Towards the Automatic Extraction of Definitions in Slavic

[1]**Adam Przepiórkowski**
[2]**Łukasz Degórski**
[8]**Beata Wójtowicz**
Institute of Computer Science PAS
Ordona 21, Warsaw, Poland
adamp@ipipan.waw.pl
ldegorski@bach.ipipan.waw.pl
beataw@bach.ipipan.waw.pl

[3]**Miroslav Spousta**
[7]**Vladislav Kuboň**
Charles University
Malostranské náměstí 25
Prague, Czech Republic
spousta@ufal.ms.mff.cuni.cz
vk@ufal.ms.mff.cuni.cz

[4]**Kiril Simov**
[5]**Petya Osenova**
Institute for Parallel Processing BAS
Bonchev St. 25A, Sofia, Bulgaria
kivs@bultreebank.org
petya@bultreebank.org

[6]**Lothar Lemnitzer**
University of Tübingen
Wilhelmstr. 19, Tübingen, Germany
lothar@sfs.uni-tuebingen.de

Abstract

This paper presents the results of the preliminary experiments in the automatic extraction of definitions (for semi-automatic glossary construction) from usually unstructured or only weakly structured e-learning texts in Bulgarian, Czech and Polish. The extraction is performed by regular grammars over XML-encoded morphosyntactically-annotated documents. The results are less than satisfying and we claim that the reason for that is the intrinsic difficulty of the task, as measured by the low interannotator agreement, which calls for more sophisticated deeper linguistic processing, as well as for the use of machine learning classification techniques.

1 Introduction

The aim of this paper is to report on the preliminary results of a subtask of the European Project *Language Technology for eLearning* (http://www.lt4el.eu/) consisting in the identification of term definitions in eLearning materials (Learning Objects; henceforth: LOs), where definitions are understood pragmatically, as those text fragments which may, after perhaps some minor editing, be put into a glossary. Such automatically extracted term definitions are to be presented to the author or the maintainer of the LO and, thus, significantly facilitate and accelerate the creation of a glossary for a given LO. From this specification of the task it follows that good recall is much more important than good precision, as it is easier to reject wrong glossary candidates than to browse the LO for term definitions which were not automatically spotted.

The project involves 9 European languages including 3 Slavic (and, regrettably, no Baltic) languages: one South Slavic, i.e., Bulgarian, and two West Slavic, i.e., Czech and Polish. For all languages, shallow grammars identifying definitions have been constructed; after mentioning some previous work on Information Extraction (IE) for Slavic languages and on extraction of definitions in section 2, we briefly describe the three Slavic grammars developed within this project in section 3. Section 4 presents the results of the application of these grammars to LOs in respective languages. These results are evaluated in section 5, where main problems, as well as some possible solutions, are discussed. Finally, section 6 concludes the paper.

Balto-Slavonic Natural Language Processing 2007, June 29, 2007, pages 43–50,
Prague, June 2007.

2 Related Work

Definition extraction is an important NLP task, most frequently a subtask of terminology extraction (Pearson, 1996), the automatic creation of glossaries (Klavans and Muresan, 2000; Klavans and Muresan, 2001), question answering (Miliaraki and Androutsopoulos, 2004; Fahmi and Bouma, 2006), learning lexical semantic relations (Malaisé et al., 2004; Storrer and Wellinghoff, 2006) and automatic construction of ontologies (Walter and Pinkal, 2006). Tools for definition extraction are invariably language-specific and involve shallow or deep processing, with most work done for English (Pearson, 1996; Klavans and Muresan, 2000; Klavans and Muresan, 2001) and other Germanic languages (Fahmi and Bouma, 2006; Storrer and Wellinghoff, 2006; Walter and Pinkal, 2006), as well as French (Malaisé et al., 2004). To the best of our knowledge, no previous attempts at definition extraction have been made for Slavic, with the exception of some work on Bulgarian (Tanev, 2004; Simov and Osenova, 2005).

