The ACL Anthology Searchbench

Ulrich Schäfer Bernd Kiefer Christian Spurk Jörg Steffen Rui Wang
Language Technology Lab
German Research Center for Artificial Intelligence (DFKI)
D-66123 Saarbrücken, Germany
{ulrich.schaefer,kiefer,cspurk,steffen,wang.rui}@dfki.de
http://www.dfki.de/lt

Abstract

We describe a novel application for structured search in scientific digital libraries. The ACL Anthology Searchbench is meant to become a publicly available research tool to query the content of the ACL Anthology. The application provides search in both its bibliographic metadata and semantically analyzed full textual content. By combining these two features, very efficient and focused queries are possible. At the same time, the application serves as a showcase for the recent progress in natural language processing (NLP) research and language technology. The system currently indexes the textual content of 7,500 anthology papers from 2002–2009 with predicate-argument-like semantic structures. It also provides useful search filters based on bibliographic metadata. It will be extended to provide the full anthology content and enhanced functionality based on further NLP techniques.

1 Introduction and Motivation

Scientists in all disciplines nowadays are faced with a flood of new publications every day. In addition, more and more publications from the past become digitally available and thus even increase the amount. Finding relevant information and avoiding duplication of work have become urgent issues to be addressed by the scientific community.

The organization and preservation of scientific knowledge in scientific publications, vulgo text documents, thwarts these efforts. From a viewpoint of a computer scientist, scientific papers are just 'unstructured information'. At least in our own scientific community, Computational Linguistics, it is generally assumed that NLP could help to support search in such document collections.

The ACL Anthology[1] is a comprehensive electronic collection of scientific papers in our own field (Bird et al., 2008). It is updated regularly with new publications, but also older papers have been scanned and are made available electronically.

We have implemented the ACL Anthology Searchbench[2] for two reasons: Our first aim is to provide a more targeted search facility in this collection than standard web search on the anthology website. In this sense, the Searchbench is meant to become a service to our own community.

Our second motivation is to use the developed system as a showcase for the progress that has been made over the last years in precision-oriented deep linguistic parsing in terms of both efficiency and coverage, specifically in the context of the DELPH-IN community[3]. Our system also uses further NLP techniques such as unsupervised term extraction, named entity recognition and part-of-speech (PoS) tagging.

By automatically precomputing normalized semantic representations (predicate-argument structure) of each sentence in the anthology, the search space is structured and allows to find equivalent or related predicates even if they are expressed differ-

[1]http://www.aclweb.org/anthology
[2]http://aclasb.dfki.de
[3]http://www.delph-in.net – DELPH-IN stands for DEep Linguistic Processing with HPSG INitiative.

ently, e.g. in passive constructions, using synonyms, etc. By storing the semantic sentence structure along with the original text in a structured full-text search engine, it can be guaranteed that recall cannot fall behind the baseline of a fulltext search.

In addition, the Searchbench also provides detailed bibliographic metadata for filtering as well as autosuggest texts for input fields computed from the corpus – two further key features one can expect from such systems today, nevertheless very important for efficient search in digital libraries.

We describe the offline preprocessing and deep parsing approach in Section 2. Section 3 concentrates on the generation of the semantic search index. In Section 4, we describe the search interface. We conclude in Section 5 and present an outlook to future extensions.

2 Parsing the ACL Anthology

The basis of the search index for the ACL Anthology are its original PDF documents, currently 8,200 from the years 2002 through 2009. To overcome quality problems in text extraction from PDF, we use a commercial PDF extractor based on OCR techniques. This approach guarantees uniform and high-quality textual representations even from older papers in the anthology (before 2000) which mostly were scanned from printed paper versions.

The general idea of the semantics-oriented access to scholarly paper content is to parse each sentence they contain with the open-source HPSG (Pollard and Sag, 1994) grammar for English (ERG; Flickinger (2002)) and then distill and index semantically structured representations for search.

To make the deep parser robust, it is embedded in a NLP workflow. The coverage (percentage of full deeply parsed sentences) on the anthology corpus could be increased from 65 % to now more than 85 % through careful combination of several robustness techniques; for example: (1) *chart pruning*, directed search during parsing to increase performance, and also coverage for longer sentences (Cramer and Zhang, 2010); (2) *chart mapping*, a novel method for integrating preprocessing information in exactly the way the deep grammar expects it (Adolphs et al., 2008); (3) new version of the ERG with better handling of open word classes; (4) more fine-grained named entity recognition, including recognition of citation patterns; (5) new, better suited parse ranking model (WeScience; Flickinger et al. (2010)). Because of limited space, we will focus on (1) and (2) below. A more detailed description and further results are available in (Schäfer and Kiefer, 2011).

