Extracting and Querying Relations in Scientific Papers on Language Technology

Ulrich Schäfer, Hans Uszkoreit, Christian Federmann, Torsten Marek, Yajing Zhang

German Research Center for Artificial Intelligence (DFKI), Language Technology Lab
Campus D 3 1, Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany
email: {ulrich.schaefer,hans.uszkoreit,christian.federmann,torsten.marek,yajing.zhang}@dfki.de

Abstract

We describe methods for extracting interesting factual relations from scientific texts in computational linguistics and language technology taken from the ACL Anthology. We use a hybrid NLP architecture with shallow preprocessing for increased robustness and domain-specific, ontology-based named entity recognition, followed by a deep HPSG parser running the English Resource Grammar (ERG). The extracted relations in the MRS (minimal recursion semantics) format are simplified and generalized using WordNet. The resulting 'quriples' are stored in a database from where they can be retrieved (again using abstraction methods) by relation-based search. The query interface is embedded in a web browser-based application we call the Scientist's Workbench. It supports researchers in editing and online-searching scientific papers.

1. Introduction

Research in the HyLaP project (in particular here the sub-project HyLaP-AM for the research on a personal digital associative memory) focuses on exploring hybrid (e.g. deep and shallow) methods to develop a framework for building a densely interlinked, associative memory on the basis of email and documents on the PC or laptop of a user.

The memory is structured and organized with the help of ontologies and taxonomies that can also support the user in querying and searching the content in an appropriate way. Statistical classification methods as well as NLP analysis tools are used to build, analyze and structure the document space. Automatic typed hyperlinking is employed for interlinking documents, emails, calendar entries and address books.

Access to the associative memory is provided by the Associative Information Access and Management Application (AIAMA). This is an electronic workbench for a scientist working in the field of language technology and computational linguistics, but potentially also in other domains, e.g. genetics or biotech.

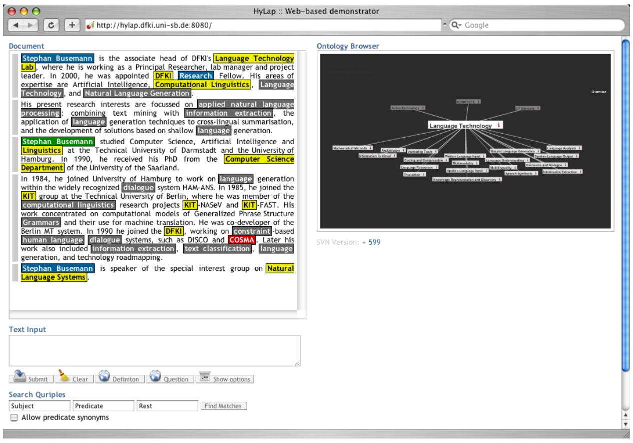
The application supports the researcher in answering questions and browsing and searching his or her collected emails, papers, presentations, address book and calendar items, but also support him or her in editing existing or new documents or emails with intelligent help and browsable content.

Thus, the application supports viewing, editing and browsing various documents hyperlinked by assistance of the associative memory and predefined ontologies, visualizing ontologies and taxonomies, and typing in questions to be answered by the system.

An application architecture has been designed where different workbench elements such as editor, ontology visualization and browser can be integrated flexibly, and extended or exchanged in the future.

Furthermore, an ontology browser and an HTML renderer for search results are embedded as well. The first prototype only used a named entity recognition system with resources augmented using a tool for recognizing instances and concepts of a given ontology.

This approach has now been enhanced by an interface to hybrid NLP for flexible content analysis and offline relation extraction. The currently used basis for the memory component is a relational database combined with an OWL ontology.



Stephan Busemann is the associate head of DFKI's Language Technology Lab, where he is working as a Principal Researcher, lab manager and project leader. In 2000, he was appointed DFKI Research Fellow. His areas of expertise are Artificial Intelligence, Computational Linguistics, Language Technology, and Natural Language Generation.

His present research interests are focussed on applied natural language processing: combining text mining with information extraction, the application of language generation techniques to cross-lingual summarisation, and the development of solutions based on shallow language generation.

Stephan Busemann studied Computer Science, Artificial Intelligence and Linguistics at the Technical University of Darmstadt and the University of Hamburg. In 1990, he received his PhD from the Computer Science Department of the University of the Saarland.