Other work on Slavic information extraction has been carried out mainly for the last 5 years. Probably the first forum where such work was comprehensively presented was the International Workshop on Information Extraction for Slavonic and Other Central and Eastern European Languages (IESL), RANLP, Borovets, 2003, Bulgaria. One of the papers presented there, (Drożdżyński et al., 2003), discusses shallow SProUT (Becker et al., 2002) grammars for Czech, Polish and Lithuanian. SProUT has subsequently been extensively used for the information extraction from Polish medical texts (Piskorski et al., 2004; Marciniak et al., 2005).[1]

3 Shallow Grammars for Definition Extraction

The input to the task of definition extraction is XML-encoded morphosyntactically-annotated text, possibly with some keywords already marked by an independent process. For example, the representation of a Polish sentence starting as *Konstruktywizm kładzie nacisk na* (Eng. "Constructivism puts emphasis on") may be as follows:[2]

```
<s id="s9">
<markedTerm id="mt7" kw="y">
<tok base="konstruktywizm" ctag="subst"
 id="t253"
 msd="sg:nom:m3">Konstruktywizm</tok>
</markedTerm>
<tok base="klasc" ctag="fin" id="t254"
 msd="sg:ter:imperf">kladzie</tok>
<tok base="nacisk" ctag="subst" id="t255"
 msd="sg:acc:m3">nacisk</tok>
<tok base="na" ctag="prep" id="t256"
 msd="acc">na</tok>
[...]
<tok base="." ctag="interp" id="t273">.
</tok>
</s>
```

For each language, definitions were manually marked in two batches of texts: the first batch, consulted during the process of grammar development, contained at least 300 definitions, and the second batch, held out for evaluation, contained about 150 definitions. All grammars are regular grammars implemented with the use of the `lxtransduce` tool (Tobin, 2005), a component of the LTXML2 toolset developed at the University of Edinburgh.[3] An example of a simple rule for prepositional phrases is given below:

```
<rule name="PP">
 <seq>
  <query match="tok[@ctag = 'prep']"/>
  <ref name="NP1">
   <with-param name="case" value="''"/>
  </ref>
 </seq>
</rule>
```

This rule identifies a sequence whose first element is a token tagged as a preposition and whose subsequent elements are identified by a rule called `NP1`. This latter rule (not shown here for brevity) is a parameterised rule which finds a nominal phrase of a given case, but the way it is called above ensures that it will find an NP of any case.

[1] SProUT has not been seriously considered for the task at hand for two reasons: first, it was decided that only open source tools will be used in the current project, if only available, second, the input format to the current task is morphosyntactically-annotated XML-encoded text, rather than raw text, as normally expected by SProUT. The second obstacle could be removed by converting input texts to the SProUT-internal XML representation.

[2] Part of the representation has been replaced by '`[...]`'.

[3] Among the tools considered here were also CLaRK (Simov et al., 2001), ultimately rejected because it currently does not work in batch mode, and GATE / JAPE (Cunningham et al., 2002), not used here because we found GATE's handling of previously XML-annotated texts rather cumbersome and ill-documented. Cf. also fn. 1.

Currently the grammars show varying degrees of sophistication, with a small Bulgarian grammar (8 rules in a 2.5-kilobyte file), a larger Polish grammar (34 rules in a 11 KiB file) and a sophisticated Czech grammar most developed (147 rules in a 28 KiB file). The patterns defined by these three grammars are similar, but sufficiently different to defy an attempt to write a single parameterised grammar.[4] The remainder of this section briefly describes the grammars.

3.1 Bulgarian

The Bulgarian grammar is manually constructed after examination of the manually annotated definitions. Here is a list of the rule schemata, together with the number and percentage of matching definitions:

Pattern	#	%
NP is NP	140	34.2
NP verb NP	18	29.8
NP - NP	21	5.0
This is NP	15	3.7
It represents NP	4	1.0
other patterns	107	26.2

Table 1: Bulgarian definition types

In the second schema above, “verb” is a verb or a verb phrase (not necessarily a constituent) which is one of the following: ‘представлява’ (to represent), ‘показва’ (to show), ‘означава’ (to mean), ‘описва’ (to describe), ‘се използва’ (to be used), ‘позволява’ (to allow), ‘дава възможност да’ (to give opportunity), ‘се нарича’ (is called), ‘подобрява’ (to improve), ‘осигурява’ (to ensure), ‘служи за’ (to serve as), ‘се разбира’ (to be understood as), ‘обозначава’ (to denote), ‘съдържа’ (to contain), ‘определя’ (to determine), ‘включва’ (to include), ‘се дефинира като’ (is defined as), ‘се основава на’ (is based on).

