitions. Patterns resulting from verb tuning were used in the evaluation. Examples of the tuned patterns are shown below:

```
T VBZ DT JJ? NP .* DEF
T , NP VBZ IN NP .* DEF
T , .+ VBZ DT .+ NP .* DEF
T VBZ DT JJ? NP .* DEF
```

The first pattern matches our both introductory

example definitions of the term "PCFG" (cf. Section 1) with `'T'` as a placeholder for the term itself, `'NP'` denoting a noun phrase, and `'DEF'` one of the term's defining phrases, in the first case, (1), "grammar", in the second case, (2), "probabilities". The examples annotated with matched pattern elements are shown below:[6]

> [PCFG]$_{\texttt{T}}$ [is]$_{\texttt{VBZ}}$ [a]$_{\texttt{DT}}$ [CFG]$_{\texttt{NP}}$ [in which each production $A \rightarrow \alpha$ in the]$_{*}$ [grammar]$_{\texttt{DEF}}$'s set of productions R is associated with an emission probability ...
>
> A [PCFG]$_{\texttt{T}}$ [defines]$_{\texttt{VBZ}}$ [the]$_{\texttt{DT}}$ [probability]$_{\texttt{DEF}}$ of a string of words as the sum of the probabilities of all admissible phrase structure parses (trees) for that string.

5 Deep Analysis for Definition Extraction

An alternative, largely domain-independent approach to the extraction of definition sentences is based on the sentence-semantic index generation of the ACL Anthology Searchbench (Schäfer et al., 2011).

Deep syntactic parsing with semantic predicate-argument structure extraction of each of the approx. 3.3 million sentences in the 18,653 papers ACL Anthology corpus is used for our experiments. We briefly describe how in this approach we get from the sentence text to the semantic representation.

The preprocessing is shared with the bootstrapping-based approach for definition sentence extraction, namely PDF-to-text extraction, sentence boundary detection (SBR), and trigram-based POS tagging with TnT (Brants, 2000). The tagger output is combined with information from a named entity recognizer that in addition delivers hypothetical information on citation expressions. The combined result is delivered as input to the deep parser PET (Callmeier, 2000) running the open source HPSG grammar (Pollard and Sag, 1994) grammar for English (ERG; Flickinger (2002)).

The deep parser is made robust and fast through a careful combination of several techniques; e.g.: (1) *chart pruning*: directed search during parsing to increase performance and coverage for longer sentences (Cramer and Zhang, 2010); (2) *chart mapping*: a framework for integrating preprocessing information from PoS tagger and named entity recognizer in exactly the way the deep grammar expects it (Adolphs et al., 2008)[7]; (3) a statistical parse ranking model (WeScience; (Flickinger et al., 2010)).

The parser outputs sentence-semantic representation in the MRS format (Copestake et al., 2005) that is transformed into a dependency-like variant (Copestake, 2009). From these DMRS representations, predicate-argument structures are derived. These are indexed with structure (semantic subject, predicate, direct object, indirect object, adjuncts) using a customized Apache Solr[8] server. Matching of arguments is left to Solr's standard analyzer for English with stemming; exact matches are ranked higher than partial matches.

The basic semantics extraction algorithm consists of the following steps: 1) calculate the closure for each (D)MRS elementary predication based on the EQ (variable equivalence) relation and group the predicates and entities in each closure respectively; 2) extract the relations of the groups, which results in a graph as a whole; 3) recursively traverse the graph, form one semantic tuple for each predicate, and fill information under its scope, i.e. subject, object, etc.

The semantic structure extraction algorithm generates multiple predicate-argument structures for coordinated sentence (sub-)structures in the index. Moreover, explicit negation is recognized and negated sentences are excluded for the task for the same reasons as in the bootstrapping approach above (see Section 4.2, "Filtering").

Further details of the deep parsing approach are described in (Schäfer and Kiefer, 2011). In the Searchbench online system[9], the definition extraction can by tested with any domain term `T` by using statement queries of the form '**s:T p:is**'.

6 Evaluation

For evaluation, we selected 20 terms, shown in Table 4, which can be considered *domain terms* in the

[6] Matching pattern elements in square brackets; tags from the pattern subscripted.

