Integrating General-Purpose and Domain-Specific Components in the Analysis of Scientific Text.

C.J. Rupp[1], Ann Copestake[1], Peter Corbett[2], Benjamin Waldron[1]

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

Abstract

The SciBorg project employs two distinct parsing frameworks: RASP (Briscoe and Carroll, 2002) and the ERG (Copestake and Flickinger, 2000), in conjunction with the OSCAR3 (Corbett and Murray-Rust, 2006) system, in the analysis of a corpus of chemistry research papers. Here, we address a number of issues arising from the integration of these diverse components.

1 Introduction

This paper describes issues arising in the integration of two distinct sets of natural language analysis tools with a specialised annotation tool for chemistry in research text. Our aim is to provide an analysed corpus of chemistry research papers for a variety Information Extraction (IE) applications. This task implies a level of robustness to provide coverage of a corpus saturated to varying degrees with specialised technical notations and language.

A major part of our strategy, in attempting to achieve robust coverage of the corpus, is the use of parsing components based on different strategies and resources. This is similar to multi-engine analysers used, for example in Verbmobil (Ruland et al., 1998; Rupp et al., 2000), for the robust analysis of spoken language. In addition, the specialised analysis of chemical terms requires a closer integration, as this information comes in closer to the lexical level. As we require a framework that allows the interaction of analysis strands at several levels, we try to exploit this by allowing different analysers to support each other. In practice, this means a higher degree of integration at the levels of tokenisation and POS (part of speech) tagging, as well as the final parse result. We have described both the architecture and representational framework for these interactions in two recent papers (Copestake et al., 2006; Rupp et al., 2006). Here, we focus more on the practical consequences and the types of interaction that arise in the integration of specialised and general-purpose analysers.

We will first characterise the nature of text in chemistry research papers. We will, also, provide an outline of the framework we have constructed to allow the integration of results at various levels of analysis. Then, we will describe the analysis components at our disposal and the patterns of interaction we foresee between those components. We will explore the challenges we have faced in integrating analyses based on differing knowledge sources, providing illustrative examples.

2 Chemistry Research Text

Our project, SciBorg, is a collaboration between university researchers and journal publishers. This gives us access to a large corpus of research papers for chemistry. In fact, it alleviates one standard set of text mining problems, namely the document format. Our corpus exists in uniform XML markup for research text (SciXML), having been converted from the various in-house XML markup schemes used by each of the publisher. The notion of SciXML (Teufel, 1999; Rupp et al., 2006), as a mark up for the logical structure of research papers, underlies our approach to the analysis of chemistry research text, taking the characterisation of scientific research text as more primitive than the specific properties of chemistry.

Of course, the prevalence of domain-specific terms, notation and vocabulary is immediately apparent in any arbitrary sample of text from a paper in the SciBorg corpus. However, the variation from patterns of standard language is not as uniform. At one extreme, we have sections in which very little actual text is recoverable. Conversely, there are sentences which, accidentally, contain no domain-specific terms at all. The conventional structure of chemistry research papers also plays a role in relative distribution of these different phenomena. Table 1 shows three passages from single paper.

Even without specialised knowledge of the domain

1. Dialkyl 9-chloro-3*H*-pyrrolo[1,2-*a*]indole-2,3-dicarboxylates are obtained in excellent yields from the 1 : 1 : 1 addition reaction between triphenylphosphine, dialkyl acetylenedicarboxylates and 3-chloroindole-2-carbaldehyde; dimethyl 9-chloro-3*H*-pyrrolo[1,2-*a*]indole-2,3-dicarboxylate is converted to dimethyl 9-oxo-9*H*-pyrrolo[1,2-*a*]indole-2,3-dicarboxylate.

2. Bridgehead nitrogen heterocycles are of interest because they constitute an important class of natural and non-natural products, many of which exhibit useful biological activity.