We classify the rules in five types: copula definitions, copula definitions with anaphoric relation, copula definitions with ellipsis of the copula, definitions with a verb phrase, definitions with a verb phrase and anaphoric relation. Each of these types of definitions defines an NP (sometimes via anaphoric relation) by another one. There are some variations of the models where some parenthetical expressions are presented in the definition.

The grammar contains several most important rules for each type. The different verb patterns are encoded as a lexicon. For some of the rules, variants with parenthetical phrases are also encoded. The rest of the grammar is devoted to the recognition of noun phrases and parenthetical phrases. For parenthetical phrases, we have encoded a list of such possible phrases, extracted on the basis of a bigger corpus. The NP grammar in our view is the crucial grammar for recognition of the definitions. Most work now has to be invested into developing the more complex and recursive NPs.

3.2 Czech

The Czech grammar for definition context extraction is constructed to follow both linguistic intuition and observation of common patterns in manually annotated data.

We adapted a grammar[5] based mainly on the observation of Czech Wikipedia entries. Encyclopedia definitions are usually clear and very well structured, but it is quite difficult to find such well-formed definitions in common texts, including learning objects. The rules were extended using part of our manually annotated texts, evaluated and adjusted in several iterations, based on the observation of the annotated data.

Pattern	#	%
NP is/are NP	52	21.2
NP verb NP	45	18.4
structural	39	15.9
NP (NP)	30	12.2
NP -/:/= NP	20	8.2
other patterns	59	24.1

Table 2: Czech definition types

There are 21 top level rules, divided into five categories. Most of the correctly marked definitions fall into the copula verb (‘is/are’) category. The sec-

[4]Because of this relative language-dependence of definition patters, which includes, e.g., idiosyncratic case information, we have not seriously considered re-using rules for other, non-Slavic, languages.

[5]The grammar was originally developed by Nguyen Thu Trang.

ond most successful rule is the one using selected verbs like ‘definuje’ (defines), ‘znamená’ (means), ‘vymezuje’ (delimits), ‘představuje’ (presents) and several others. The remaining categories make use of the typical patterns of characters (dash, colon, equal sign and brackets) or additional structural information (e.g., HTML tags).

3.3 Polish

The Polish grammar rules are divided into three layers. Similarly to the Czech grammar, each layer only refers to itself or lower layers. This allows for expressing top level rules in a clear and easily manageable way.

The top level layer consists of rules representing typical patterns found in Polish documents:

Pattern	#	%
NP (...) are/is NP-INS	40	15.6
NP -/: NP	39	15.2
NP (are/is) *to* NP-NOM	27	10.6
NP VP-3PERS	25	9.8
NP - i.e./or WH-question	11	4.3
N ADJ - PPAS	8	3.1
NP, i.e./or NP	7	2.7
NP-ACC one may describe/define as NP-ACC	5	2.0
other patterns (not in the grammar)	94	36.7

Table 3: Polish definition types

The middle layer consists of rules catching patterns such as “simple NP in given case, followed by a sequence of non-punctuation elements” or “copula”.

The bottom layer rules basically only refer to POS markup in the input files (or other bottom layer rules).

4 Results

As mentioned above, the testing corpus for each language consists of about 150 definitions, unseen during the construction of the grammar.[6]

The Bulgarian test corpus, containing around 76,800 tokens, consists of the third part of the Calimera guidelines (`http://www.calimera.org/`). We view this document as appropriate for testing because it reflects the chosen domain and it combines definitions from otherwise different subdomains, such as XML language, Internet usage, etc. There are 203 manually annotated definitions in this corpus: 129 definitions contained in one sentence, 69 definitions split across 2 sentences, 4 definitions in 3 sentences and one definition in 4 sentences. Note that the real test part is the set of the 129 definitions in one sentence, since the Bulgarian grammar does not consider cross-sentence definitions in any way.

