

An Architecture for Language Processing for Scientific Texts

Ann Copestake¹, Peter Corbett², Peter Murray-Rust², CJ Rupp¹,
Advait Siddharthan¹, Simone Teufel¹, Ben Waldron¹

[1] Computer Laboratory, University of Cambridge

[2] Unilever Centre for Molecular Informatics, University of Cambridge

Abstract

We describe the architecture for language processing adopted on the eScience project ‘Extracting the Science from Scientific Publications’ (nicknamed SciBorg). In this approach, papers from different sources are first processed to give a common XML format (SciXML). Language processing modules operate on the SciXML in an architecture that allows for (partially) parallel deep and shallow processing and for a flexible combination of domain-independent and domain-dependent techniques. Robust Minimal Recursion Semantics (RMRS) acts both as a language for representing the output of processing and as an integration language for combining different modules. Language processing produces RMRS markup represented as standoff annotation on the original SciXML. Information extraction (IE) of various types is defined as operating on RMRSs. Rhetorical analysis of the texts also partially depends on IE-like patterns and supports novel methods of information access.

1 Introduction

The eScience project ‘Extracting the Science from Scientific Publications’ (nicknamed SciBorg, www.sciborg.org.uk) aims to build dynamic, flexible and expandable natural language processing (NLP) infrastructure which will support applications in eScience. We hope to show that autonomous, adaptive methods, based on NLP techniques, can be used to mine the primary literature and other text to build an evolving knowledge base for eScience. Overall, the goals of SciBorg are:

1. To develop a markup language for natural language which will act as a platform for extraction of information.
2. To develop Information Extraction (IE) technology and core ontologies for use by publishers, researchers, readers, vendors, and regulatory organisations.
3. To model scientific argumentation and citation purpose in order to support novel modes of information access.
4. To demonstrate the applicability of this infrastructure in a real-world eScience environment.

The SciBorg project started in October 2005 and is a collaboration between the Computer Laboratory, the Unilever Centre for Molecular Informatics and the Cambridge eScience Centre, with support from the Royal Society of Chemistry, Nature

Publishing Group and the International Union of Crystallography. We are concentrating on Chemistry texts in particular, but we aim to develop techniques which are largely domain-independent with clear interfaces to domain-dependent processing. For instance, we are using some of the same tools as the Flyslip project (Hollingsworth et al., 2005; Vlachos and Gasperin, 2006), which concerns extraction of functional genomic information to aid FlyBase curation. We are also collaborating with the Citation Relations and Argumentative Zoning (CitRAZ) project, especially on the discourse processing aspects of the project.

The goal of this paper is to introduce and motivate the architecture which we have adopted for language processing within SciBorg and to describe some of the progress so far on implementing that architecture and the various language processing modules it comprises.

Characteristic features of our general approach to language processing are:

1. We are integrating, adapting and further developing general tools for language processing. We intend to avoid domain-specific solutions wherever possible, even if this leads to lower performance in the short term. The primary goal of the project is to improve the language technology.
2. We are incorporating relatively deep syntactic and compositional semantic processing. By ‘deep’,

we mean systems which use very precise and detailed grammars of natural languages to analyse and generate, especially approaches based on current linguistic theory (see §5).

3. We are developing a semantically-based representation language Robust Minimal Recursion Semantics (RMRS:Copestake (2003)) for integration of all levels of language processing and to provide a standardised output representation. RMRS is an application-independent representation which captures the information that comes from the syntax and morphology of natural language while having a sound logical basis compatible with Semantic Web standards. Unlike previous approaches to semantic representation, underspecified RMRSs can be built by shallow language processing, even part-of-speech (POS) taggers. See §4.

4. Integration with XML is built in to the architecture, as discussed in §3. We view language processing as providing standoff annotation expressed in RMRS-XML, with deeper language processing producing more detailed annotation.

5. As far as possible, all technology we develop is Open Source. Much of it will be developed in close collaboration with other groups.

Our overall aim is to produce an architecture that allows robust language processing even though it incorporates relatively non-robust methods, including deep parsing using hand-written grammars and lexicons. The architecture we have developed is not pipelined, since shallow processing can operate in parallel to deeper processing modules as well as providing input to them. We do not have space in this paper for detailed comparison with previous approaches, but note that this work draws on results from the Deep Thought project: Uszkoreit (2002) discusses the role of deep processing in detail.