3. Diethyl 9-chloro-3*H*-pyrrolo[1,2-*a*]indole-2,3-dicarboxylate (**3b**) Yellow crystals, mp 8788 C (from *n*-hexaneethyl acetate 1 1), yield 0.64 g, 96%. IR (KBr) ($_{max}$/cm^1): 1731 and 1693 (CO). MS, *m*/*z* (%): 333 (M^+, 27), 298 (10), 260 (100), 232 (60), 188 (20), 152 (12). Anal. Calcd for $C_{17}H_{16}NO_4Cl$ (333.77): C, 61.18; H, 4.83; N, 4.20. Found: C, 61.2; H, 4.8; N, 4.2%. ^{1}H NMR: 1.25 and 1.35 (6 H, 2 t, *J* = 7.1 Hz, 2 OCH_2CH_3), 4.23 (2 H, AMX_3 system, ^2J = 10.8 Hz and ^3J = 7.1 Hz, OCH_2CH_3), 4.31 (2 H, m, ABX_3 system, OCH_2CH_3), 5.51 (1 H, d, ^4J = 1.4 Hz, CH), 7.16 (1 H, t, *J* = 7.7 Hz, CH), 7.28 (1 H, t, *J* = 7.7 Hz, CH), 7.34 (1 H, d, *J* = 7.7 Hz, CH), 7.62 (1 H, t, *J* = 7.7 Hz, CH), 7.63 (1 H, d, ^4J = 1.4 Hz, CH). ^{13}C NMR: 14.08 and 14.25 (2 OCH_2CH_3), 61.20 and 62.48 (2 OCH_2CH_3), 64.01 (CH), 101.21 (C), 110.06, 119.99, 120.86, and 124.85 (4 CH), 129.76 (C), 130.53 (CH), 133.88, 135.18, and 139.46 (3 C), 162.14 and 166.60 (2 CO).

Table 1: Three fragments of chemistry research text.

it is relatively easy to determine which is the abstract, which is from the introduction and which is a data section from the body of the paper. While the remit of the SciBorg project includes the analysis of the logical structure of chemistry research we do not extend this to predicting the distribution of specialised terms and notation. We assume instead that majority of the text combines specialised and standard language in some mixture and, generally preserves syntactic patterns found in other forms of research text and standard language. We, therefore, feel justified in approaching the analysis of chemistry research text on the basis of general-purpose linguistic analysis tools, augmented to a large extent by domain-specific information.

We are fortunate to have access to a tool designed specifically for the annotation of text with information about the chemistry it contains. Indeed the intended application of OSCAR3 includes assigning chemical structures to terms it recognises. We utilise OSCAR3 primarily to locate and classify chemical terms and to mark out those sections of the text which will not be susceptible to analysis as natural language text.

3 Architecture and Common Representation Framework

The architecture and representational framework that we employ has been described in detail in two recent papers Copestake et al. (2006); Rupp et al. (2006). Here, we will focus on the features which enable the integration of diverse analysis components and support robust processing, in general. A distinctive feature of our architecture is that it allows multiple analysers to support each other. The most obvious example of this is the way that the OSCAR3 annotation of chemistry terms is combined in all analysis strands, but equally the two parsing frameworks we employ can complement each other.

We have adopted two conventions to aid the communication between modules and the storing of intermediate results in a broadly compatible format. The first of these is a uniform formalism to represent all analyses at all levels as standoff annotations in a common XML formalism, named rather prosaically, or perhaps ambitiously, Standoff Annotation Formalism (SAF). This has been reported in Rupp et al. (2006) and also in Waldron and Copestake (2006). The other main form of representation that we use is also a convergence point, but in a different, more substantive way. The ERG parsing framework returns linguistic analyses in an underspecified semantic formalism: Robust Minimal Recursion Semantics (Copestake, 2003). This is a variant of the MRS semantics more generally employed in HPSG grammars. RMRS allows varying degrees of underspecification so that partial analyses and even POS taggings can be assigned a representation. RMRS construction from RASP analysis trees has be defined and incorporated in the version of the RASP parser that we employ. This means that we have a convergence on parse outputs because they can all be expressed in a common formalism, which also allows RMRSs with partial information to be merged.

```
<!ELEMENT saf   (olac? , fsm ) >
<!ATTLIST saf   addressing CDATA #REQUIRED
                document CDATA #REQUIRED >

<!ELEMENT fsm   (state* , annot*)>
<!ATTLIST fsm   initial IDREF #IMPLIED
                final IDREF #IMPLIED>

<!ELEMENT annot (slot|fs|rmrs)* >

<!ATTLIST annot id ID #REQUIRED
                refid CDATA #IMPLIED
                from CDATA #IMPLIED
                to CDATA #IMPLIED
                source CDATA #IMPLIED
                target CDATA #IMPLIED
                value CDATA #IMPLIED
                type CDATA #REQUIRED
                deps IDREFS #IMPLIED >
```