Czech data used for evaluation consist of several chapters of the Calimera guidelines and Microsoft Excel tutorial. The tutorial is a typical text used in e-learning, consisting of five chapters describing sheets, tables, formating, graphs and lists. The corpus consists of over 90,000 tokens and contains 162 definitions, out of which 153 are contained in a single sentence, 6 span 2 sentences, and 3 definitions span 3 sentences.

Polish test corpus consists of over 83,200 tokens containing 157 definitions: 148 definitions are contained within one sentence, while 9 span 2 sentences. The corpus is made up of 10 chapters of a popular introduction to and history of computer science and computer hardware.

Each grammar was quantitatively evaluated by comparing manually annotated files with the same files annotated automatically by the grammar. After considering various ways of quantitative evaluation, we decided to do the comparison at token level: precision was calculated as the ratio of the number of those tokens which were parts of *both* a manually marked definition and an automatically discovered definition to the number of all tokens in automatically discovered definitions, while recall was taken to be the ratio of the number of tokens simultaneously in both kinds of definitions to the number of tokens in all manually annotated definitions. Since, for this task, recall is more important than precision, we used the F_2-measure for the combined result.[7]

[6]Obviously, three different corpora had to be used to evaluate the grammars for the three languages, but the corpora are similar in size and character, so any differences in results stem mostly from the differences in the three grammars.

[7]In general, $F_\alpha = (1+\alpha)\cdot(\text{precision}\cdot\text{recall})/(\alpha\cdot\text{precision}+\text{recall})$. Perhaps α larger than 2 could be used, but it is currently not clear to us what criteria should be assumed when

The results for the three grammars are given in Table 4. Note that the processing model for Czech differs from the other two languages, as the input text is converted to a flat format, as described in section 5.3, and grammar rules are sensitive to sentence boundaries (and may operate over them).

	precision	recall	F_2
Bulgarian	20.5%	2.2%	3.1
Czech	18.3%	40.7%	28.9
Polish	14.8%	22.2%	19.0

Table 4: Token-based evaluation of shallow grammars

5 Evaluation and Possible Improvements

5.1 Interannotator Agreement

We calculated Cohen's kappa statistic (1) for the current task, where both the relative observed agreement among raters $\Pr(a)$ and the probability that agreement is due to chance $\Pr(e)$ where calculated at token level.

$$\kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)} \quad (1)$$

More specifically, we assumed that two annotators agree on a token if the token belongs to a definition either according to both annotations or according to neither. In order to estimate the probability of agreement due to chance $\Pr(e)$, we measured, separately for each annotator, the proportion of tokens found in definitions to all tokens in text, which resulted in two probability estimates p_1 and p_2, and treated $\Pr(e)$ as the probability that the two annotators agree if they randomly, with their own probability, classify a token as belonging to a definition, i.e.:

$$\Pr(e) = p_1 \cdot p_2 + (1 - p_1) \cdot (1 - p_2) \quad (2)$$

The interannotator agreement (IAA) was measured this way for Czech and Polish, where — for each language — the respective test corpus was annotated by two annotators. The results are 0.44 for Czech and 0.31 for Polish. Such results are very low for any classification task, and especially low for a binary classification task. They show that the task of identifying definitions in running texts and agreeing on which parts of text count as a definition is intrinsically very difficult. They also call for the reconsideration of the evaluation and IAA measurement methodology based on token classification.[8]

deciding on the exact value of α. Note that it would not make sense to use recall alone, as it is trivial to write all-accepting grammars with 100% recall.