Except for a small part of the named entity recognition components (citations, some terminology) and the parse ranking model, there are no further adaptations to genre or domain of the text corpus. This implies that the NLP workflow could be easily and modularly adapted to other (scientific or non-scientific) domains—mainly thanks to the generic and comprehensive language modelling in the ERG.

The NLP preprocessing component workflow is implemented using the Heart of Gold NLP middleware architecture (Schäfer, 2006). It starts with sentence boundary detection (SBR) and regular expression-based tokenization using its built-in component JTok, followed by the trigram-based PoS tagger TnT (Brants, 2000) trained on the Penn Treebank (Marcus et al., 1993) and the named entity recognizer SProUT (Drożdżyński et al., 2004).

2.1 Precise Preprocessing Integration with Chart Mapping

Tagger output is combined with information from the named entity recognizer, e.g. delivering hypothetical information on citation expressions. The combined result is delivered as input to the deep parser PET (Callmeier, 2000) running the ERG. Here, citations, for example, can be treated as either persons, locations or appositions.

Concerning punctuation, the ERG can make use of information on opening and closing quotation marks. Such information is often not explicit in the input text, e.g. when, as in our setup, gained through OCR which does not distinguish between ‘ and ’ or “ and ”. However, a tokenizer can often guess (reconstruct) leftness and rightness correctly. This information, passed to the deep parser via chart mapping, helps it to disambiguate.

2.2 Increased Processing Speed and Coverage through Chart Pruning

In addition to a well-established discriminative maximum entropy model for post-analysis parse selection, we use an additional generative model as described in Cramer and Zhang (2010) to restrict the search space during parsing. This restriction increases efficiency, but also coverage, because the parse time was restricted to at most 60 CPU seconds on a standard PC, and more sentences could now be parsed within these bounds. A 4 GB limit for main memory consumption was far beyond what was ever needed. We saw a small but negligible decrease in parsing accuracy, 5.4 % best parses were not found due to the pruning of important chart edges.

Ninomiya et al. (2006) did a very thorough comparison of different performance optimization strategies, and among those also a local pruning strategy similar to the one used here. There is an important difference between the systems, in that theirs works on a reduced context-free backbone first and reconstructs the results with the full grammar, while PET uses the HPSG grammar directly, with subsumption packing and partial unpacking to achieve a similar effect as the packed chart of a context-free parser.

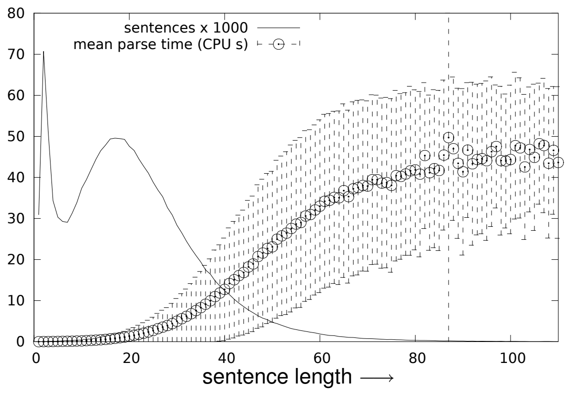


Figure 1: Distribution of sentence length and mean parse times for mild pruning

In total, we parsed 1,537,801 sentences, of which 57,832 (3.8 %) could not be parsed because of lexicon errors. Most of them were caused by OCR artifacts resulting in unexpected punctuation character combinations. These can be identified and will be deleted in the future.

Figure 1 displays the average parse time of processing with a mild chart pruning setting, together with the mean quadratic error. In addition, it contains the distribution of input sentences over sentence length. Obviously, the vast majority of sentences has a length of at most 60 words[4]. The parse times only grow mildly due to the many optimization techniques in the original system, and also the new chart pruning method. The sentence length distribution has been integrated into Figure 1 to show that the predominant part of our real-world corpus can be processed using this information-rich method with very low parse times (overall average parse time < 2 s per sentence).