Figure 1: Partial DTD for the SAF formalism.

3.1 Standoff Annotation Formalism

The Standoff Annotation Formalism (SAF) is a generalisation of similar frameworks, e.g. MAF (Clement and de la Clergerie, 2005) and SMAF (Waldron et al., 2006), to accommodate annotation to any span of an original text without predicting the nature of that annotation. The restrictions on what can be expressed in a SAF annotation are quite weak as some very general representations are supported within SAF annotations, but so are quite specific structures, namely the XML expression of RMRS semantic representations. What unifies the SAF annotations is their expression as annotation on a span of text, indexed by start and end positions, typically expressed as character positions, although we do support a mixture of character position and XML tree position in XPoint notation. The collection of SAF annotations form a lattice, similar to a chart or well-formed substring table in a chart parser, or to a word hypothesis graph or word lattice in Speech Recognition. Although, the SAF annotations encode different types for information from the outset which is less typical of these other representations.

SAF annotations are also comparable to the standoff annotations used within the UIMA framework (Ferrucci and Lally, 2004b,a). UIMA annotations are indexed to character positions relative to a specified document, or *subject of analysis*. This would facilitate treating the publisher markup as just another form of annotation, converting the inline XML markup to standoff annotations. The price to pay for that approach is that the annotations are indexed to an artefact, such as a file containing only the text to be analysed, that may not exist outside the analysis framework. We prefer to fix the SciXML document as our point of reference, as the existence of this format is independently motivated. In return, we are able to make use of both character positions and XML structure for indexing.

Figure 1 shows the core DTD definition for a SAF lattice, including embedded RMRS feature structure (`fs`) and slot-filler structures. Definitions for each are, of course, included in the full DTD. The form of a single annotation would resemble the following example:

```
<annot type='rasp_token' id='t1599'
from='15282' to='15306'
source='v1803' target='v1804'
value='n-butyl-substituted' deps='s63'/>
```

In practice, the main advantage of a lattice representation is to maintain different analyses, building on the well documented property of standoff annotation schemes in general: that they can represent orthogonal structures over the same data, including overlapping tree structure prohibited to inline XML annotations. We can exploit this property in combining information from different sources which may not agree on either the analysis, or even the segmentation of the text. This allows the maintenance of possibly conflicting hypotheses about the structure of the text, until there are grounds to select from the best available options. Typically such choices fall out from higher level analyses, but we must ensure that intermediate results are made available to subsequent analysers in an appropriate form.

3.2 RMRS Semantic Structures

The benefit we gain from SAF annotations follows from the application of a simple principle. There is also a simple principle underlying underspecified semantic representations, which is instantiated in one of its most extreme forms in Robust Minimal Recursion Semantics. Underspecification in semantics is implemented by encoding a representation of a logical expression rather than the expression itself. A partial representation can then represent a set of formulae, leaving underspecified which member of that set may ultimately be the intended semantics in a given context. This starts to get interesting when you need to formulate the representations, so that specific semantic functions of the original formula can be underspecified in a gradiated way. The original application was in Machine Translation where the logical complexities of quantifier scope were perceived to be irrelevant to the translation task in many instances. This led to representations which factor out quantifier dependencies that are integral to the syntax of a traditional logic. RMRS is built on such a language, MRS, but also abstracts away from logical argument structure, to the extent that analyses that cannot resolve argument attachments can also be represented. Factoring the individual properties of a logical formula into separate relations requires the introduction of