In 1984, he joined University of Hamburg to work on language generation within the widely recognized dialogue system HAM-ANS. In 1985, he joined the KIT group at the Technical University of Berlin, where he was member of the computational linguistics research projects KIT-NASeV and KIT-FAST. His work concentrated on computational models of Generalized Phrase Structure Grammars and their use for machine translation. He was co-developer of the Berlin MT system. In 1990 he joined the DFKI, working on constraint-based human language dialogue systems, such as DISCO and COSMA. Later his work also included information extraction, text classification, language generation, and technology roadmapping.

Stephan Busemann is speaker of the special interest group on Natural Language Systems.

Figure 1: Scientist's Workbench in HyLaP

As part of the application scenario, intelligent search based on factual relations extracted from parsed scientific paper texts is provided which we will describe in this paper. Because precision is in focus when facts are to be found in research papers, deep linguistic analysis of the paper contents is performed, assisted by shallow resources and tools for incorporating domain knowledge and ontology information, and for improving robustness of deep processing.

In this paper, we concentrate on two aspects of this rather complex enterprise. First, we explain how we extract the facts (relations) from the papers, and second, we describe the GUI application that has been built and that includes access to the extracted information.

We start with a discussion of how to access the content of scientific papers available as PDF files in Section 2. In Section 3., we describe the hybrid parsing approach to get semantic structures from natural language sentences. In Section 4., we explain how we compute the core relations from the linguistics-oriented output results of the previous step. We present the GUI application Scientist's Workbench and the underlying system architecture in Section 5. Finally, we briefly discuss related work, conclude and give an outlook to future work.

2. PDF Extraction

In order to get documents related to the application domain, we downloaded the contents of the ACL Anthology [1]. The ACL Anthology is a collection of scientific articles from international conferences, workshops and journals on computational linguistics and language technology. We acquired all papers from the years 2002–2007, resulting in a collection of 4845 PDF documents, along with bibliographical information in BibTeX format, if available.

Due to the design nature of PDF, which is a rendering-oriented format, extracting the actual text from a document is a non-trivial process. Existing end-to-end solutions like the tools included in the Poppler PDF rendering library[2] may, depending on the input document, extract the text in the wrong order or displace headers and also make it difficult to distinguish footnotes, table or figure captions and equations from body text.

In order to get the best quality possible given the challenges, we used PDFBox[3], a Java library for parsing and creating PDF documents. We modified the included proof-of-concept text extractor to also write out information about positions and font sizes of text blocks and improved the handling of text in two-column layouts commonly found in scientific articles.

The output we obtained this way still contained errors like wrongly-ordered paragraphs and non-body text parts, but the added information made it possible to reliably fix these errors. We wrote a program to create from this raw input a document that more closely resembles the semantic structure of the original 'underlying' article, i.e. sections and paragraphs rather than lines, columns and pages.

With this process, we tried to remove all effects of typesetting, pagination and hyphenation. Using positional information, it was also possible to recreate the 'reading' order of the text from the order of rendering commands in the PDF stream.

The result of the extraction process is a structured document (XML). It contains metadata about the article, which is obtained from either the bibliographical information available in the ACL anthology or guessed from the document, the abstract, the body text, the conclusion and all table and figure captions in a logical structure for further processing.

The quality of the text extraction largely depends on the tool or succession of tools used to create the PDF document from the input format it was written in. The highest quality usually was reached with documents that were created using the PDF renderer of the LaTeX document preparation system.

Documents that were created using a DVI → PS → PDF conversion process proved to be more problematic, as did documents created with the PDF generator of newer Microsoft Word versions.

In the end, we were able to extract the full text of 4429 documents, from the original 4845. Documents that could not be extracted were created using other means, like logical PDF printers or contained font encodings unknown to the PDFBox library.

These documents could theoretically be handled by applying OCR (Optical Character Recognition) programs to the rendered page images, which would also be the only solution for historical papers, that usually only contain bitmaps rather than vector drawing commands.

However, due to the already high turnout of the straightforward extraction process (91% of all documents could be processed) that provided us with enough documents for the next steps, we decided not to employ OCR techniques.

Another reason for this decision was the additional errors that are introduced by OCR algorithms and the fact that human post-correction of such a large number of documents was not feasible in the scope of this work.