In the next section, we outline the application tasks we intend to address in SciBorg. This is followed by a description of the overall architecture and the RMRS language. §5 and §6 provide a little more detail about the modules in the architecture, concentrating on architecture and integration details. §7 describes our approach to discourse analysis.

2 Application tasks

In this section, we outline three tasks that we intend to address in the project. As we will explain below, these tasks all depend on matching patterns specified in terms of RMRS. They can all be considered as forms of information extraction (IE), although we use that term very broadly. Most existing IE technology is based on relatively shallow processing of texts to directly instantiate domain-specific templates or databases. However, for each new type of information, a hand-crafted system or an exten-

sive manually-created training corpus is required. In contrast, we propose a layered architecture using an approach to IE that takes the RMRS markup, rather than text, as a starting point. Again, this follows Deep Thought, but only a limited investigation of the approach was attempted there.

Chemistry IE We are interested in extraction of specific types of chemistry knowledge from texts in order to build a database of key concepts fully automatically. For example, the following two sentences are typical of the part of an organic synthesis paper that describes experimental methods:

The reaction mixture was warmed to rt, whereat it was stirred overnight. The resultant mixture was kept at 0C for 0.5 h and then allowed to warm to rt over 1 h.

Organic synthesis is a sufficiently constrained domain that we expect to be able to develop a formal language which can capture the steps in procedures. We will then extract ‘recipes’ from papers and represent the content in this language. A particular challenge is extraction of temporal information to support the representation of the synthesis steps.

Ontology construction The second task involves semi-automatically extending ontologies of chemical concepts, expressed in OWL. Although some ontologies exist for chemistry and systematic chemical names also give some types of ontological information (see §6.2), resources are still relatively limited, so automatic extraction of ontological information is important. Our proposed approach to this is along the lines pioneered by Hearst (1992) and refined by other authors. For instance, from:

...the concise synthesis of naturally occurring alkaloids and other complex polycyclic azacycles.

we could derive an expression that conveyed the information an ‘alkaloid IS-A azacycle’.

```
<owl:Class rdf:ID="Alkaloid">
<rdfs:subClassOf
    rdf:resource="#Azacycle">
```

Ontologies considerably increase the flexibility of the chemistry IE, for instance by allowing for matches on generic terms for groups of chemicals (cf., the GATE project (gate.ac.uk) ‘Ontology Based Information Extraction/OBIE’).

Research markup The third application is geared towards humans browsing papers: the aim is to help them quickly see the most salient points and the interconnections between papers. The theoretical background to this is discussed in more detail in

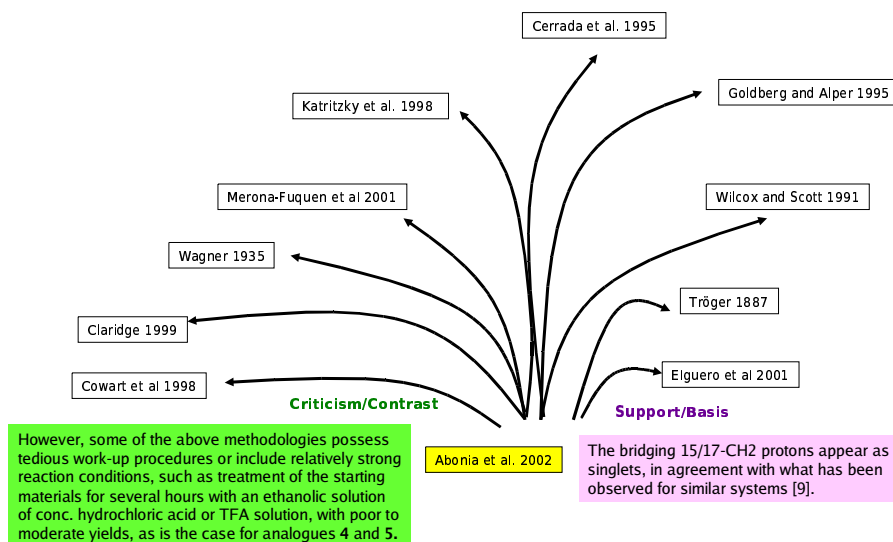


Figure 1: A rhetorical citation map

§7, but here we just illustrate its application to citation. Our aim is to provide a tool which will enable the reader to interpret the rhetorical status of a citation at a glance, as illustrated in Fig. 1. The example paper (Abonia et al., 2002) cites the other papers shown, but in different ways: Cowart et al (1998) is a paper with which it contrasts, while Elguero et al (2001) is cited in support of an observation. We also aim to extract and display the most important textual sentence about each citation, as illustrated in the figure.