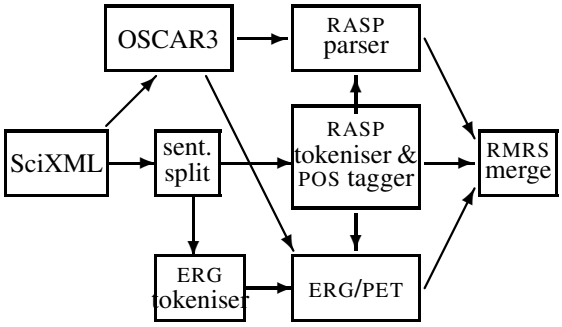

Figure 2: Parsing architecture of the SciBorg system

additional variables at the representational level. These serve a similar role to the indices in a standoff annotation in guiding how the individual facts can be reassemble to describe a formula. So, at a very abstract level, our two representational frameworks are similar in breaking things up in order to maintain the consistent parts of multiple analyses, until a more informed choice can be made between alternatives. The fallback position of not ultimately making the choice must also be held in reserve.

4 Analysis Components

We make use of two sets of analysis tools which have been developed over a number of years for natural language analysis tasks in general. We, essentially, make use of the analysis components as provided. Our main engineering tasks have concerned ensuring that the internal interfaces support the annotation framework that we use to record all analyses in a uniform way, so that they may be reused by another component or at a later time. On the one hand, this makes it easier to integrate partial analyses derived from the OSCAR3 annotations of chemical terms, on the other hand it allows us to break up the two pipelines of components provided by the original parsers, resulting in a more heterogeneous but somewhat more complex architecture.

The architecture diagram in Figure 2 shows connections between OSCAR3 and both ERG and RASP parsers and inputs to the ERG parser, from both the ERG tokeniser and the RASP tokeniser and tagger. We use a uniform representation for analysis results at all levels, so a range of interfaces would, in principle, be possible. The internal interfaces of each analysis strand are effectively opened up by treating intermediate results in a uniform manner, but the restricting factor is, largely, the interfaces that are foreseen in subsequent analsysis components. The RASP parser itself requires a unique sequence of tokens paired with tags, so the integration of analyses for chemistry terms from OSCAR3 must match that pattern, selecting tokens that reflect the structure of the chemical terms and converting their classification into a POS tag, or borrowing this information from the original tagging where this is more appropriate. The ERG is more liberal in its input conditions effectively accepting a chart of lexical tokens, but it also foresees an unknown word mechanism that can be seeded with part of speech information, so we can equally include the RASP tagging in an input chart to support this fallback mechanism. While our general approach to interface structures enhances the interoperability of the individual analysis components, the limiting factors are the interfaces that the components themselves provide.

4.1 OSCAR3 Annotations

The OSCAR3 system is designed to annotate a text containing chemical terms with the structures that those terms represent. In the context of the linguistic analysis of chemistry research text, its main contribution is to recognise and classify chemical terms. In effect, we use the OSCAR3 annotations as an extended form on Named Entity Recognition (NER) for chemistry, assuming that naming is not restricted to nominal categories. Although the largest contribution is in the area of systematic chemical names, which are productive and could not be listed in a specialised technical lexicon. OSCAR3 also provides a domain-specific classification of a range of further types of chemical terms, and specialised language use. It can, equally, locate data sections that cannot be productively subjected to linguistic analysis. This also saves a considerable amount of work as linguistic tokenisers would, for example, attempt to treat decimal points and parentheses as punctuation, inflating the number of potential tokens with no return in terms of coherent analyses.

The primary classification that OSCAR3 provides focuses on distinctions relevant to the chemistry annotation task:

CM Compound names: (e.g.) dialkylpyridines, citric acid, $C_6H_{12}O_6$.