The large amount of short inputs is at first surprising, even more so that most of these inputs can not be parsed. Most of these inputs are non-sentences such as headings, enumerations, footnotes, table cell content. There are several alternatives to deal with such input, one to identify and handle them in a preprocessing step, another to use a special root condition in the deep analysis component that is able to combine phrases with well-defined properties for inputs where no spanning result could be found.

We employed the second method, which has the advantage that it handles a larger range of phenomena in a homogeneous way. Figure 2 shows the change in percentage of unparsed and timed out inputs for the mild pruning method with and without the root condition combining fragments.

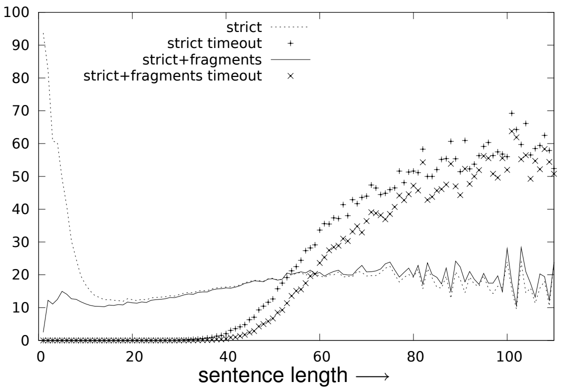
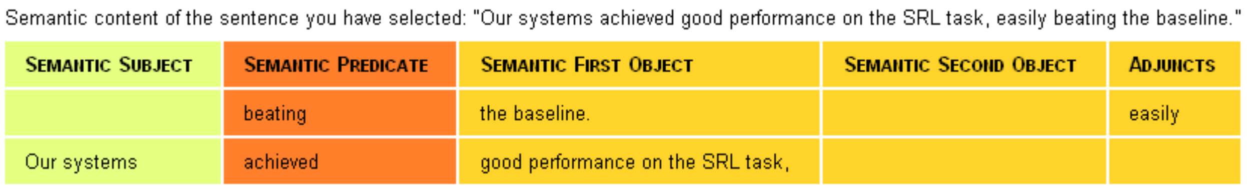


Figure 2: Unparsed and timed out sentences with and without fragment combination

Figure 2 shows that this changes the curve for unparsed sentences towards more expected characteristics and removes the uncommonly high percentage of short sentences for which no parse can be computed. Together with the parses for fragmented

[4]It has to be pointed out that extremely long sentences also may be non-sentences resulting from PDF extraction errors, missing punctuation etc. No manual correction took place.


Semantic content of the sentence you have selected: "Our systems achieved good performance on the SRL task, easily beating the baseline."

Semantic Subject	Semantic Predicate	Semantic First Object	Semantic Second Object	Adjuncts
	beating	the baseline.		easily
Our systems	achieved	good performance on the SRL task,		

Figure 3: Multiple semantic tuples may be generated for a sentence

input, we get a recall (sentences with at least one parse) over the whole corpus of 85.9 % (1,321,336 sentences), without a significant change for any of the other measures, and with potential for further improvement.

3 Semantic Tuple Extraction with DMRS

In contrast to shallow parsers, the ERG not only handles detailed syntactic analyses of phrases, compounds, coordination, negation and other linguistic phenomena that are important for extracting semantic relations, but also generates a formal semantic representation of the meaning of the input sentence in the Minimal Recursion Semantics (MRS) representation format (Copestake et al., 2005). It consists of elementary predications for each word and larger constituents, connected via argument positions and variables, from which predicate-argument structure can be extracted.

MRS representations resulting from deep parsing are still relatively close to linguistic structures and contain more detailed information than a user would like to query and search for. Therefore, an additional extraction and abstraction step is performed before storing semantic structures in the search index.

Firstly, MRS is converted to DMRS (Copestake, 2009), a dependency-style version of MRS that eases extraction of predicate-argument structure using the implementation in LKB (Copestake, 2002). The representation format we devised for the search index we call *semantic tuples*, in fact quintuples <subject, predicate, first object, second object, adjuncts>; example in Figure 3. The basic extraction algorithm consists of the following three steps: (1) calculate the closure for each 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 in the corresponding information under its scope, i.e. subject, object, etc.