The Chemistry Researchers' amanuensis By the end of the project, we will combine these three types of IE task in a research prototype of a 'chemistry researchers' amanuensis' which will allow us to investigate the utility of the technology in improving the way that chemists extract information from the literature. Our publisher partners are supplying us with large numbers of papers which will form a database with which to experiment with searches. The searches could combine the types of IE discussed above: for instance, a search might be specified so that the search term had to match the goal of a paper.

3 Overall architecture

Figure 2 is intended give an idea of the overall architecture we are adopting for language processing. The approach depends on the initial conversion of papers to a standard version of XML, SciXML, which is in use not just on SciBorg but on the FlySlip and CITRAZ projects as well (Rupp et al., 2006). For SciBorg, we have the XML source of papers, while for FlySlip and CitRAZ, the source is pdf. Following this conversion, we invoke the domain-specific and domain-independent language processing modules on the text. In all cases, the language processing adds standoff annotation to the SciXML base. Standoff annotation uses SAF (Wal-

dron and Copestake, 2006; Waldron et al., 2006). SAF allows ambiguity to be represented using a lattice, thus allowing the efficient representation of multiple results from modules.

Language processing depends on the domain-dependent OSCAR-3 module (see §6) to recognise compound names and to markup data regions in the texts. The three sentence level parsing modules shown here (described in more detail in §5) are the RASP POS tagger, the RASP parser and the ERG/PET deep parser. The language processing modules are not simply pipelined, since we rely on parallel deep and shallow processing for robustness. Apart from the sentence splitter and tokenisers, all modules shown output RMRS. Output from shallower processing can be used by deeper processors, but might also be treated as contributing to the final output RMRS, with RMRSs for particular stretches of text being selected/combined by the RMRS merge module. The arrows in the figure indicate the data flows which we are currently using but others will be added: in later versions of the system the ERG/PET processor will be invoked specifically on subparts of the text identified by the shallower processing. The deep parser will also be able to use shallow processing results for phrases within selected areas where there are too many out-of-vocabulary items for deep parsing to give good results.

Processing which applies 'above' the sentence level, such as anaphora resolution and (non-syntactically marked) word sense disambiguation (WSD), will operate on the merged RMRS, enriching it further. However, note that the SciXML markup is also accessible to these modules, allowing for sensitivity to paragraph breaks and section boundaries, for instance. The application tasks are all defined to operate on RMRSs.

This architecture is intended to allow the benefits of deep processing to be realised where possible but with shallow processing outputs being used as nec-

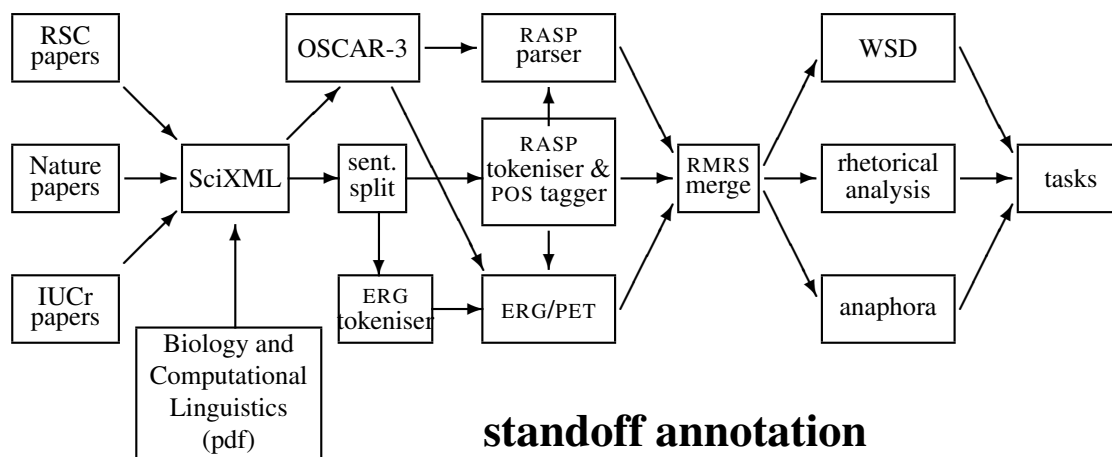


Figure 2: Overall architecture of the SciBorg system

essary for robustness. The aim is to always have an RMRS for a processed piece of text. The intent is that application tasks be defined to operate on RMRSs in a way that is independent of the component that produced the RMRS. Other modules could be added as long as they can be modified to produce RMRSs: we may, for instance, investigate the use of a chunker. The architecture allows the incorporation of modules which have been independently developed without having to adjust the interfaces between individual pairs of modules.