CJ Chemical adjective: citric, pyrazolic, aqueous.

RN Reaction name: dihydroxylate, iodise, chlorinated.

ASE Enzyme: methylase, nitrogenase, ethyltransferase.

CPR Chemical prefix: 1,3-dipolar, cis-isomers, α-position.

Compound names are by far the most frequent of these terms and, since they clearly correspond to names, map easily onto nominal categories, either as POS tags or lexical types. Chemical adjectives can also be interpreted as unknown adjectives in a similar way. However, the terms associated with reactions may belong to a variety of lexical categories, corresponding to verbs and any category that can be derived by deverbal morphology. Our current versions of OSCAR3 attempt to provide a POS suggestion based on comparison with a tagger trained specifically for the biomedical domain, but this conversion is, in principle, less reliable, because it represents a generalisation over a much more complex relation. The prefix category is problematic in a different way. Of course, our linguistic analyses will not normally provide us with a classification for an isolated prefix. Moreover, positional prefixes are often attached to word forms that may either be in use in standard language or correspond to specialised usages of common words. However, our linguistic analysers may overlook these words if they are not tokenised separately from the prefix.

4.2 RASP

The backbone of our architecture is the RASP system, as this provides an instantiation of each of the components required for a full analysis of text: a parser, part of speech tagger, tokeniser and sentence splitter. The parser is a statistical CFG parser trained on a balanced, multi-genre corpus. The tagger is also a general-purpose component, using the CLAWS7 tagset. The tokeniser is a Flex program, as is the sentence splitter, which actually makes some concession to scientific text, in adding some common abbreviations.

For most users the RASP system can be run by a script which links up the components in a pipeline. We require a little bit more of the internal interfaces of this system. To provide SAF annotations representing all analyses, including taggings, tokens and sentence boundaries, we need to derive index positions in the original SciXML file determining the exact span of each analysis.

The RASP components support character positions and the standard RASP release has some facilities for processing text with XML markup, but these facilities do not add up to the preservation of consistent indexing relative to the source file. We therefore have to adjust the character positions passed through the RASP framework. In fact, we make use of the offsets provided by a string input to the tokeniser and increment them with lengths of the intervening XML elements. The `rasp_token` example shown in Section 3.1 is derived from the token, indexed on tokenised character positions from the start of the current sentence:

```
<w s='6' e='24'>n-butyl-substituted</w>
```

but we generate:

```
 <w s='15282' e='15306' id='t1599'>
n-butyl-substituted</w>
```

from a SAF representation to pass on to the RASP tagger interface[1]. Once we are able to locate the individual RASP analyses, at all levels, relative to the original source file, we can submit these analyses to any other component in our architecture.

The input of information from other sources into the RASP analysis stream is also subject to some restrictions. There is scope for submitting alternative tokenisations or a combined sequence of tokens with associated taggings, but these must conform to a single sequence, because of the pipeline nature of the original RASP architecture. An ambiguous tagging is permitted, as the RASP tagger has a statistically weighted output mode, but the tokenisation must be decided. This implies in general a path selection from the SAF lattice to generate RASP interface structures. By weighting the token and tag annotations provided by OSCAR3 we can ensure a maximum uptake of specialised analyses in these interface structures, but we have no fallback position, if these do not lead to a coherent analysis. The addition of an OSCAR3 edge substitutes the original RASP analysis.

The RMRS analyses provided by the RASP parser are constructed by the application of a specific rule system to an analysis tree. They typically produce a spanning analysis but this may include fragmentary structures. The successful uptake of OSCAR3 analyses must, therefore, be judged according to the quality of the RMRS analyses provided by the RASP analysis stream. This is not a simple task given, the formal complexity of RMRS structures.

4.3 ERG Analysers

The most sophisticated parsers we have access to are based on the ERG grammar, defined as an HPSG grammar of English. Of these, the PET parser (Callmeier, 2002) is the most efficient. This parser makes use of a model for parse selection, based on a tree bank. The set of the best ranked RMRS analyses is returned if an analysis is available.