3. Parsing Science

The texts from the PDF papers (so far only their abstracts, parsing the full papers is work in progress) have been converted to plain text and split with a sentence splitter. Then, they were parsed using the hybrid NLP platform Heart of Gold (Schäfer, 2007). Heart of Gold is an XML-based middleware for the integration of multilingual shallow and deep natural language processing components.

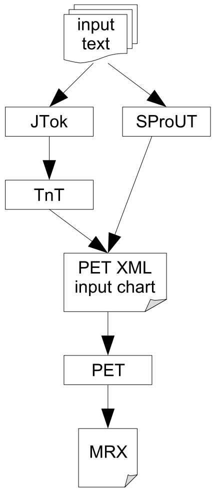


Figure 2: Heart of Gold workflow for hybrid parsing

The employed Heart of Gold configuration instance starts with a tokenizer, the shallow part-of-speech tagger TNT (Brants, 2000) and the named entity recognizer SProUT (Drożdżyński et al., 2004). These components help to identify and classify open class words such as person names, events (e.g. conferences) or locations.

Furthermore, the (trigram-based) tagger helps to guess part-of-speech tags of words unknown to the deep lexicon. For

[1] http://acl.ldc.upenn.edu/, (Bird et al., 2008)

[2] http://poppler.freedesktop.org

[3] http://www.pdfbox.org

both (unknown words and named entities), generic lexicon entries are generated in the deep parser. Using an XML format, the shallow preprocessing results were then (within Heart of Gold; cf. Fig 2) passed to the high-speed HPSG parser PET (Callmeier, 2000) running the open source broad-coverage grammar ERG (Copestake and Flickinger, 2000).

Details of the general approach and further configurations as well as evaluations of the benefits of hybrid parsing are described in (Schäfer, 2007).

In contrast to shallow parsing systems, the ERG not only handles detailed syntactic analyses of phrases, compounds, coordination, negation and other linguistic phenomena that are important for extracting relations, but also generates a formal semantic representation of the meaning of the input sentence.

Ambiguities resulting in multiple readings per input sentence were ranked using a statistical model based on the Redwoods treebank (Oepen et al., 2002). We got full parses for 62.5% of the 17 716 abstract sentences in HoG. The average sentence length was 18.9 words for the parsed, 27.05 for the unparsed, and 21.95 for all.

The deep parser returns a semantic analysis in the MRS representation format (Minimal Recursion Semantics; (Copestake et al., 2005) that is, in its robust variant, specifically suited to represent hybrid, i.e. deep and comparably underspecified shallow NLP semantics. A sample MRS as produced by ERG in Heart of Gold is shown in Figure 4.

In case a full parse is not possible, longest fragments can be used instead to obtain a maximal number of analyzed sentences. Another solution would be to use an underspecified analysis obtained by a purely shallow parser as fall-back result. However, as a quick solution and because we are interested in maximizing precision, we currently simply omit sentences that cannot be parsed entirely.

As part of the shallow preprocessing pipeline, named entity recognition is performed to recognize names and domain-relevant terms. To improve recognition in the domain of science on language technology and computational linguistics, we enriched the lingware resources of the generic named entity recognizer SProUT (Drożdżyński et al., 2004) by instance and concept information from an existing domain ontology.

We used the LT World ontology (Uszkoreit et al., 2003), containing about 1200 concepts and approx. 20 000 instances such as conferences, persons, projects, products, companies and organizations, and extended SProUT with LT World contents by applying OntoNERdIE.

OntoNERdIE (Schäfer, 2006) is an offline procedure that maps OWL/RDF-encoded ontologies with large, dynamically maintained instance data to named entity recognition (NER) and information extraction (IE) engine resources, preserving hierarchical concept information and links back to the ontology concepts and instances.

The named entities enriched with ontology information are then employed in the robustness-oriented, hybrid deep-shallow architecture that combines domain-specific shallow NER and deep, domain-independent HPSG parsing for generating a semantics representation of the meaning of parsed sentences as described above.