The parallel architecture is a development of approaches investigated in VerbMobil (Ruland et al., 1998; Rupp et al., 2000) and Deep Thought (Callmeier et al., 2004). It is intended to be more flexible than either of these approaches, although it has resemblances to the DeepThought Heart of Gold machinery and we are using some of the software developed for that. The QUETAL project (Frank et al., 2006) is also using RMRS. We have discovered that a strictly pipelined architecture is unworkable when attempting to combine modules that have been developed independently. For instance, different parsing technology makes very different assumptions about tokenisation, depending in particular on the treatment of punctuation. Furthermore, developers of resources change their assumptions over time, so attempting to make tokenisation consistent leads to a brittle system. Our non-pipelined, standoff annotation approach supports differences in tokenisation. It is generic enough to allow us to investigate a range of different strategies for combining deep and shallow processing.

4 RMRS

RMRS is an extension of the Minimal Recursion Semantics (MRS: Copestake et al. (2005)) approach which is well established in deep processing in NLP. MRS is compatible with RMRS but RMRS can

also be used with shallow processing techniques, such as part-of-speech tagging, noun phrase chunking and stochastic parsers which operate without detailed lexicons. Shallow processing has the advantage of being more robust and faster, but is less precise: RMRS output from the shallower systems is less fully specified than the output from the deeper systems, but in principle fully compatible. In circumstances where deep parsing can be successfully applied, a detailed RMRS can be produced, but when resource limitations (in processing capability or lexicon availability, for instance) preclude this, the system can back-off to RMRSs produced by shallower analysers. Different analysers can be flexibly combined: for instance shallow processing can be used as a preprocessor for deep analysis to provide structures for unknown words, to limit the search space or to identify regions of the text which are of particular interest. Conversely, RMRS structures from deep analysis can be further instantiated by anaphora resolution, word sense disambiguation and other techniques. Thus RMRS is used as the common integration language to enable flexible combinations of resources.

Example RMRS output for a POS tagger and a deep parser for the sentence *the mixture was allowed to warm* is shown below:¹

| | |
|--------------------|-------------------|
| Deep processing | POS tagger |
| h6:_the_q(x3) | h1:_the_q(x2) |
| RSTR(h6,h8) | |
| BODY(h6,h7) | |
| h9:_mixture_n(x3) | h3:_mixture_n(x4) |
| ARG1(h9,u10) | |
| h11:_allow_v_1(e2) | h5:_allow_v(e6) |
| ARG1(h11,u12) | |
| ARG2(h11,x3) | |
| ARG3(h11,h13) | |

¹For space reasons, we have shown a rendered format, rather than RMRS-XML, and have omitted much information including tense and number.

```
qeq(h13,h17)
h17:_warm_v(e18)      h7:_warm_v(e8)
  ARG1(h17,x3)
```

RMRSS consist of ‘flat’ structures where the information is factorised into minimal units. This facilitates processing and is key to the approach to underspecification. Predicates correspond to morphological stems, annotated with ‘v’, ‘n’ etc to give a coarse-grained indication of sense. These names can be constructed automatically on the basis of POS-tagged text. The POS tagged text can thus share the same lexicalised predicates as the deep parser output (*_mixture_n*, *_allow_v*, *_warm_v*, *_the_q*), although the deep parser can make more fine-grained sense distinctions (*allow_v.1*) and may insert predications that arise from particular grammatical constructions, such as compound nouns (not shown here).

The POS tagger has no relational information (indicated in the deep output by ARG1 etc). In the deep parser, predicates are specified in the lexicon and are of fixed arity. Uninstantiated relational positions in the deep output are indicated by ‘u’s, ‘e’s are eventualities, and ‘x’s other entities. The qeq condition in the deep output is a partial scope constraint which relates ‘h’ labels.