As with the RASP parser we found it helpful to separate out an early processing module into a separate component to give freer access to an internal interface. The ERG tokeniser, however, is designed with the possibility of multiple ambiguous tokenisations

[1] Note that the span is greater in the SciXML file because the term is partially italicised.

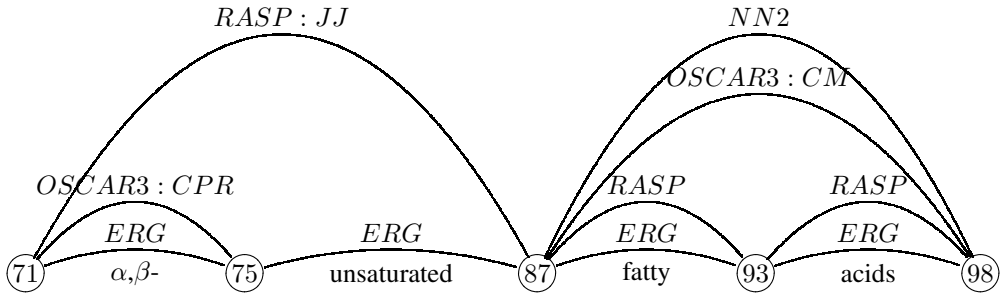

Figure 3: A chart for the tokenisation and tagging of: "α,β-unsaturated fatty acids"

in mind. This allows us to integrate the OSCAR3 annotations in a more robust manner, by taking the full sublattice of tokens with the original ERG tokens available alongside the OSCAR3 ones. The most successful analysis will then prevail irrespective of the proportion of domain-specific tokens it contains. Thus, there is less pressure on both OSCAR3 to avoid false positive NER results and on the ERG lexicon to include specialised vocabulary.

However, the fact that an ERG analyser relies on a detailed lexicon can be a restriction when applied to free text. PET includes an unknown word mechanism, assigning generalised lexical types to unknown forms, but this requires some indication of the intended category. The limitations on lexical coverage are not restricted to specialised language covered by OSCAR3, so the most reliable source of seed information, required for the handling of unknown words, is the POS tagging provided by RASP. This process, of course, requires that the mapping from tags to lexical types is defined for a given tagset. The mapping for RASP tags includes:

```
pos.[tag='NN1'] -> gMap.type='n_-_c-sg-unk_le'
pos.[tag='JJ'] -> gMap.type='aj_-_i-unk_le'
```

where the types are also defined:

```
n_-_c-sg-unk_le := generic_n_intr_lex_entry &
  [ SYNSEM noun_nocomp_synsem &
    [ LOCAL sing_noun ] ].
aj_-_i-unk_le := unknown_word &
  [ SYNSEM intrans_adj_synsem &
           [ LOCAL [ CAT [ HEAD [ MOD
                                    < [ --SIND
                                      #ind ] >,
                                  PRD #bool ],
                           POSTHD #bool ],
                     CONT [ HOOK.XARG #ind,
                            RELS <! #key !> ] ],
             LKEYS.KEYREL #key &
                          adj_wcarg_relation &
                          [ PRED unknown_adj_rel,
                            ARG1 #ind ] ] ].
```

(albeit somewhat esoterically). The lattice submitted to the PET parsing component, therefore contains both OSCAR3 token edges and RASP tag edges, to achieve the maximum coverage at the RMRS level.

There is another mismatch between the RASP and ERG analysis streams at a more fundamental level, in that they do not share the same tokenisation conventions. While RASP tends to emphasise conventional word boundary markers, such as spaces and punctuation. The ERG typically includes trailing punctuation in the word form, but will treat a hyphen as a potential word boundary. Both of these tokenisations are purely formal, as tokenisation precedes any linguistic analysis. This is typical for English, where morphological analysis not a prerequisite for tokenisation. By treating OSCAR3 terms which have been subject to an internal analysis as additional tokens, we include a further type of tokenisation which may also be a source of conflicts.