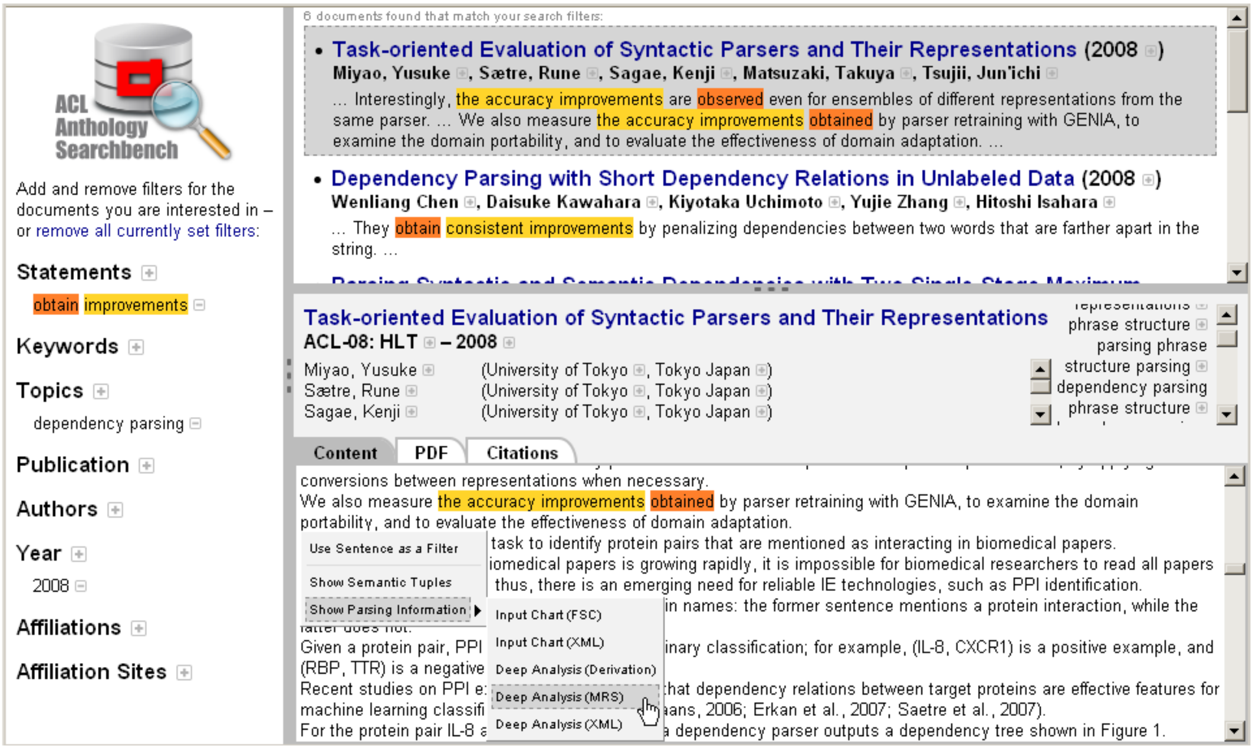
In the example shown in Figure 3, entity groups like ‘our systems’, ‘the baseline’, and ‘good performance on the SRL task’, as well as predicate groups ‘beating’ and ‘achieved’ are formed at the first step. In the second step, the graph structure is extracted, i.e., the relation between the groups. Finally, two semantic tuples are filled in with both the predicates and the corresponding information. Notice that the modifier(s) of the entity belong to the same entity group, but the modifier(s) of the predicate will be put into the Adjuncts slot. Similarly, the coordination of the entities will be put into one entity group, while the coordination of predicates will form multiple semantic tuples.

Since we are extracting predicate-argument structure, syntactic variations such as passive constructions and relative clauses will be all ‘normalized’ into the same form. Consequently, ‘the book which I read’, ‘I read the book’, and ‘the book was read by me’ will form the exact same semantic tuple <I, read, the book, N/A, N/A>. The resulting tuple structures along with their associated text are stored in an Apache Solr/Lucene[5] server which receives queries from the Searchbench user interface.