RMRS has been designed to be suitable for natural language representation and as such has to be very expressive while at the same time allowing for underspecification. Formally, RMRSS (like MRSS) are partial descriptions which correspond to a set of logical forms in a higher-order base language. RMRS itself is a restricted first order language: scope relationships are reified (via the ‘h’ labels) and natural language quantifiers, such as *every* and *most*, correspond to predicates, though these in turn correspond to generalised quantifiers in the base language. Inference in the base language will not, in general, be tractable, but some inferences can be directly expressed using RMRS without resolving to the base language. RMRSS can be linked to ontologies, so that the notion of underspecification of an RMRS reflects the hierarchical ontological relationship.

5 Domain-independent sentence processing modules

Apart from OSCAR-3 (see next section), the modules shown in Figure 2 are essentially domain-independent. Not all modules are shown explicitly. Parsing depends on the text being initially split into chunks (typically, but not necessarily, sentences). Domain-specific processing is required to identify some regions as unsuitable for passing to the parsers (e.g., data sections).

The three modules shown in Figure 2 have all been developed previously and used in a variety of applications. The RASP part of speech tagger (Briscoe and Carroll, 2002) statistically determines tags for individual tokens in a text. It processes about 10,000 words/sec (here and below the cited processing speeds are very approximate, based on a 1Ghz Pentium running Linux with 2 Gbyte of RAM). The RASP parser (Briscoe and Carroll, 2002) is a statistically-trained parser which operates without a full lexicon (speed around 100 words/sec). The English Resource Grammar (ERG) (Copestake and Flickinger, 2000) can be processed by the LKB (Copestake, 2002) or PET (Callmeier, 2002). It incorporates a lexicon with detailed linguistic information. PET is highly optimised (5–30 words/sec, depending on the corpus) while the LKB is more suited for development and can be used for generation. PET, LKB and ERG are Open Source.

The ERG can produce more detailed RMRSS than RASP, but relies on a detailed lexicon to do this. For SciBorg, lexical information comes from the hand-built ERG lexicon, plus additional hand-constructed lexical entries for very common terms in Chemistry texts, plus unknown word handling based on OSCAR-3 (see below) and POS tags. Since the ERG cannot use the same tokeniser as OSCAR-3 or the RASP tagger, unknown word processing requires a rough match of text spans.

6 Domain-specific processing modules and resources

6.1 OSCAR-3

One area in which this architecture differs from more standard approaches is the role of the software which recognises entities such as chemical compounds in the text. Consider, for instance, the following text snippet:

```
We have recently communicated that
the condensation of
5-halo-1,2,3-thiadiazole-4-carboxylate(1)
with <it>o</it>-phenylenediamine(2)
affords thiadiazepine(3)
```

Domain-specific processing is required to deal with systematic chemistry names such as that of the first compound given here. Systematic names describe the structure of the compound, they are constructed productively, and new compounds are described very frequently, so this is not a matter of simply listing all names. The amaneunsis application needs to know what compound is being referred to: this is necessary to support search on particular compounds, for instance. In contrast, most domain-independent modules need to know that this stretch of text refers to some compound, but not the identity of the specific compound, since the linguistic

properties of a sentence are insensitive to chemical composition. Schematically, the section should appear to the domain-independent parsers as:

```
We have recently communicated that
the condensation of [compound-1]
with [compound-2] affords [compound-3]
```

However the output of the language processing component must be relatable to the specific identification of the compound for searches.

It is also important that the modules know about text regions which should not be subject to general morphological processing, lexical look-up etc. In particular, in data sections of papers, standard techniques for sentence splitting result in very large chunks of text being treated as a single sentence. Given that language processing has to be bounded by resource consumption, the expense of attempting to parse such regions could prevent analysis of 'normal' text. In our current project, the domain-specific processing is handled by OSCAR-3 (Corbett and Murray-Rust, 2006).