[7] PoS tagging, e.g., helps the deep parser to cope with words unknown to the deep lexicon, for which default entries based on the PoS information are generated on the fly.

[8] `http://lucene.apache.org/solr`

[9] `http://aclasb.dfki.de`

integer linear programming (ILP)
conditional random field (CRF)
support vector machine (SVM)
latent semantic analysis (LSA)
combinatory categorial grammar (CCG)
lexical-functional grammar (LFG)
probabilistic context-free grammar (PCFG)
discourse representation theory (DRT)
discourse representation structure (DRS)
phrase-based machine translation (PSMT;PBSMT)
statistical machine translation (SMT)
multi-document summarization (MDS)
word sense disambiguation (WSD)
semantic role labeling (SRL)
coreference resolution
conditional entropy
cosine similarity
mutual information (MI)
default unification (DU)
computational linguistics (CL)

Table 4: Domain-terms used in the rating experiment

domain of computational linguistics. Five general terms, such as 'English text' or 'web page', were also included in the evaluation as a control sample; since general terms of this kind are not likely to be defined in scientific papers in CL, their definition sentences were of low quality (false positives). We do not include them in the summary of the evaluation results for space reasons. "Computational linguistics", while certainly a domain term in the domain, is not likely to be defined in the articles in the ACL Anthology, however, the term as such should rather be included in a glossary of computational linguistics, therefore, we included it in the evaluation.

Due to the lack of a gold-standard glossary definitions in the domain, we performed a rating experiment in which we asked domain experts to judge top-ranked definitional sentences extracted using the two approaches. Below we briefly outline the evaluation setup and the procedure.

6.1 Evaluation Data

A set of definitional sentences for the 20 domain terms was extracted as follows:

Lexico-syntactic patterns (LSP). For the lexico-syntactic patterns approach, sentences extracted by the set of refined patterns (see Section 4.3) were considered for evaluation only if they contained at least one of the term's potential defining phrases as identified by the first stage of the glossary extraction (Section 4.2). Acronyms were allowed as fillers of the domain term placeholders.

The candidate evaluation sentences were ranked using single linkage clustering in order to find subsets of similar sentences. `tf.idf`-based cosine between vectors of lemmatized words was used as a similarity function. As in (Shen et al., 2006), the longest sentence was chosen from each of the clusters. Results were ranked by considering the size of the clusters as a measure of how likely it represents a definition. The larger the cluster, the higher it was ranked. Five top-ranked sentences for each of the 20 terms were used for the evaluation.

Deep analysis (DA). The only pattern used for deep analysis extraction was 'subject:`T` predicate:**is**', with '**is**' restricted by the HPSG grammar to be the copula relation and not an auxiliary such as in passive constructions, etc. Five top-ranked sentences – as per the Solr's matching algorithm – extracted with this pattern were used for the evaluation.

In total, 200 candidate definition sentences for 20 domain terms were evaluated, 100 per extraction methods. Examples of candidate glossary sentences extracted using both methods, along with their ratings, are shown in the appendix.

6.2 Evaluation Method

Candidate definition sentences were presented to 6 human domain experts by a web interface displaying one sentence at a time in random order. Judges were asked to rate sentences on a 5-point ordinal scale with the following descriptors:[10]

5: The passage provides a precise and concise description of the concept

4: The passage provides a good description of the concept

3: The passage provides useful information about the concept, which could enhance a definition

[10]Example definitions at each scale point selected by the authors were shown for the concept "hidden markov model".

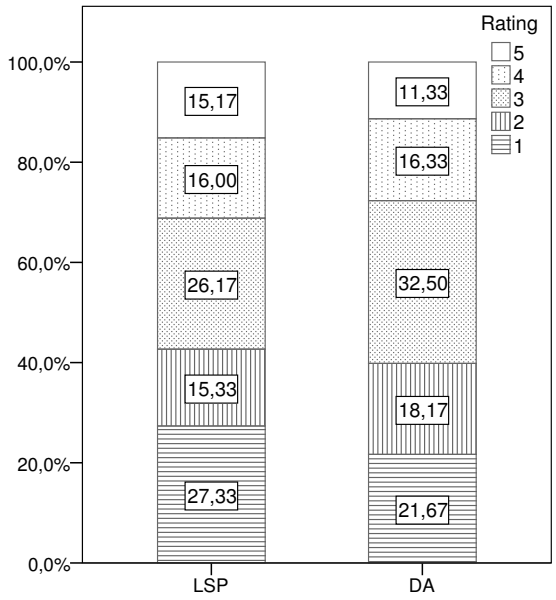

Figure 1: Distribution of ratings across the 5 scale points; LSP: lexico-syntactic patterns, DA: deep analysis