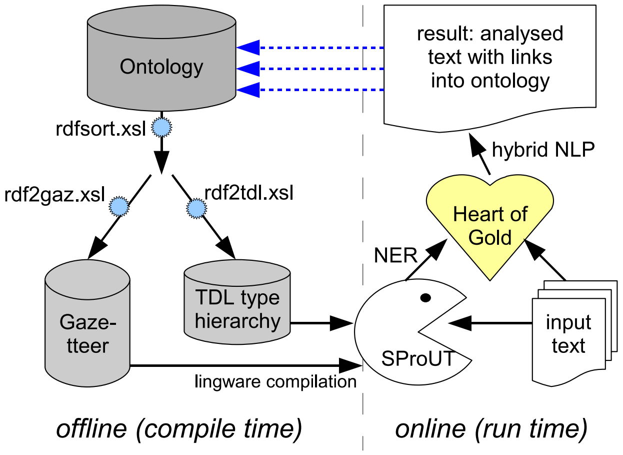


Figure 3: Enriching hybrid parsing with domain-specific ontology information (OntoNERdIE)

The MRS representations resulting from hybrid parsing, however, are 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 necessary before storing the semantic structures with links to the original text in a database.

4. From MRS to Quriples: Relation Extraction and Abstraction

The produced MRSes contain relations (EPs) with names e.g. for verbs that contain the stem of the recognized verb in the lexicon, such as *show_rel*. We can extract these relations and represent them in 'quriples'.

4.1. Quriple generation

Instead of using simple triples, we decided to use *quriples* to represent all arguments in the sentence. These quriples are query-oriented quintuples including subject (SUB), predicate (PRD), direct object (DOBJ), other complement (OCMP) and adjunct (ADJU), where OCMP includes indirect object, preposition complement, etc. ADJU contains all other information which does not belong to any of other four parts.

Due to semantic ambiguity, the parser may return more than one reading per sentence. Currently up to three readings are provided (the most probable ones according to the trained parse ranking model), and quriples are generated for each reading respectively. Multiple readings may lead to the same quriple structure, in which case only a single one is stored in the database.

We illustrate the extraction procedure using the following example. Its corresponding MRS representation produced by Heart of Gold is shown in Figure 4.

(1) *We evaluate the efficiency and performance empirically against the corpus.*

Generally speaking, the extraction procedure starts with the predicate, and all its arguments are extracted one by one using depth-first until all EPs are exhausted. In practice, we

$$
\left[\begin{array}{ll}
\text{TEXT} & \text{We evaluate the efficiency and performance empirically against the corpus.}\\
\text{TOP} & h1\\
\text{RELS} & \left\{\begin{array}{l}
\left[\begin{array}{ll}\multicolumn{2}{l}{\textit{prop-or-ques\_m\_rel}}\\ \text{LBL} & h1\\ \text{ARG0} & e2\\ \text{MARG} & h3\end{array}\right]
\left[\begin{array}{ll}\multicolumn{2}{l}{\textit{pron\_rel}}\\ \text{LBL} & h6\\ \text{ARG0} & x7\end{array}\right]
\left[\begin{array}{ll}\multicolumn{2}{l}{\textit{pronoun\_q\_rel}}\\ \text{LBL} & h8\\ \text{ARG0} & x7\\ \text{RSTR} & h9\\ \text{BODY} & h10\end{array}\right]
\left[\begin{array}{ll}\multicolumn{2}{l}{\textit{\_evaluate\_v}}\\ \text{LBL} & h11\\ \text{ARG0} & e2\ \text{tense=present}\\ \text{ARG1} & x7_{\substack{\text{num=pl}\\ \text{pers=1}}}\\ \text{ARG2} & x12\ \text{pers=3}\end{array}\right]
\left[\begin{array}{ll}\multicolumn{2}{l}{\textit{\_the\_q}}\\ \text{LBL} & h13\\ \text{ARG0} & x12\\ \text{RSTR} & h14\\ \text{BODY} & h15\end{array}\right]\\
\left[\begin{array}{ll}\multicolumn{2}{l}{\textit{udef\_q\_rel}}\\ \text{LBL} & h16\\ \text{ARG0} & x18\\ \text{RSTR} & h17\\ \text{BODY} & h19\end{array}\right]
\left[\begin{array}{ll}\multicolumn{2}{l}{\textit{\_efficiency\_n}}\\ \text{LBL} & h20\\ \text{ARG0} & x18\end{array}\right]
\left[\begin{array}{ll}\multicolumn{2}{l}{\textit{udef\_q\_rel}}\\ \text{LBL} & h21\\ \text{ARG0} & x23\\ \text{RSTR} & h22\\ \text{BODY} & h24\end{array}\right]
\left[\begin{array}{ll}\multicolumn{2}{l}{\textit{\_and\_c}}\\ \text{LBL} & h25\\ \text{ARG0} & x12\ \text{pers=3}\\ \text{L-INDEX} & x18_{\substack{\text{gender=n}\\ \text{num=sg}\\ \text{pers=3}}}\\ \text{R-INDEX} & x23_{\substack{\text{num=sg}\\ \text{pers=3}}}\end{array}\right]
\left[\begin{array}{ll}\multicolumn{2}{l}{\textit{\_performance\_n}}\\ \text{LBL} & h28\\ \text{ARG0} & x23\end{array}\right]\\
\left[\begin{array}{ll}\multicolumn{2}{l}{\textit{generic\_adv\_rel}}\\ \text{LBL} & h10001\\ \text{ARG0} & e29\ \text{tense=u}\\ \text{ARG1} & e2\end{array}\right]
\left[\begin{array}{ll}\multicolumn{2}{l}{\textit{\_against\_p}}\\ \text{LBL} & h10002\\ \text{ARG0} & e31\ \text{tense=u}\\ \text{ARG1} & e2\\ \text{ARG2} & x30_{\substack{\text{num=sg}\\ \text{pers=3}}}\end{array}\right]
\left[\begin{array}{ll}\multicolumn{2}{l}{\textit{\_the\_q}}\\ \text{LBL} & h32\\ \text{ARG0} & x30\\ \text{RSTR} & h33\\ \text{BODY} & h34\end{array}\right]
\left[\begin{array}{ll}\multicolumn{2}{l}{\textit{\_corpus\_n}}\\ \text{LBL} & h35\\ \text{ARG0} & x30\end{array}\right]
\end{array}\right\}\\
\text{HCONS} & \{h3 \text{ qeq } h11, h9 \text{ qeq } h6, h14 \text{ qeq } h25, h17 \text{ qeq } h20, h22 \text{ qeq } h28, h33 \text{ qeq } h35\}\\
\text{ING} & \{h11 \text{ ing } h10001, h11 \text{ ing } h10002\}
\end{array}\right]
$$