5.2 Evaluation Methodology

To the best of our knowledge, there is no established evaluation methodology for the task of definition extraction, where definitions may span several sentences.[9] For this reason we evaluated the results again, in a different way: we treated an automatically discovered definition as correct, if it overlapped with a manually annotated definition. We calculated precision as the number of automatic definitions overlapping with manual definitions, divided by the number of automatic definitions, while recall — as the number of manual definitions overlapping automatic definitions, divided by the number of manual definitions.[10]

The results for the three grammars, given in Table 5, are much higher than those in Table 4 above, although still less than satisfactory.

	precision	recall	F_2
Bulgarian	22.5%	8.9%	11.1
Czech	22.3%	46%	33.9
Polish	23.3%	32%	28.4

Table 5: Definition-based evaluation of shallow grammars

5.3 Definitions and Sentence Boundaries

Regardless of the inherent difficulties of the task and difficulties with the evaluation of the results, there is clear room for improvement; one possible path

[8] A better approximation would be to measure IAA on the basis of sentence or (as suggested by an anonymous reviewer) NP classification; we intend to pursue this idea in future work.

[9] With the assumption that definitions are no longer than a sentence, usually the task is treated as a classification task, where sentences are classified as definitional or not, and appropriate precision and recall measures are applied at sentence level.

[10] At this stage definition fragments distributed across a number of different sentences were treated as different definitions, which negatively affects the evaluation of the Bulgarian grammar, as the Bulgarian test corpus contains a large number of multi-sentence definitions.

to explore concerns multi-sentence definitions. As noted above, for all languages considered here, there were definitions which were spanning 2 or more sentences; this turned out to be a problem especially for Bulgarian, were 36% of definitions crossed a sentence boundary.[11]

Such multi-sentence definitions are a problem because in the DTD adopted in this project definitions are subelements of sentences rather than the other way round. In case of a multi-sentence definition, for each sentence there is a separate element encapsulating the part of the definition contained in this sentence. Although these are linked via special attributes and the information that they are part of the same definition can subsequently be recovered, it is difficult to construct an `lxtransduce` grammar which would be able to automatically mark such multi-sentence definitions: an `lxtransduce` grammar expects to find a sequence of elements and wrap them in a single larger element.

A solution to this technical problem has been implemented in the Czech grammar, where first the input text is flattened (via an XSLT script), so that, e.g.:

```
<par id="d1p2">
 <s id="d1p2s1">
  <tok id="d1p2s1t1" base="Pavel"
    ctag="N" msd="NMS1-----A---->
    Pavel</tok>
  <tok id="d1p2s1t2" base="satrapa"
   ctag="N" msd="NMS1-----A---->
   Satrapa</tok>
 </s>
</par>
```

becomes:

```
<par id="Sd1p2"/>
<s id="Sd1p2s1"/>
<tok id="d1p2s1t1" base="Pavel"
 ctag="N" msd="NMS1-----A---->
 Pavel</tok>
<tok id="d1p2s1t2" base="satrapa"
 ctag="N" msd="NMS1-----A---->
 Satrapa</tok>
<s id="Ed1p2s1"/>
<par id="Ed1p2"/>
```

[11] An example of a Polish manually annotated multi-sentence definition is: ... *opracowano techniki antyspamowe. Techniki te drastycznie zaniżają wartość strony albo ją banują...* (Eng. "... anti-spam techniques were developed. Such techniques drastically lower the value of the page or they ban it..."). The definition is split into two fragments fully contained in respective sentences: *techniki antyspamowe* and *Techniki te...*. No attempt at anaphora resolution is made.

This flattened representation is an input to a grammar which is sensitive to the empty `s` and `par` elements and may discover definitions containing such elements; in such a case, the postprocessing script, which restores the hierarchical paragraph and sentence structure, splits such definitions into smaller elements, fully contained in respective sentences.

5.4 Problems Specific to Slavic

At least in case of the two West Slavic languages considered here, the task of writing a definition grammar is intrinsically more difficult than for Germanic or Romance languages, mainly for the following two reasons.

First, Czech and Polish have very rich nominal inflection with a large number of paradigm-internal syncretisms. These syncretisms are a common cause of tagger errors, which percolate to further stages of processing. Moreover, the number of cases makes it more difficult to encode patterns like "NP verb NP", as different verbs may combine with NPs of different case. In fact, even two different copulas in Polish take different cases!