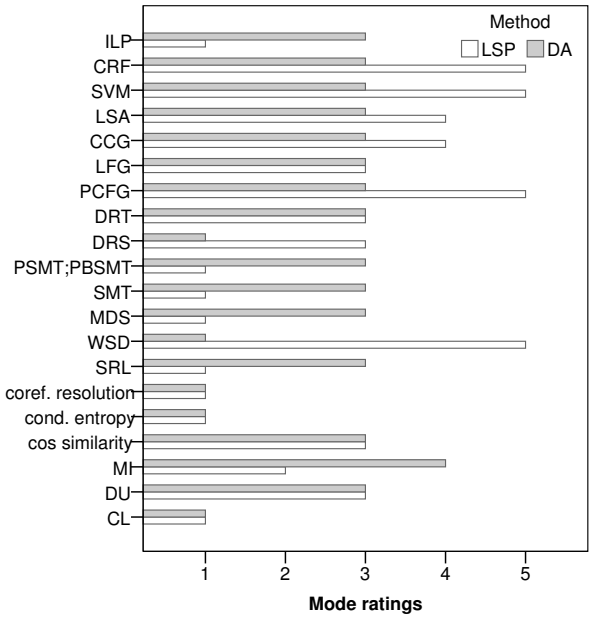

Figure 2: Mode values of ratings per method for the individual domain terms; see Table 4

2: The passage is not a good enough description of the concept to serve as a definition; for instance, it's too general, unfocused, or a subconcept/superconcept of the target concept is defined instead

1: The passage does not describe the concept at all

The judges participating in the rating experiment were PhD students, postdoctoral researchers, or researchers of comparable expertise, active in the areas of computational linguistics/natural language processing/language technology. One of the raters was one of the authors of this paper. The raters were explicitly instructed to think along the lines of "what they would like to see in a glossary of computational linguistics terms".

6.3 Results

Figure 1 shows the distribution of ratings across the five scale points for the two systems. Around 57% of the LSP ratings and 60% of DA ratings fall within the top three scale-points (positive ratings) and 43% and 40%, respectively, within the bottom two scale-points (low ratings). Krippendorff's ordinal α (Hayes and Krippendorff, 2007) was 0.66 (1,000 bootstrapped samples) indicating a modest degree of agreement, at which, however, tentative conclusions can be drawn.

Figure 2 shows the distribution of mode ratings of the individual domain terms used in the evaluation. Definitions of 6 terms extracted using the LSP method were rated most frequently at 4 or 5 as opposed to the majority of ratings at 3 for most terms in case of the DA method.

A Wilcoxon signed-rank test was conducted to evaluate whether domain experts favored definitional sentences extracted by one the two methods.[11] The results indicated no significant difference between ratings of definitions extracted using LSP and DA ($Z = 0.43$, $p = 0.68$).

Now, considering that the ultimate purpose of the sentence extraction is glossary creation, we were also interested in how the top-ranked sentences were rated; that is, assuming we were to create a glossary using only the highest ranked sentences (according to the methods' ranking schemes; see Section 6.1) we wanted to know whether one of the methods proposes rank-1 candidates with higher ratings, independently of the magnitude of the difference. A sign test indicated no statistical difference in ratings of the rank-1 candidates between the two methods.

[11] Definition sentences for each domain term were paired by their rank assigned by the extraction methods: rank-1 DA sentence with rank-1 LSP, etc.; see Section 6.1.

7 Conclusions and Future Work

The results show that both methods have the potential of extracting good quality glossary sentences: the majority of the extracted sentences provide at least useful information about the domain concepts. However, both methods need improvement.