5 Integration Issues

We have so far emphasised the positive aspects of our approach in integrating general-purpose linguistic analysers with a domain-specific annotator, providing specialised NER terms, to achieve broad coverage of chemistry research texts. We have indicated how we attempt to optimise the level of integration by making use of the various interface options that the individual components make available. That is in addition to the overall integration of the components in a framework that maintains competing hypotheses in a lattice structure of partial analyses and allows the convergence of final analyses by adopting a uniform semantic formalism.

The practical application of this approach to a large corpus is ongoing work. Some aspects of this cannot yet be adequately evaluated, for lack of either sufficient data or formal evaluation tools. However, we can provide examples, particularly from the early stages of processing which suggest that there are additional integration issues to address.

5.1 Tokenisation

As we have indicated above we have, in effect, three different tokenisations, based on the RASP and ERG tokenisers and on taking OSCAR3 as an NER com-

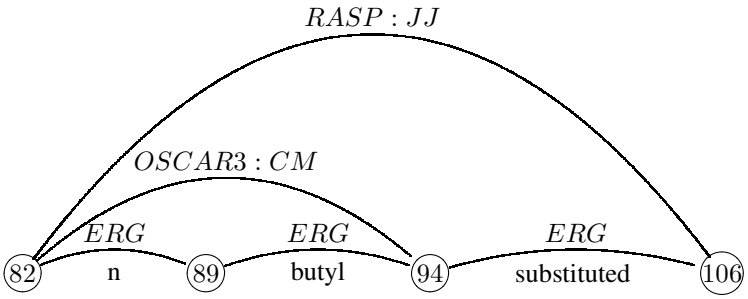

Figure 4: A chart representing the tokenisation of: "*n*-butyl substituted"

ponent. Where these coincide, it is relatively easy to pass information between the different analysis strands. However, various types of mismatch occur, some are predictable and some are recoverable.

Figures 3 and 4 show two examples highlighting several of the potential mismatches between the three tokenisation schemes. Figure 3 shows an instance of the chemical prefix (CPR) classification provided by OSCAR3, but the ERG parser will have no obvious way of combining this information with the remaining word: "unsaturated" and RASP only recognises a single token, correctly tagging it as an adjective (JJ). To continue the ERG analysis, it would be best to treat the prefix as syntactically null, taking information from the lexical entry of the stem, "unsaturated", with the possible combination of the overarching RASP tagging. In this case, the OSCAR3 classification effectively marks what not to analyse, at the morpheme level. This example also shows a chemical name including whitespace, "fatty acids", where both tokenisers only recognise two separate tokens. The chart includes a spanning edge with a plural noun tag (NN2), because we select the path including the OSCAR3 term to submit to the RASP tagger, as more reliable than the two tokens provided by the general-purpose RASP tokeniser. In general, the path with the most OSCAR3 information and the least edges is preferred by this selection. The ERG parser, in contrast, can accept multiple tokenisations, although there is currently no mechanism built in to prefer OSCAR3 edges.

We attempt to merge conflicting tokenisations in a manner that furthers the OSCAR3 terms, as more reliable for chemistry text. We prefer to manage the merging and path selection processes, rather than modifying the general-purpose linguistic analysers, but this is not always adequate. In Figure 4, OSCAR3 classifies "*n*-butyl" as a chemical name, but neither the ERG or RASP tokenisations can easily assimilate this information. The RASP tagging as an adjective (JJ) is probably the most useful information attached to the phrase. One strand of analysis survives, albeit with a more approximated analysis, but some of the OSCAR3 information gets lost.