Second, the relatively free word order increases the number of rules that must be encoded, and makes the grammar writing task more labour-intensive and error-prone. The current version of the Polish grammar, with 34 rules, is rather basic, and even the 147 rules of the Czech grammar do not take into consideration all possible patterns of grammar definitions. As Tables 4 and 5 show, there is a positive correlation between the grammar size and the value of F_2, and the Bulgarian and Polish grammars certainly have room to grow. Moreover, a path that is well worth exploring is to drastically increase the number of rules and, hence, the recall, and then deal with precision via Machine Learning methods (cf. section 5.6).

5.5 Levels of Linguistic Processing

The work reported here has been an excercise in definition extraction using shallow parsing methods. However, the poor results suggest that this is one of the tasks that require a much more sophisticated and deeper approach to language analysis. In fact, in turns out that virtually all successful attempts at definition extraction that we are aware of build on worked-out deep linguistic approaches (Klavans and

Muresan, 2000; Fahmi and Bouma, 2006; Walter and Pinkal, 2006), some of them combining syntactic and semantic information (Miliaraki and Androutsopoulos, 2004; Walter and Pinkal, 2006).

Unfortunately, for most Baltic and Slavic languages, such deep parsers are unavailable or have not yet been extensively tested on real texts. One exception is Czech, where a number of parsers were already described and evaluated (on the Prague Dependency Treebank) in (Zeman, 2004, § 14.2); the best of these parsers reach 80–85% accuracy.

For Polish, apart from a number of linguistically motivated toy parsers, there is a possibly wide coverage deep parser (Woliński, 2004), but it has not yet been evaluated on naturally occurring texts. The situation is probably most dire for Bulgarian, although there have been attempts at the induction of a dependency parser from the BulTreeBank (Marinov and Nivre, 2005; Chanev et al., 2006).

Nevertheless, if other possible paths of improvement suggested in this section do not bring satisfactory results, we plan to make an attempt at adapting these parsers to the task at hand.

5.6 Postprocessing: Machine Learning and Keyword Identification

Various approaches to the machine learning treatment of the task of classifying sentences or snippets as definitions or non-definitions can be found, e.g., in (Miliaraki and Androutsopoulos, 2004; Fahmi and Bouma, 2006) and references therein. In the context of the present work, such methods may be used to postprocess apparent definitions found at earlier processing stages and decide which of them are genuine definitions. For example, (Fahmi and Bouma, 2006) report that a system trained on 2299 sentences, including 1366 definition sentences, may increase the accuracy of a definition extraction tool from 59% to around 90%.[12]

Another possible improvement may consist in, again, aiming at very high recall and then using an independent keyword detector to mark keywords (and key phrases) in text and classifying as genuine definitions those definitions, whose defined term has been marked as a keyword.

Whatever postprocessing technique or combination of techniques proves most efficient, it seems that the linguistic processing should aim at high recall rather than high precision, which further justifies the use of the F_2 measure for evaluation.[13]

6 Conclusion

To the best of our knowledge, this paper is the first report on the task of definition extraction for a number of Slavic languages. It shows that the task is intrinsically very difficult, which partially explains the relatively low results obtained. It also calls attention to the fact that there is no established evaluation methodology where possibly multi-sentence definitions are involved and suggests what such methodology could amount to. Finally, the paper suggests ways of improving the results, which we hope to follow and report in the future.

References

Markus Becker et al. 2002. SProUT — shallow processing with typed feature structures and unification. In *Proceedings of the International Conference on NLP (ICON 2002)*, Mumbai, India.

Sharon A. Caraballo. 2001. *Automatic Construction of a Hypernym-Labeled Noun Hierarchy from Text*. Ph. D. dissertation, Brown University.

Atanas Chanev, Kiril Simov, Petya Osenova, and Svetoslav Marinov. 2006. Dependency conversion and parsing of the BulTreeBank. In *proceedings of the LREC workshop Merging and Layering Linguistic Information*, Genoa, Italy.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*.

Witold Drożdżyński, Petr Homola, Jakub Piskorski, and Vytautas Zinkevičius. 2003. Adapting SProUT to processing Baltic and Slavonic languages. In *Information Extraction for Slavonic and Other Central and Eastern European Languages*, pp. 18–25.