4 Searchbench User Interface

The Searchbench user interface (UI) is a web application running in every modern, JavaScript-enabled web browser. As can be seen in Figure 4, the UI is divided into three parts: (1) a sidebar on the left (*Filters View*), where different filters can be set that constrain the list of found documents; (2) a list of found documents matching the currently set filters in the upper right part of the UI (*Results View*); (3)

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



Add and remove filters for the documents you are interested in – or remove all currently set filters:

Statements
obtain improvements
Keywords
Topics
dependency parsing
Publication
Authors
Year
2008
Affiliations
Affiliation Sites

6 documents found that match your search filters:

- **Task-oriented Evaluation of Syntactic Parsers and Their Representations** (2008)
  Miyao, Yusuke, Sætre, Rune, Sagae, Kenji, Matsuzaki, Takuya, Tsujii, Jun'ichi
  ... Interestingly, the accuracy improvements are observed even for ensembles of different representations from the same parser. ... We also measure the accuracy improvements obtained by parser retraining with GENIA, to examine the domain portability, and to evaluate the effectiveness of domain adaptation. ...
- **Dependency Parsing with Short Dependency Relations in Unlabeled Data** (2008)
  Wenliang Chen, Daisuke Kawahara, Kiyotaka Uchimoto, Yujie Zhang, Hitoshi Isahara
  ... They obtain consistent improvements by penalizing dependencies between two words that are farther apart in the string. ...

Task-oriented Evaluation of Syntactic Parsers and Their Representations
ACL-08: HLT – 2008

Miyao, Yusuke	(University of Tokyo, Tokyo Japan)
Sætre, Rune	(University of Tokyo, Tokyo Japan)
Sagae, Kenji	(University of Tokyo, Tokyo Japan)

phrase structure
parsing phrase
structure parsing
dependency parsing
phrase structure

conversions between representations when necessary.
We also measure the accuracy improvements obtained by parser retraining with GENIA, to examine the domain portability, and to evaluate the effectiveness of domain adaptation.
task to identify protein pairs that are mentioned as interacting in biomedical papers.
iomedical papers is growing rapidly, it is impossible for biomedical researchers to read all papers
thus, there is an emerging need for reliable IE technologies, such as PPI identification.
in names: the former sentence mentions a protein interaction, while the
Given a protein pair, PPI e inary classification; for example, (IL-8, CXCR1) is a positive example, and
(RBP, TTR) is a negative
Recent studies on PPI e hat dependency relations between target proteins are effective features for
machine learning classifi ans, 2006; Erkan et al., 2007; Saetre et al., 2007).
For the protein pair IL-8 a dependency parser outputs a dependency tree shown in Figure 1.

Figure 4: Searchbench user interface with different filters set and currently looking at the debug menu for a sentence.

the *Document View* in the lower right part of the UI with different views of the current document.

A focus in the design of the UI has been to allow the user to very quickly browse the papers of the ACL Anthology and then to find small sets of relevant documents based on metadata and content. This is mainly achieved by these techniques: (i) changes in the collection of filters automatically update the Results View; (ii) metadata and searchable content from both the Results View and the Document View can easily be used with a single click as new filters; (iii) filters can easily be removed with a single click; (iv) manually entering filter items is assisted by sensible autosuggestions computed from the corpus; (v) accidental filter changes can easily be corrected by going back in the browser history.

The following kinds of filters are supported: **Statements** (filter by semantic statements, i.e., the actual content of sentences, see Section 4.1), **Keywords** (filter by simple keywords with a full-text search), **Topics** (filter by topics of the articles that were extracted with an extended approach of the unsupervised term extractor of Frantzi et al. (1998)), **Publication** (filter by publication title/event), **Authors** (filter by author names), **Year** (filter by publication year), **Affiliations** (filter by affiliation organizations), **Affiliation Sites** (filter by affiliation cities and countries)[6]. Found papers always match *all* currently set filters. For each filter type multiple different filter items can be set; one could search for papers written jointly by people from different research institutes on a certain topic, for example. Matches of the statements filter and the keywords filter are highlighted in document snippets for each paper in the Results View and in the currently selected paper of the Document View.

Besides a header displaying the metadata of the currently selected paper (including the automatically extracted topics on the right), the Document View provides three subviews of the selected paper: (1) the *Document Content View* is a raw list of the sentences of the paper and provides different kinds of interaction with these sentences; (2) the *PDF View* shows the original PDF version of the paper; (3) the *Citations View* provides citation information includ-

[6]Affiliations have been added using the ACL Anthology Network data (Radev et al., 2009).