The rating experiment suggests that the concept of definition quality in a specialized domain is largely subjective (borderline acceptable agreement overall and $\alpha = 0.65$ for rank-1 sentences). This calls for a modification of the evaluation methodology and for additional tests of consistency of ratings. The low agreement might be remedied by introducing a blocked design in which groups of judges would evaluate definitions of a small set of concepts with which they are most familiar, rather than a large set of concepts from various CL sub-areas.

An analysis of the extracted sentences and their ratings[12] revealed that deep analysis reduces noise in sentence extraction. Bootstrapping, however, yields more candidate sentences with good or very good ratings. While in the present work pattern refinement was based only on verbs, we observed that also the presence and position of (wh-)determiners and prepositions might be informative. Further experiments are needed 1) to find out how much specificity can be allowed without blocking the patterns' productivity and 2) to exploit the complementary strengths of the methods by combining them.

Since both approaches use generic linguistic resources and preprocessing (POS-tagging, named-entity extraction, etc.) they can be considered domain-independent. To our knowledge, this is, however, the first work that attempts to identify definitions of Computational Linguistics concepts. Thus, it contributes to evaluating pattern bootstrapping and deep analysis in the context of the definition extraction task in our own domain.

Acknowledgments

The C-Value algorithm was implemented by Mihai Grigore. We are indebted to our colleagues from the Computational Linguistics department and DFKI in Saarbrücken who kindly agreed to participate in the rating experiment as domain experts. We are also grateful to the reviewers for their feedback. The work described in this paper has been partially funded by the German Federal Ministry of Education and Research, projects TAKE (FKZ 01IW08003) and Deependance (FKZ 01IW11003).

[12] Not included in this paper for space reasons

References

P. Adolphs, S. Oepen, U. Callmeier, B. Crysmann, D. Flickinger, and B. Kiefer. 2008. Some Fine Points of Hybrid Natural Language Parsing. In *Proceedings of the 6th LREC*, pages 1380–1387.

V. Aleven, K. R. Koedinger, and K. Cross. 1999. Tutoring Answer Explanation Fosters Learning with Understanding. In *Artificial Intelligence in Education*, pages 199–206. IOS Press.

S. Bird, R. Dale, B. Dorr, B. Gibson, M. Joseph, M.-Y. Kan, D. Lee, B. Powley, D. Radev, and Y. F. Tan. 2008. The ACL Anthology Reference Corpus: A Reference Dataset for Bibliographic Research in Computational Linguistics. In *Proceedings of the 6th LREC*, pages 1755–1759.

C. Borg, M. Rosner, and G. Pace. 2009. Evolutionary Algorithms for Definition Extraction. In *Proceedings of the 1st Workshop on Definition Extraction*, pages 26–32.

L. Bozzato, M. Ferrari, and A. Trombetta. 2008. Building a Domain Ontology from Glossaries: A General Methodology. In *Proceedings of the 5th Workshop on Semantic Web Applications and Perspectives*, pages 1–10.

T. Brants. 2000. TnT – a statistical part-of-speech tagger. In *Proceedings of ANLP*, pages 224–231.

U. Callmeier. 2000. PET – A Platform for Experimentation with Efficient HPSG Processing Techniques. *Natural Language Engineering*, 6(1):99–108.

A. Copestake, D. Flickinger, I. A. Sag, and C. Pollard. 2005. Minimal Recursion Semantics: an Introduction. *Research on Language and Computation*, 3(2–3):281–332.

A. Copestake. 2009. Slacker semantics: why superficiality, dependency and avoidance of commitment can be the right way to go. In *Proceedings of the 12th EACL Conference*, pages 1–9.

B. Cramer and Y. Zhang. 2010. Constraining robust constructions for broad-coverage parsing with precision grammars. In *Proceedings of the 23rd COLING Conference*, pages 223–231.

I. Durán-Muñoz, 2010. *eLexicography in the 21st century: New challenges, new applications*, volume 7, chapter Specialised lexicographical resources: a survey of translators' needs, pages 55–66. Presses Universitaires de Louvain.

O. Etzioni, M. Cafarella, D. Downey, A-M. Popescu, T. Shaked, S. Soderland, D.S. Weld, and A. Yates. 2005. Unsupervised named-entity extraction from the Web: an experimental study. *Artificial Intelligence*, 165:91–134.