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*.

[12] The numbers are so high "probably due to the fact that the current corpus consists of encyclopedic material only" (Fahmi and Bouma, 2006, fn. 4).

[13] Note that the situation here is different than in the task of acquiring hyponymic relations from texts, where high-precision manual rules (Hearst, 1992) must be augmented with statistical clustering methods to increase recall (Caraballo, 2001).

Marti Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the Fourteenth International Conference on Computational Linguistics*, Nantes, France.

Judith L. Klavans and Smaranda Muresan. 2000. DEFINDER: Rule-based methods for the extraction of medical terminology and their associated definitions from on-line text. In *Proceedings of the Annual Fall Symposium of the American Medical Informatics Association*.

Judith L. Klavans and Smaranda Muresan. 2001. Evaluation of the DEFINDER system for fully automatic glossary construction. In *Proceedings of AMIA Symposium 2001*.

Véronique Malaisé, Pierre Zweigenbaum, and Bruno Bachimont. 2004. Detecting semantic relations between terms in definitions. In S. Ananadiou and P. Zweigenbaum, editors, *COLING 2004 CompuTerm 2004: 3rd International Workshop on Computational Terminology*, pp. 55–62, Geneva, Switzerland. COLING.

Małgorzata Marciniak, Agnieszka Mykowiecka, Anna Kupść, and Jakub Piskorski. 2005. Intelligent content extraction from Polish medical texts. In L. Bolc et al., editors, *Intelligent Media Technology for Communicative Intelligence, Second International Workshop, IMTCI 2004, Warsaw, Poland, September 13-14, 2004, Revised Selected Papers*, volume 3490 of *Lecture Notes in Computer Science*, pp. 68–78. Springer-Verlag.

Svetoslav Marinov and Joakim Nivre. 2005. A data-driven parser for Bulgarian. In *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories*, pp. 89–100, Barcelona.

Spyridoula Miliaraki and Ion Androutsopoulos. 2004. Learning to identify single-snippet answers to definition questions. In *Proceedings of COLING 2004*, pp. 1360–1366, Geneva, Switzerland. COLING.

Jennifer Pearson. 1996. The expression of definitions in specialised texts: a corpus-based analysis. In M. Gellerstam et al., editors, *Proceedings of the Seventh Euralex International Congress*, pp. 817–824, Göteborg.

Jakub Piskorski et al. 2004. Information extraction for Polish using the SProUT platform. In M. A. Kłopotek et al., editors, *Intelligent Information Processing and Web Mining*, pp. 227–236. Springer-Verlag, Berlin.

Kiril Simov and Petya Osenova. 2005. BulQA: Bulgarian-Bulgarian Question Answering at CLEF 2005. In *CLEF*, pp. 517–526.

Kiril Simov et al. 2001. CLaRK — an XML-based system for corpora development. In P. Rayson et al., editors, *Proceedings of the Corpus Linguistics 2001 Conference*, pp. 558–560, Lancaster.

Angelika Storrer and Sandra Wellinghoff. 2006. Automated detection and annotation of term definitions in German text corpora. In *Proceedings of LREC 2006*.

Hristo Tanev. 2004. Socrates: A question answering prototype for Bulgarian. In *Recent Advances in Natural Language Processing III, Selected Papers from RANLP 2003*, pages 377–386. John Benjamins.

Richard Tobin, 2005. *Lxtransduce, a replacement for fsgmatch*. University of Edinburgh. `http://www.cogsci.ed.ac.uk/~richard/ltxml2/lxtransduce-manual.html`.

Stephan Walter and Manfred Pinkal. 2006. Automatic extraction of definitions from German court decisions. In *Proceedings of the Workshop on Information Extraction Beyond The Document*, pp. 20–28, Sydney, Australia. Association for Computational Linguistics.

Marcin Woliński. 2004. *Komputerowa weryfikacja gramatyki Świdzińskiego*. Ph. D. dissertation, ICS PAS, Warsaw.

Daniel Zeman. 2004. *Parsing with a Statistical Dependency Model*. Ph. D. dissertation, Charles University, Prague.