Figure 4: MRS representation (robust variant) as produced by ERG in Heart of Gold

start with the top handle *h1* which in this case contains the *prop-or-ques_m_rel* relation (cf. Figure 4). Its first argument ARG0 indicates the predicate (*evaluate*) which has 3 arguments. ARG0 normally refers to the EP itself, and the rest ARG1 and ARG2 are the arguments taken by the predicate. Therefore, a transitive verb has altogether three ARGs and a ditransitve verb has four ARGs.

The predicate arguments are processed sequentially. ARG1 of the predicate i.e. *x7* has two relations: *pron_rel* and *pronoun_q_rel*, both of which are realized as *we* in the sentence. ARG2 of the predicate i.e. *x12* in this case has *_and_c* relation where two EPs shown in L-INDEX and R-INDEX respectively are coordinated. The extraction of L-INDEX results in two lexemes: *its* and *efficiency*, and the extraction of R-INDEX results in *performance*.

Finally the adjunct, indicated by the message relation MARG in this case, is extracted. Following *qeq* (equality modulo quantifiers) constraint, handle *h3* is *qeq* to label *h11*. As the implicit conjunction indicates (marked by ING relation), *h11* conjuncts with label *h10005*, the *_against_p* relation, introducing the adjunct *against the corpus*. In this way, all EPs can be extracted from the RMRS representation, and the quriple generated are shown in Table 1.

SUB	We
PRD	evaluate
DOBJ	the, efficiency, and, performance
OCMP	-
ADJU	against, the, corpus

Table 1: Quriple generated for Example 1

Although ARG0 of the top handle refers to the predicate of the sentence, it is not necessarily the main verb of the sentence. We discuss two more cases: conjunction and passive.

Conjunction: In Example 1, the conjunction relation connects two noun phrases, both of them being DOBJ, therefore, no new quriple is necessary. However, we decided to distinguish cases where conjunction connects two sentences or verb phrases. In such cases, quriples are generated for each part respectively. Example 2 shows an *AND* relation, however, conjunction relations may be realized in different lexemes, e.g. *and, but, or, as well as*, etc.