5.2 POS Tagging, Classification and Lexical Type

Each of the main analysers provide a primary classification at the lexical level. For RASP this is the POS tag, in OSCAR3 it is the class of the annotation and for ERG it is the lexical type, where there are minimal lexical types for unknown words. We have defined mappings between these to aid the integration of results from one analysis in another, but these mappings are not all complete. We can, for example, map the CLAWS7 POS tags provided by RASP onto the primary lexical types expected by the unknown word mechanism in the PET parser, because the basic classification of possible unknown word types is a simple subset of the more detailed tagging. For OSCAR3 annotations on the other hand the primary classification has a slightly a different function, so that while you can expect a chemical name to be a noun, allowing a mapping to a minimal lexical type with an appropriate contribution to its semantic relation:

```
oscar.[type='CM'] -> gMap.type='n_-_pn-unk_le'
                     tokenStr='OSCARCOMPOUND'
oscar.[type='CM'] -> gMap.pred='chem_compound_rel'
                     gMap.carg=content.SMILES
```

a word denoting a reaction can be of virtually any part of speech for which deverbal derivational morphology exists. We have modified the OSCAR3 annotations to provide a prediction of the most likely tag, by comparison with results of the GENIA tagger (Tsuruoka et al., 2005), which has been trained on biomedical data. However, at the point where this information is integrated into an analysis it is taken as a hint and maybe superceded by the prediction of the RASP tagger in that context.

6 Conclusion

We have described the integration of two general-purpose parsing systems with a specialised NER component and highlighted some of the issues arising out of this work. This is ongoing work and,

while the most significant issues have been addressed, some problems remain, before we can demonstrate robustness in applying this analysis framework to our substantial corpus of chemistry research text.

7 Acknowledgements

We are very 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 Usz koreit, eds., *Collaborative Language Engineering: a case study in efficient gra mmar-based processing*. Stanford: CSLI Publications.

Clement, L., and E.V. de la Clergerie. 2005. MAF: a morphosyntactic annotation framework. In *Proceedings of the 2nd Language and Technology Conference*. Poznan, Poland.

Copestake, Ann. 2003. Report on the design of RMRS. DeepThought project deliverable.

Copestake, Ann, Peter Corbett, Peter Murray-Rust, C. J. Rupp, Advaith Siddharthan, Simone Teufel, and Ben Waldron. 2006. An Architecture for Language Technology for Processing Scientific Texts. In *Proceedings of the 4th UK E-Science All Hands Meeting*. Nottingham, UK.

Copestake, Ann, and Dan Flickinger. 2000. An open-source grammar development environment and broad-cover age English grammar using HPSG. In *Proceedings of the Second conference on Language Resources and Evaluation (LREC-2000)*, 591–600.

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.

Ferrucci, David, and Adam Lally. 2004a. Building an example application with the Unstructured Information Management Architecture. *IBM Systems Journal* 43(3): 455–475.

Ferrucci, David, and Adam Lally. 2004b. UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Nat. Lang. Eng.* 10(3–4): 327–348.

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, C. J., Jörg Spilker, Martin Klarner, and Karsten Worm. 2000. Combining Analyses from Various Parsers. In Wolfgang Wahlster, ed., *Verbmobil: Foundations of Speech-to-Speech Translation*, 311–320. Berlin: Springer-Verlag.

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.

Teufel, Simone. 1999. Argumentative Zoning: Information Extraction from Scientific Text. Ph.D. thesis, School of Cognitive Science, University of Edinburgh, Edinburgh, UK.

Tsuruoka, Yoshimasa, Yuka Tateishi, Jin-Dong Kim, Tomoko Ohta, John McNaught, Sophia Ananiadou, and Jun'ichi Tsujii. 2005. Developing a Robust Part-of-Speech Tagger for Biomedical Text. In *Advances in Informatics - 10th Panhellenic Conference on Informatics, LNCS 3746*, 382–392.

Waldron, Benjamin, and Ann Copestake. 2006. A Standoff Annotation Interface between DELPH-IN Components. In *The fifth workshop on NLP and XML: Multi-dimensional Markup in Natural Language Processing (NLPXML-2006)*.

Waldron, Benjamin, Ann Copestake, Ulrich Schäfer, and Bernd Kiefer. 2006. Preprocessing and Tokenisation Standards in DELPH-IN Tools. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC-2006)*. Genoa, Italy.