I. Fahmi and G. Bouma. 2006. Learning to identify definitions using syntactic features. In *Proceedings of the EACL Workshop on Learning Structured Information in Natural Language Applications*, pages 64–71.

D. Flickinger, S. Oepen, and G. Ytrestøl. 2010. WikiWoods: Syntacto-semantic annotation for English Wikipedia. In *Proceedings of the 7th LREC*, pages 1665–1671.

D. Flickinger. 2002. On building a more efficient grammar by exploiting types. In *Collaborative Language Engineering. A Case Study in Efficient Grammar-based Processing*, pages 1–17. CSLI Publications, Stanford, CA.

K. Frantzi, S. Ananiadou, and H. Mima. 1998. Automatic recognition of multi-word terms: the C-value/NC-value method. In *Proceedings of the 2nd European Conference on Research and Advanced Technology for Digital Libraries*, pages 585–604.

R. Del Gaudio and A. Branco. 2009. Language independent system for definition extraction: First results using learning algorithms. In *Proceedings of the 1st Workshop on Definition Extraction*, pages 33–39.

R. Girju, A. Badulescu, and D. Moldovan. 2006. Automatic discovery of part-whole relations. *Computational Linguistics*, 32(1):83–135.

A. F. Hayes and K. Krippendorff. 2007. Answering the call for a standard reliability measure for coding data. *Communication Methods and Measures*, 1(1):77–89.

M. A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th COLING Conference*, pages 539–545.

D. Jurafsky and J. H. Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall Series in Artificial Intelligence. 2nd Ed. Online draft (June 25, 2007).

Z. Kozareva, E. Riloff, and E. Hovy. 2008. Semantic Class Learning from the Web with Hyponym Pattern Linkage Graphs. In *Proceedings of the 46th ACL Meeting*, pages 1048–1056.

C. D. Manning and H. Schütze. 1999. *Foundations of statistical natural language processing*. MIT Press, Cambridge, MA, USA.

M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Building a large annotated corpus of English. The Penn Treebank. *Computational Linguistics*, 19:313–330.

S. Muresan and J. Klavans. 2002. A method for automatically building and evaluating dictionary resources. In *Proceedings of the 3rd LREC*, pages 231–234.

P. Pantel and M. Pennacchiotti. 2006. Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. In *Proceedings of the 21st COLING and the 44th ACL Meeting*, pages 113–120.

J. Pearson. 1996. The expression of definitions in specialised texts: A corpus-based analysis. In *Proceedings of Euralex-96*, pages 817–824.

C. Pollard and I. A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. Studies in Contemporary Linguistics. University of Chicago Press, Chicago.

U. Schäfer and B. Kiefer. 2011. Advances in deep parsing of scholarly paper content. In R. Bernardi, S. Chambers, B. Gottfried, F. Segond, and I. Zaihrayeu, editors, *Advanced Language Technologies for Digital Libraries*, number 6699 in LNCS, pages 135–153. Springer.

U. Schäfer, B. Kiefer, C. Spurk, J. Steffen, and R. Wang. 2011. The ACL Anthology Searchbench. In *Proceedings of ACL-HLT 2011, System Demonstrations*, pages 7–13, Portland, Oregon, June.

Y. Shen, G. Zaccak, B. Katz, Y. Luo, and O. Uzuner. 2006. Duplicate Removal for Candidate Answer Sentences. In *Proceedings of the 1st CSAIL Student Workshop*.

A. Thanopoulos, N. Fakotakis, and G. Kokkinakis. 2002. Comparative evaluation of collocation extraction metrics. In *Proceedings of the 3rd Language Resources Evaluation Conference*, pages 620–625.

P. Velardi, R. Navigli, and P. D'Amadio. 2008. Mining the Web to Create Specialized Glossaries. *IEEE Intelligent Systems*, pages 18–25.

S. Walter. 2008. Linguistic description and automatic extraction of definitions from german court decisions. In *Proceedings of the 6th LREC*, pages 2926–2932.

W. Weiten, D. Deguara, E. Rehmke, and L. Sewell. 1999. University, Community College, and High School Students' Evaluations of Textbook Pedagogical Aids. *Teaching of Psychology*, 26(1):19–21.