6.2 Ontologies and other domain-specific resources

In chemistry, the need for explicit ontologies is reduced by the concept of molecular structure: structures and systematic names are (at least in principle) interconvertible, many classes of compounds (such as ketones) are defined by structural features, and structures may be related to each other using concepts such as isomerism and substructure-superstructure relationships that may be determined algorithmically. However, there are still many cases where explicit ontology is required, such as in the mapping of trivial names to structures, and in the assignment of compounds to such categories as pesticides and natural products. The three ontologies for Chemistry that we are aware of are: Chemical Entities of Biological Interest (ChEBI: www.ebi.ac.uk/chebi/index.jsp); FIX (methods and properties: obo.sourceforge.net/cgi-bin/detail.cgi?fix) and REX (processes e.g., chemical reactions obo.sourceforge.net/cgi-bin/detail.cgi?rex). FIX and REX are currently development versions, while ChEBI has been fully released. ChEBI comes in two parts: a GO/OBO DAG-based ontology ('ChEBI ontology'), and a conventional structural database. This second half is used in OSCAR-3, providing a source of chemical names and structures.

The ChEBI ontology includes chemicals, classes of chemicals and parts of chemicals: these are organised according to structure, biological role and application. Unfortunately ChEBI does not explicitly distinguish between these types of entity.

The application and biological role ontologies are currently relatively underdeveloped. However the ChEBI ontology has the potential to interact with other ontologies in the GO family.

The most useful other resource which is generally available is PubChem (pubchem.ncbi.nlm.nih.gov), which is designed to provide information on the biological activities of small molecules, but also provides information on their structure and naming. There is also the IUPAC Gold Book which is a compendium of chemical terminology with hyperlinks between entries. While not capable of directly supporting the IE needs mentioned in §2, these resources are potentially useful for lexicons and to support creation of ontology links.

7 Research markup and citation analysis

Searching in unfamiliar scientific literature is hard, even when a relevant paper has been identified as a starting point. One reason is that the status of a given paper with respect to similar papers is often not apparent from its abstract or the keywords it contains. For instance, a chemist might be more interested in papers containing direct experimental evidence rather than evidence by simulation, or might look for papers where some result is contradicted. Such subtle relationships between the core claims and evidence status of papers are currently not supported by search engines such as Google Scholar; if we were able to model them, this would add considerable value.

The best sources for this information are the papers themselves. Discourse analysis can help, via an analysis of the argumentation structure of the paper. For instance, authors typically follow the strategy of first pointing to gaps in the literature before describing the specific research goal – thereby adding important contrastive information in addition to the description of the research goal itself. An essential observation in this context is that conventional phrases are used to indicate the rhetorical status of different parts of a text. For instance, in Fig. 3 similar phrases are used to indicate the introduction of a goal, despite the fact that the papers come from different scientific domains.

Argumentative Zoning (AZ), a method introduced by Teufel (Teufel et al., 1999; Teufel and Moens, 2000), uses cues and other superficial markers to pick out important parts of scientific papers and supervised machine learning to find zones of different argumentative status in the paper. The zones are: AIM (the specific research goal of the current paper); TEXTUAL (statements about section structure); OWN (neutral descriptions of own work presented in current paper); BACKGROUND (gen-

| |
|---|
| <p>Cardiology: The goal of this study was to elucidate the effect of LV hypertrophy and LV geometry on the presence of thallium perfusion defects. Heupler et al. (1997): 'Increased Left Ventricular Cavity Size, Not Wall Thickness, Potentiates Myocardial Ischemia', <i>Am Heart J</i>, 133(6)</p> |
| <p>Crop agriculture: The aim of this study was to investigate possible relationships between type and extent of quality losses in wheat with the infestation level of <i>S. mosellana</i>. Helenius et al. (1989): 'Quality losses in wheat caused by the orange wheat blossom midge <i>Sitodiplosis mosellana</i>', <i>Annals of Applied Biology</i>, 114: 409-417</p> |
| <p>Chemistry: The primary aims of the present study are (i) the synthesis of an amino acid derivative that can be incorporated into proteins /via/ standard solid-phase synthesis methods, and (ii) a test of the ability of the derivative to function as a photoswitch in a biological environment. Lougheed et al. (2004): 'Photomodulation of ionic current through hemithioindigo-modified gramicidin channels', <i>Org. Biomol. Chem</i>, Vol. 2, No. 19, 2798-2801</p> |
| <p>Natural language processing: In contrast, TextTiling has the goal of identifying major subtopic boundaries, attempting only a linear segmentation. Hearst (1997): 'TextTiling: Segmenting Text into Multi-paragraph Subtopic passages', <i>Computational Linguistics</i>, 23(1)</p> |
| <p>Natural language processing: The goal of the work reported here is to develop a method that can automatically refine the Hidden Markov Models to produce a more accurate language model. Kim et al. (1999): HMM Specialization with Selective Lexicalization, <i>EMNLP-99</i></p> |