(2) *The system automatically extracts pairs of syntactic units from a text and assigns a semantic relation to each pair.*

For this example, two quriples are generated separately with their own PRD, DOBJ and OCMP (cf. Table 2):

SUB	The, system
PRD	extract
DOBJ	pairs, syntactic, units
OCMP	from, a, text
ADJU	automatically
SUB	The, system
PRD	assign
DOBJ	a, semantic, relation
OCMP	to, each, pair
ADJU	automatically

Table 2: Two quriples generated for the conjunction in Example 2

In MRS, a conjunction with more than two arguments is represented in the coordination relation with an embedded structure, i.e. ARG1 first coordinates with ARG2, where

ARG2 includes all the rest arguments. ARG2 then coordinates with ARG3 where ARG1 and ARG2 are excluded. Currently, we only deal with conjunctions on the top level, the embedded structure is not touched.

Passive: For passive sentences, ARG1 refers to the semantic object and ARG2 refers to the subject. The past participle, but not *be*, is extracted as PRD (cf. Table3).

(3) *Unseen input was classified by trained neural networks with varying error rates depending on corpus type.*

SUB	trained, neural, networks, with, varying, error, rates, depending, corpus, type
PRD	classify
DOBJ	unseen, input
OCMP	-
ADJU	-

Table 3: Quriple generated for the passive sentence in Example 3

All fully parsed sentences were processed and quriples were generated. Currently a relational database is used to save the quriples. It should be pointed out here that the same extraction steps that are used in the offline extraction process can also be used for processing natural language queries (parsed using the same hybrid pipeline as described before) for efficient and robust online search in the built relation database.

However, the query interface implemented so far only consists of a form-based search with input fields for subject, predicate and rest (objects etc.). Tables 5 and 6 contain the most frequent subjects and predicates as they occur in the extracted corpus.

4.2. Integration of WordNet

Since the same relation can be very often realized differently using various lexemes, e.g. *present(x,y)* can be realized in *presents/shows/demonstrates* ..., therefore, the pure string match using the predicate of a sentence is far from enough. In order to improve the robustness of our system, WordNet (Miller et al., 1993) is integrated to search for the synsets of predicates.

As an explicit synset class is not defined for verbs in WordNet, for approximation we retrieve verbs from all senses of the current predicate. In this way synonyms of more than 900 predicates are retrieved, and these synonyms are saved in the database as an extra table. Table 4 shows somes PRDs and their synonyms.

The integration of WordNet enables the user to conduct the search process not only based on the PRDs themselves, but also take the synonyms of PRDs into consideration. The second optionality can be activated by selecting 'allow predicate synonyms' on the workbench GUI (cf. Figure1).

5. Implementation of the Scientist's Workbench

The Scientist's Workbench has been implemented as a lightweight web application that can be used within any modern web browser. This allowed us to quickly connect the different modules such as the quriple store and the underlying server interfaces without having to build a complete editor application. Furthermore it does not require anything else than a web browser to use the AIAMA GUI which makes the whole approach very flexible.

As the usage of a browser-based application introduced some constraints and restrictions, we have decided to split the Scientist's Workbench into two software layers. First, we provide an HTML based *GUI component* that runs in the client's browser. Second, we have created a so called *broker server* that coordinates and controls the underlying AIAMA services. The broker server is implemented using the Python programming language. It uses SQL and XML-RPC connections to communicate with the different AIAMA servers and AJAX methods to update the GUI component.

PREDICATE	SYNONYM
demonstrate	demo, prove, establish, show, manifest, exhibit, present
evaluate	measure, judge, value, assess, valuate
assign	put, attribute, specify
result in	result, leave, lead
search for	look, explore, search, research, seek
find out	check, determine, discover, learn, pick up, see, find

Table 4: Predicates and their synonyms extracted from WordNet

SUBJECT	#	SUBJECT	#
We	2897	Our approach	28
This paper	637	Our results	26
It	171	I	26
paper	83	the algorithm	24
The system	78	The model	23
The paper	75	They	22
the results	40	Our system	22
Experimental results	33	Our experiments	21
the method	30	The parser	19
Our method	29	This approach	19

Table 5: Most frequent subjects and number of occurrences

5.1. System Overview

The following figure gives a systematic overview on the design of the AIAMA GUI application. Note that XML-RPC connections are used to connect all services to the broker server, while quriple information is retrieved from an RDBMS using SQL.