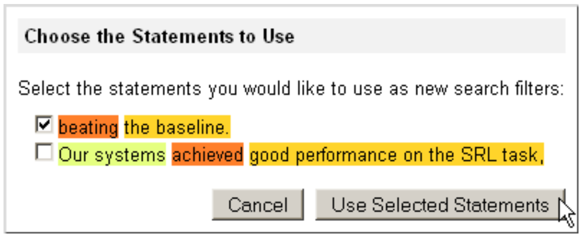
ing link to the ACL Anthology Network (Radev et al., 2009).

Figure 4 shows the search result for a query combining a statement ('obtain improvements'), a topic 'dependency parsing' and the publication year 2008. As can be seen in the Results View, six papers match these filters; sentences with semantically similar predicates and passive voice are found, too.

4.1 Semantic Search

The main feature which distinguishes the ACL Anthology Searchbench from other search applications for scientific papers is the semantic search in paper content. This enables the search for (semantic) statements in the paper content as opposed to searching for keywords in the plain text. Our use of the term "statement" is loosely along the lines of the same term used in logic. Very simple sentences often bear a single statement only, while more complex sentences (especially when having multiple clauses) contain multiple statements. Each of the semantic tuples extracted from the papers of the ACL Anthology (cf. Section 3) corresponds to a statement.

The Statements filter is responsible for the semantic search. Statements used in filters may be underspecified, e.g., one may search for statements with a certain semantic subject but with arbitrary semantic predicates and objects. There are two ways in which a new statement filter can be set: (1) entering a statement manually; (2) clicking a sentence in the Document Content View and choosing the statements of this sentence that shall be set as new statement filters (cf. Figure 5), i.e. it is possible to formulate and refine queries 'by example'.


Choose the Statements to Use

Select the statements you would like to use as new search filters:

☑ beating the baseline.
☐ Our systems achieved good performance on the SRL task,

Figure 5: Dialog for choosing statements to be used as new filters (for sentence "Our systems achieved good performance on the SRL task, easily beating the baseline.").

Throughout the user interface, no distinction is made between the different kinds of semantic objects and adjuncts so as to make it easy also for non-linguists to use the search and to be more robust against bad analyses of the parser. Therefore, the different semantic parts of a statement are highlighted in three different colors only, depending on whether a part is the semantic subject, the semantic predicate or anything else (object/adjunct).

In order to disengage even further from the concrete wording and make the semantic search even more 'meaning-based', we additionally search for synonyms of the semantic predicates in statement filters. These synonyms have been computed as an intersection of the most frequent verbs (semantic predicates) in the anthology corpus with WordNet synsets (Fellbaum, 1998), the main reason being reduction of the number of meanings irrelevant for the domain. This relatively simple approach could of course be improved, e.g. by active learning from user clicks in search results etc.

5 Summary and Outlook

We have described the ACL Anthology Searchbench, a novel search application for scientific digital libraries. The system is fully implemented and indexes 7,500 papers of the 8,200 parsed ones. For the other 700, bibliographic metadata was missing. These and the remaining 10,000 papers are currently being processed and will be added to the search index. The goal of the Searchbench is both to serve as a showcase for benefits and improvement of NLP for text search and at the same time provide a useful tool for researchers in Computational Linguistics. We believe that the tool by now already supports targeted search in a large collection of digital research papers better than standard web search engines. An evaluation comparing Searchbench query results with web search is in progress.

Optionally, the Searchbench runs in a linguistic debug mode providing NLP output a typical user would not need. These analyses are accessible from a context menu on each sentence (cf. Figure 4). Both a tabular view of the semantic tuples of a sentence (cf. Figure 3) and different kinds of information related to the parsing of the sentence (including the MRS and a parse tree) can be displayed.

Future work, for which we are urgently seeking funding, could include integration of further

NLP-based features such as coreference resolution or question answering, as well as citation classification and graphical navigation along the ideas in Schäfer and Kasterka (2010).

Acknowledgments

We are indebted to Peter Adolphs, Bart Cramer, Dan Flickinger, Stephan Oepen, Yi Zhang for their support with ERG and PET extensions such as chart mapping and chart pruning. Melanie Reiplinger, Benjamin Weitz and Leonie Grön helped with preprocessing. We also thank the anonymous reviewers for their encouraging comments. The work described in this paper has been carried out in the context of the project TAKE (Technologies for Advanced Knowledge Extraction), funded under contract 01IW08003 by the German Federal Ministry of Education and Research, and in the context of the world-wide DELPH-IN consortium.