E. Westerhout and P. Monachesi. 2008. Creating glossaries using pattern-based and machine learning techniques. In *Proceedings of the 6th LREC*, pages 3074–3081.

M. Wolska, U. Schäfer, and The Nghia Pham. 2011. Bootstrapping a domain-specific terminological taxonomy from scientific text. In *Proceedings of the 9th International Conference on Terminology and Artificial Intelligence (TIA-11)*, pages 17–23. INALCO, Paris.

G. Ytrestøl, D. Flickinger, and S. Oepen. 2009. Extracting and annotating wikipedia sub-domains. In *Proceedings of the 7th Workshop on Treebanks and Linguistic Theories*, pages 185–197.

Appendix

Rated glossary sentences for 'word sense disambiguation (WSD)' and 'mutual information (MI)'. As shown in Figure 2, for WSD, mode ratings of LSP sentences were higher, while for MI it was the other way round.

word sense disambiguation (WSD)

mode ratings of LSP sentences:

WSD is the task of determining the sense of a polysemous word within a specific context (Wang et al., 2006).	5
Word sense disambiguation or WSD, the task of identifying the correct sense of a word in context, is a central problem for all natural language processing applications, and in particular machine translation: different senses of a word translate differently in other languages, and resolving sense ambiguity is needed to identify the right translation of a word.	4
Unlike previous applications of co-training and self-training to natural languagelearning, where one general classifier is build to cover the entire problem space, supervised word sense disambiguation implies a different classifier for each individual word, resulting eventually in thousands of different classifiers, each with its own characteristics (learning rate, sensitivity to new examples, etc.).	3
NER identifies different kinds of names such as "person", "location" or "date", while WSD distinguishes the senses of ambiguous words.	3
This paper presents a corpus-based approach to word sense disambiguation that builds an ensemble of Naive Bayesian classifiers, each of which is based on lexical features that represent co-occurring words in varying sized windows of context.	1

DA sentences:

Word Sense Disambiguation (WSD) is the task of formalizing the intended meaning of a word in context by selecting an appropriate sense from a computational lexicon in an automatic manner.	5
Word Sense Disambiguation(WSD) is the process of assigning a meaning to a word based on the context in which it occurs.	{4,5}
Word sense disambiguation (WSD) is a difficult problem in natural language processing.	2
word sense disambiguation, Hownet, sememe, co-occurrence Word sense disambiguation (WSD) is one of the most difficult problems in NLP.	{1,2}
There is a general concern within the field of word sense disambiguation about the inter-annotator agreement between human annotators.	1

mutual information (MI)

mode ratings of LSP sentences:

According to Fano (1961), if two points (words), x and y, have probabilities $P(x)$ and $P(y)$, then their mutual information, $I(x, y)$, is defined to be $I(x, y) = log_2 \frac{P(x,y)}{P(x)P(y)}$); informally, mutual information compares the probability of observing x and y together (the joint probability) with the probabilities of observing x and y independently (chance).	5
Mutual information, $I(v; c/s)$, measures the strength of the statistical association between the given verb v and the candidate class c in the given syntactic position s.	3
In this equation, $pmi(i, p)$ is the pointwise mutual information score (Church and Hanks, 1990) between a pattern, p (e.g. consist-of), and a tuple, i (e.g. engine-car), and max_{pmi} is the maximum PMI score between all patterns and tuples.	{1,3}
Note that while differential entropies can be negative and not invariant under change of variables, other properties of entropy are retained (Huber et al., 2008), such as the chain rule for conditional entropy which describes the uncertainty in Y given knowledge of X, and the chain rule for mutual information which describes the mutual dependence between X and Y.	2
The first term of the conditional probability measures the generality of the association, while the second term of the mutual information measures the co-occurrence of the association.	2

DA sentences:

Mutual information (Shannon and Weaver, 1949) is a measure of mutual dependence between two random variables.	4
3 Theory Mutual information is a measure of the amount of information that one random variable contains about another random variable.	4
Conditional mutual information is the mutual information of two random variables conditioned on a third one.	{1,3}
Thus, the mutual information is $log_2 5$ or 2.32 bits, meaning that the joint probability is 5 times more likely than chance.	1
Thus, the mutual information is $log_2 0$, meaning that the joint is infinitely less likely than chance.	1