5.2. Information flow in the AIAMA GUI

When a user submits new (edited) text to the system, this text will be sent to the broker server which will perform a first shallow analysis of the given input. This process annotates every term in the input text that is known in the un-

PREDICATE	#	PREDICATE	#
present	799	investigate	140
be	681	provide	139
show	610	introduce	134
describe	517	achieve	121
propose	392	report	110
use	279	explore	95
can	205	have	88
evaluate	166	compare	84
discuss	150	find	83
demonstrate	146	develop	74

Table 6: Most frequent predicates and number of occurrences

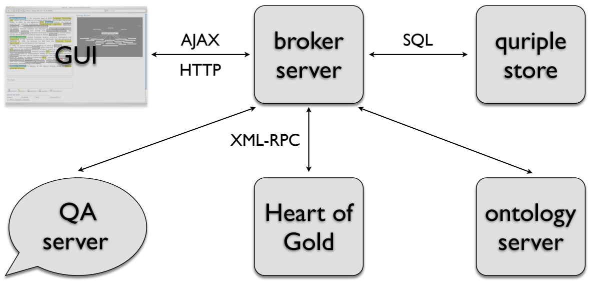


Figure 5: AIAMA GUI system overview

derlying ontology. As no ontology can ever be complete, we also integrated a Heart of Gold service to perform a SProUT named entity analysis of the text data. This ensures that even terms that are not contained within the ontology can be found and annotated. In case of person names, even variants such as 'Dr. Smith' can be analyzed.

Next to ontology concepts, the broker server will also detect any embedded natural language question that has been entered in the GUI. To ease question detection we add some markup around questions in the GUI component which the broker server tries to find inside the given input. Whenever a question is found the broker server sends a corresponding query to an open domain QA system which then tries to find an answer, e.g. for definition questions. See (Figueroa and Neumann, 2007) for more information on the design and implementation of the QA server. Later, also natural language questions to the quriple store will be implemented.

After the edited text has been successfully analyzed, it is enriched by the detected annotations and sent back to the GUI. Different colors are used to distinguish between the possible annotation concepts such as person name, project, etc. Figure 1 shows the AIAMA GUI after some paragraphs have been entered and successfully analysed.

5.3. Search Results

All annotations that have been returned from the broker server can be clicked by the user and reveal more information about the annotated term. These *search results* may contain the following information:

- emails that a detected person has sent or received
- papers that a person has authored
- meetings that a person has scheduled
- ontology concepts

For papers we show both associated metadata such as title, authors, etc. and also the quriples that have been found within them. If the user clicks on such a quriple, the original PDF document will be opened in another window and the corresponding text lines will be highlighted in the original layout.

5.4. Ontology Navigation

We use a Flash-based visualization of ontology concepts. The *ontology browser* shows all concept classes that are contained in the ontology and allows to navigate through the different concepts. It also allows to view the list of instances for a given concept. Whenever the user clicks on an ontology concept within the AIAMA GUI, the ontology browser will automatically show this concept and the surrounding classes.

5.5. Relation Querying

The AIAMA GUI includes an interface to directly query quriples within the parsed papers. *Search Quriples* allows to specify a subject, a predicate and some 'rest' which will then be used to filter out any matching quriples. The user may choose to use Wordnet synsets to allow predicate synonyms when defining the query.

Once the user has defined a query it is sent to the broker server which looks up matching quriples from the quriple store and returns them to the GUI. If the user clicks on any of these results, this will open the original PDF document and highlight the respective sentence.

In order to enable relation queries, we stored all extracted relations in a relational database, along with information where to find them, i.e. the source document, page and position the relation was extracted from. Since for the first system, the number of extracted relations was rather small (in the end, there were around 10 000 distinct relations in the database), a simplistic search using the built-in string matching functions of the SQL server turned out out to be fast enough.

In order to query the relations, users need to specify any of the subject, the predicate or 'other' (which will match either the objects or the additional relation parts extracted from the sentences). The predicate is always matched completely, with the option of expanding the query with synonyms taken from WordNet. For the subject and the other parts, a match is found if any substring matches the pattern specified by the user.

In the results view, the user can choose to view the original document (i.e. the original article in PDF form) and the sentence this relation was extracted from will also be highlighted, so that users immediately find the source instead of having to search through a probably lengthy document themselves.

6. Related Work

Using HPSG combined with shallow domain-specific modeling for high-precision analysis of scientific texts is an emerging research area. Another ERG-based approach to relation and information extraction from scientific texts in the DELPH-IN context[4] is SciBorg (Rupp et al., 2007) (chemistry research papers).