Figure 3: Similar phrases across the domains of chemistry and computational linguistics

erally accepted scientific background); CONTRAST (comparison with or contrast to other work); BASIS (statements of agreement with other work or continuation of other work); and OTHER (neutral descriptions of other researchers' work). AZ was originally developed for computational linguistics papers, but as a general method of analysis, AZ can and has been applied to different text types (e.g., legal texts (Grover et al., 2003) and biological texts (Mizuta and Collier, 2004)) and languages (e.g., Portuguese (Feltrim et al., 2005)); we are now adapting it to the special language of chemistry papers and to the specific search tasks in eChemistry. Progress towards construction of citation maps is reported in Teufel (2005) and Teufel et al. (2006).

The zones used in the original computational linguistics domain concentrated on the phenomena of attribution of authorship to claims (is a given sentence an original claim of the author, or a statement of a well-known fact) and of citation sentiment (does the author criticise a certain reference or use it as part of their own work). For application of AZ to chemistry, changes need to be made which mirror the different writing and argumentation styles in chemistry, in comparison to computational linguistics. Argumentation patterns are generally similar across the disciplines (they are there to convince the reader that the work undertaken is sound and grounded in evidence rather than directly carrying scientific information), but several factors such as the use of citations, passive voice, or cue phrases vary across domains.

For Chemistry, we intend to exploit the RMRS technology discussed earlier to detect cues. RMRS

encoding is advantageous because it allows more concise and flexible specification of cues than do string-based patterns and because it allows identification of more complex cues. For instance, papers quite frequently explain a goal via a contrast using a phrase such as: *our goal is not to X but to Y*:

our goal is not to verify P but to construct
a test sequence from P²

Getting the contrast and scope of negation correct in such examples requires relatively deep processing. Processing of AZ cue phrases with ERG/PET should be feasible because their vocabulary and structure is relatively consistent.

8 Conclusion

The aim of this paper has been to describe how the separate strands of work on language processing within SciBorg fit together into a coherent architecture. There are many aspects of the project that we have not discussed in this paper because we have not yet begun serious investigation. This includes word sense disambiguation and anaphora resolution. The intention is to use existing algorithms, adapted as necessary to our architecture. We have also not discussed the application of Grid computing that will be necessary as we scale up to processing thousands of papers.

²Gargantini and Heitmeyer (1999), 'Using Model Checking to Generate Tests from Requirements Specifications' In Nierstrasz and Lemoine (eds), *Software Engineering - ESEC/FSE'99*, Springer

Acknowledgements

We are grateful to the Royal Society of Chemistry, Nature Publishing Group and the International Union of Crystallography for supplying papers. This work was funded by EPSRC (EP/C010035/1) with additional support from Boeing.