References

Peter Adolphs, Stephan Oepen, Ulrich Callmeier, Berthold Crysmann, Daniel Flickinger, and Bernd Kiefer. 2008. Some fine points of hybrid natural language parsing. In *Proceedings of LREC-2008*, pages 1380–1387, Marrakesh, Morocco.

Steven Bird, Robert Dale, Bonnie Dorr, Bryan Gibson, Mark Joseph, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir Radev, and Yee Fan Tan. 2008. The ACL anthology reference corpus: A reference dataset for bibliographic research. In *Proceedings of LREC-2008*, pages 1755–1759, Marrakesh, Morocco.

Torsten Brants. 2000. TnT – a statistical part-of-speech tagger. In *Proc. of ANLP*, pages 224–231, Seattle, WA.

Ulrich Callmeier. 2000. PET – A platform for experimentation with efficient HPSG processing techniques. *Natural Language Engineering*, 6(1):99–108.

Ann Copestake, Dan Flickinger, Ivan A. Sag, and Carl Pollard. 2005. Minimal recursion semantics: an introduction. *Research on Language and Computation*, 3(2–3):281–332.

Ann Copestake. 2002. *Implementing Typed Feature Structure Grammars*. CSLI publications, Stanford.

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

Bart Cramer and Yi Zhang. 2010. Constraining robust constructions for broad-coverage parsing with precision grammars. In *Proceedings of COLING-2010*, pages 223–231, Beijing, China.

Witold Drożdżyński, Hans-Ulrich Krieger, Jakub Piskorski, Ulrich Schäfer, and Feiyu Xu. 2004. Shallow processing with unification and typed feature structures – foundations and applications. *Künstliche Intelligenz*, 2004(1):17–23.

Christiane Fellbaum, editor. 1998. *WordNet, An Electronic Lexical Database*. MIT Press.

Dan Flickinger, Stephan Oepen, and Gisle Ytrestøl. 2010. WikiWoods: Syntacto-semantic annotation for English Wikipedia. In *Proceedings of LREC-2010*, pages 1665–1671.

Dan Flickinger. 2002. On building a more efficient grammar by exploiting types. In Dan Flickinger, Stephan Oepen, Hans Uszkoreit, and Jun'ichi Tsujii, editors, *Collaborative Language Engineering. A Case Study in Efficient Grammar-based Processing*, pages 1–17. CSLI Publications, Stanford, CA.

Katerina T. Frantzi, Sophia Ananiadou, and Jun'ichi Tsujii. 1998. The C-value/NC-value method of automatic recognition for multi-word terms. In *Proceedings of ECDL*, pages 585–604.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English. The Penn Treebank. *Computational Linguistics*, 19:313–330.

Takashi Ninomiya, Yoshimasa Tsuruoka, Yusuke Miyao, Kenjiro Taura, and Jun'ichi Tsujii. 2006. Fast and scalable HPSG parsing. *Traitement automatique des langues (TAL)*, 46(2).

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

Dragomir R. Radev, Pradeep Muthukrishnan, and Vahed Qazvinian. 2009. The ACL anthology network corpus. In *Proceedings of the ACL-2009 Workshop on Natural Language Processing and Information Retrieval for Digital Libraries*, Singapore.

Ulrich Schäfer and Uwe Kasterka. 2010. Scientific authoring support: A tool to navigate in typed citation graphs. In *Proceedings of the NAACL-HLT 2010 Workshop on Computational Linguistics and Writing*, pages 7–14, Los Angeles, CA.

Ulrich Schäfer and Bernd Kiefer. 2011. Advances in deep parsing of scholarly paper content. In Raffaella Bernardi, Sally Chambers, Björn Gottfried, Frédérique Segond, and Ilya Zaihrayeu, editors, *Advanced Language Technologies for Digital Libraries*, LNCS Hot Topics Series. Springer. to appear.

Ulrich Schäfer. 2006. Middleware for creating and combining multi-dimensional NLP markup. In *Proceedings of the EACL-2006 Workshop on Multi-dimensional Markup in Natural Language Processing*, pages 81–84, Trento, Italy.