(Sætre et al., 2008) use shallow dependency structure and results from HPSG parsing for extracting protein-protein interactions (PPI) from research papers. The same group has also worked on medical texts: MEDIE[5] is a semantic search engine to retrieve biomedical correlations from MEDLINE articles.

What distinguishes our approach from those, besides concentration on a different scientific field, is the focus on and use of ontology information as integrated part of linguistic analysis, and the interactive editor user interface (Scientist's Workbench application).

7. Summary and Outlook

We have described methods to extract interesting factual relations from scientific texts in the computational linguistics and language technology fields taken from the ACL Anthology.

The approach is currently still work in progess and thus not yet fully evaluated.

The coverage of 62.5% full parses of the abstract sentences without any specific adaptations to the domain except for the recognition of instances from the LT World domain is very good and promising. We will try to further improve this result by looking carefully at possible errors in the deep-shallow interfaces and HPSG grammar. A promising fallback solution already investigated in the DeepThought project is using (robust) MRS analyses from a shallow parser as is done in SciBorg (Rupp et al., 2007). Also, using fragmentary parsing results in cases where a sentence could not be analyzed entirely is ongoing research and surely will help to improve the overall coverage.

Future work may include a deeper investigation of adaptability to other ontologies and domains than described here, and extension of the mapping approach to additional relations supported by OWL.

Furthermore, various approaches exist to anaphora resolution that could be incorporated and help to improve coverage and quality of the extracted relations.

8. Acknowledgments

This work has been supported by a grant from the German Federal Ministry of Education and Research (FKZ 01 IW F02). We are grateful to Daniel Beck and Can Kayalı for their contributions to the implementation of AIAMA.

9. References

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*, Marrakech, Morocco.

Thorsten Brants. 2000. TnT - A Statistical Part-of-Speech Tagger. In *Proceedings of Eurospeech*, Rhodes, Greece.

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

Ann Copestake and Dan Flickinger. 2000. An open-source grammar development environment and broad-coverage English grammar using HPSG. In *Proceedings of the 2nd Conference on Language Resources and Evaluation (LREC-2000)*, pages 591–598, Athens, Greece.

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

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.

Alejandro Figueroa and Günter Neumann. 2007. A multilingual framework for searching definitions on web snippets. In *Proceedings of KI-2007, Osnabrück, Germany*.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. 1993. Five papers on WordNet. Technical report, Cognitive Science Laboratory, Princeton University.

Stephan Oepen, Dan Flickinger, Kristina Toutanova, and Christoper D. Manning. 2002. LinGO redwoods: A rich and dynamic treebank for HPSG. In *Proceedings of the Workshop on Treebanks and Linguistic Theories, September 20–21 (TLT02)*, Sozopol, Bulgaria.

CJ Rupp, Ann Copestake, Peter Corbett, and Ben Waldron. 2007. Integrating general-purpose and domain-specific components in the analysis of scientific text. In *Proceedings of the UK e-Science Programme All Hands Meeting 2007 (AHM2007)*, Nottingham, UK.

Rune Sætre, Sagae Kenji, and Jun'ichi Tsujii. 2008. Syntactic features for protein-protein interaction extraction. In Christopher J.O. Baker and Su Jian, editors, *Short Paper Proceedings of the 2nd International Symposium on Languages in Biology and Medicine (LBM 2007)*, pages 6.1–6.14, Singapore, 1. ISSN 1613-0073319.

Ulrich Schäfer. 2006. OntoNERdIE – mapping and linking ontologies to named entity recognition and information extraction resources. In *Proceedings of the 5th International Conference on Language Resources and Evaluation LREC-2006*, pages 1756–1761, Genoa, Italy, 5.

Ulrich Schäfer. 2007. *Integrating Deep and Shallow Natural Language Processing Components – Representations and Hybrid Architectures*. Ph.D. thesis, Faculty of Mathematics and Computer Science, Saarland University, Saarbrücken, Germany.

Hans Uszkoreit, Brigitte Jörg, and Gregor Erbach. 2003. An ontology-based knowledge portal for language technology. In *Proceedings of ENABLER/ELSNET Workshop*, Paris.

[4] DEep Linguistic Processing with HPSG Initiative; `http://www.delph-in.net`

[5] `http://www-tsujii.is.s.u-tokyo.ac.jp/medie/`