References

- Briscoe, Ted, and John Carroll. 2002. Robust accurate statistical annotation of general text. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002)*.
- Callmeier, Ulrich. 2002. Pre-processing and encoding techniques in PET. In Stephan Oepen, Daniel Flickinger, Jun'ichi Tsujii, and Hans Uszkoreit, eds., *Collaborative Language Engineering: a case study in efficient grammar-based processing*. Stanford: CSLI Publications.
- Callmeier, Ulrich, Andreas Eisele, Ulrich Schäfer, and Melanie Siegel. 2004. The DeepThought Core Architecture Framework. In *Proc. of LREC-2004*.
- Copestake, Ann. 2002. *Implementing Typed Feature Structure Grammars*. CSLI Publications.
- Copestake, Ann. 2003. Report on the design of RMRS. DeepThought project deliverable.
- Copestake, Ann, and Dan Flickinger. 2000. An open-source grammar development environment and broad-coverage English grammar using HPSG. In *Proceedings of the Second conference on Language Resources and Evaluation (LREC-2000)*, 591–600.
- Copestake, Ann, Dan Flickinger, Ivan Sag, and Carl Pollard. 2005. Minimal Recursion Semantics: an introduction. *Journal of Research in Language and Computation* 3(2–3): 281–332.
- Corbett, Peter, and Peter Murray-Rust. 2006. High-throughput identification of chemistry in life science texts. In *Proceedings of the 2nd International Symposium on Computational Life Science (CompLife '06)*. Cambridge, UK.
- Feltrim, Valeria, Simone Teufel, G. Gracas Nunes, and S. Alusio. 2005. Argumentative Zoning applied to Critiquing Novices' Scientific Abstracts. In James G. Shanahan, Yan Qu, and Janyce Wiebe, eds., *Computing Attitude and Affect in Text*. Dordrecht, The Netherlands: Springer.
- Frank, Anette, Hans-Ulrich Krieger, Feiyu Xu, Hans Uszkoreit, Berthold Crysmann, Brigitte Jörg, and Ulrich Schäfer. 2006. Question Answering from Structured Knowledge Sources. *Journal of Applied Logic, Special Issue on Questions and Answers: Theoretical and Applied Perspectives* 1.
- Grover, Claire, Ben Hachey, and Chris Korycinsky. 2003. Summarising legal texts: Sentential tense and argumentative roles. In *Proceedings of the NAACL/HLT-03 Workshop on Automatic Summarization*.
- Hearst, Marti A. 1992. Direction-Based Text Interpretation as an Information Access Refinement. In Paul S. Jacobs, ed., *Text-based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval*. Hillsdale, NJ: Lawrence Erlbaum.
- Hollingsworth, Bill, Ian Lewin, and Dan Tidhar. 2005. Retrieving Hierarchical Text Structure from Typeset Scientific Articles — a Prerequisite for E-Science Text Mining. In *Proc. of the 4th UK E-Science All Hands Meeting*, 267–273. Nottingham, UK.
- Mizuta, Yoko, and Nigel Collier. 2004. An Annotation Scheme for Rhetorical Analysis of Biology Articles. In *Proceedings of LREC'2004*.
- Ruland, Tobias, C. J. Rupp, Jörg Spilker, Hans Weber, and Karsten L. Worm. 1998. Making the Most of Multiplicity: A Multi-Parser Multi-Strategy Architecture for the Robust Processing of Spoken Language. In *Proc. of the 1998 International Conference on Spoken Language Processing (ICSLP 98)*, 1163–1166. Sydney, Australia.
- Rupp, CJ, Ann Copestake, Simone Teufel, and Ben Waldron. 2006. Flexible Interfaces in the Application of Language Technology to an eScience Corpus. In *Proceedings of the 4th UK E-Science All Hands Meeting*. Nottingham, UK.
- Rupp, C.J., J. Spilker, M. Klarner, and K.L. Worm. 2000. Combining Analyses from Various Parsers. In W. (ed.) Wahlster, ed., *VerbMobil: Foundations of Speech-to-Speech Translation*, 311–320. Berlin: Springer Verlag.
- Teufel, Simone. 2005. Argumentative Zoning for improved citation indexing. In James G. Shanahan, Yan Qu, and Janyce Wiebe (Eds.), eds., *Computing Attitude and Affect in Text: Theory and Applications*, 159–170. Springer.
- Teufel, Simone, Jean Carletta, and Marc Moens. 1999. An annotation scheme for discourse-level argumentation in research articles. In *Proceedings of the Ninth Meeting of the European Chapter of the Association for Computational Linguistics (EACL-99)*, 110–117.
- Teufel, Simone, and Marc Moens. 2000. What's yours and what's mine: Determining Intellectual Attribution in Scientific Text. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.
- Teufel, Simone, Advait Siddharthan, and Dan Tidhar. 2006. An annotation scheme for citation function. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*. Sydney, Australia.
- Uszkoreit, Hans. 2002. New chances for deep linguistic processing. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*. Taipei, Taiwan.
- Vlachos, Andreas, and Caroline Gasperin. 2006. Bootstrapping and Evaluating Named Entity Recognition in the Biomedical Domain. In *Proc. Proceeding of BioNLP (Poster session) in HLT-NAACL*. New York.
- Waldron, Benjamin, and Ann Copestake. 2006. A Stand-off Annotation Interface between DELPH-IN Components. In *Proceedings of the fifth workshop on NLP and XML (NLPXML-2006)*. Trento, Italy.
- Waldron, Benjamin, Ann Copestake, Ulrich Schäfer, and Bernd Kiefer. 2006. Preprocessing and Tokenisation Standards in DELPH-IN Tools. In *Proceedings of LREC2006*. Genoa, Italy.