# Semantic Annotation of Papers: Interface & Enrichment Tool (SAPIENT)

**Maria Liakata**†**, Claire Q**††**, Larisa N. Soldatova**†††
Department of Computer Science
University of Wales, Aberystwyth
SY23 3DB UK
†mal@aber.ac.uk, ††ceq08@aber.ac.uk, †††lss@aber.ac.uk

## Abstract

In this paper we introduce a web application (SAPIENT) for sentence based annotation of full papers with semantic information. SAPIENT enables experts to annotate scientific papers sentence by sentence and also to link related sentences together, thus forming spans of interesting regions, which can facilitate text mining applications. As part of the system, we developed an XML-aware sentence splitter (SSSplit) which preserves XML markup and identifies sentences through the addition of in-line markup. SAPIENT has been used in a systematic study for the annotation of scientific papers with concepts representing the Core Information about Scientific Papers (CISP) to create a corpus of 225 annotated papers.

## 1 Introduction

Given the rapid growth in the quantity of scientific literature, particularly in the Biosciences, there is an increasing need to work with full papers rather than abstracts, both to identify their key contributions and to provide some automated assistance to researchers (Karamanis et al., 2008; Medlock and Briscoe, 2007). Initiatives like OTMI[1], which aim to make full papers available to researchers for text mining purposes is further evidence that relying solely on abstracts presents important limitations for such tasks. A recent study on whether information retrieval from full text is more effective than searching abstracts alone (Lin Jimmy, 2009) showed that the former is indeed the case. Their experimental results suggested that span-level analysis is a promising strategy for taking advantage of the full papers, where spans are defined as paragraphs of text assessed by humans and deemed to be relevant to one of 36 pre-defined topics. Therefore, when working with full papers, it is important to be able to identify and annotate spans of text. In previous research, sentence based annotation has been used to identify text regions with scientific content of interest to the user (Wilbur et al., 2006; Shatkay et al., 2008) or zones of different rhetorical status (AZ) (Teufel and Moens, 2002). Sentences are the structural units of paragraphs and can be more flexible than paragraphs for text mining purposes other than information retrieval.

Current general purpose systems for linguistic annotation such as Callisto[2] allow the creation of a simple annotation schema that is a tag set augmented with simple (e.g. string) attributes for each tag. Knowtator (Ogren, 2006) is a plug-in of the knowledge representation tool Protégé[3], which works as a general purpose text annotation tool and has the advantage that it can work with complex ontology-derived schemas. However, these systems are not particularly suited to sentence by sentence annotation of full papers, as one would need to highlight entire sentences manually. Also these systems work mainly with plain text, so they do not necessarily interpret the structural information already available in the paper, which can be crucial to annotation decisions for the type of high level annotation mentioned

[1] http://opentextmining.org/wiki/Main_Page

[2] http://callisto.mitre.org/manual/use.html

[3] http://protege.stanford.edu/

*Proceedings of the Workshop on BioNLP*, pages 193–200,
Boulder, Colorado, June 2009. ©2009 Association for Computational Linguistics

above. The OSCAR3 (Corbett et al., 2007) tool for the recognition and annotation of chemical named entities fully displays underlying paper information in XML but is not suited to sentence by sentence annotation.

To address the above issues, we present a system (SAPIENT) for sentence by sentence annotation of scientific papers which supports ontology-motivated concepts representing the core information about scientific papers (CISP) (Soldatova and Liakata, 2007). An important aspect of the system is that although annotation is sentence based, the system caters for identifiers, which link together sentences pertaining to the same concept. This way spans of interest or key regions are formed. SAPIENT also incorporates OSCAR3 capability for the automatic recognition of chemical named entities and runs within a browser, which makes it platform independent. SAPIENT takes as input full scientific papers in XML, splits them into individual sentences, displays them and allows the user to annotate each sentence with one of 11 CISP concepts as well as link the sentence to other sentences referring to the same instance of the concept selected. The system is especially suitable for so called multi-dimensional annotation (Shatkay et al., 2008) or ontology-motivated annotation, where a label originates from a class with properties. SAPIENT is currently being employed by 16 Chemistry experts to develop a corpus of scientific papers (ART Corpus) annotated with Core Information about Scientific Papers (CISP) covering topics in Physical Chemistry and Biochemistry.

## 2 SAPIENT System Description

We chose to implement SAPIENT as a web application, so as to make it platform independent and easier to incorporate as part of an online workflow. We have used state of the art web technologies to develop SAPIENT, namely Java, Javascript (with Asynchronous JavaScript and XML (AJAX) functionality), XSLT, CSS and XML. The system has a client-server architecture (see Figure 1), with papers being uploaded and stored on the server but functionality for annotation contained in Javascript, which runs client-side in the browser. This is inspired by but in contrast with OSCAR3 (Corbett et al., 2007), which also allows manual annotation alongside the automated annotation of chemical named entities, but where each minor edit is saved to the server, writing to a file. We chose to make more of the functionality client-side in order to reduce the number of server requests, which could become problematic if the system became widely distributed.

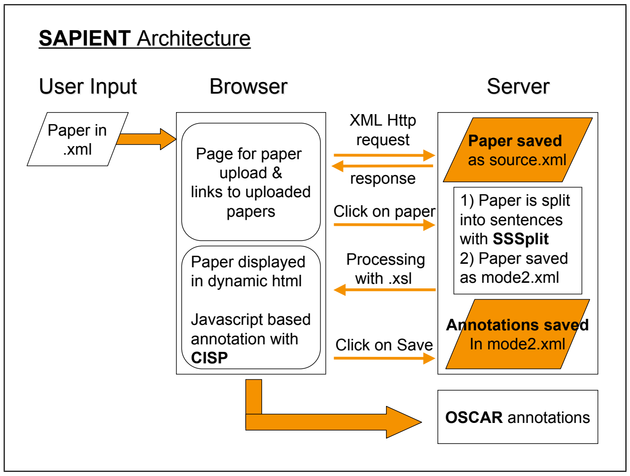
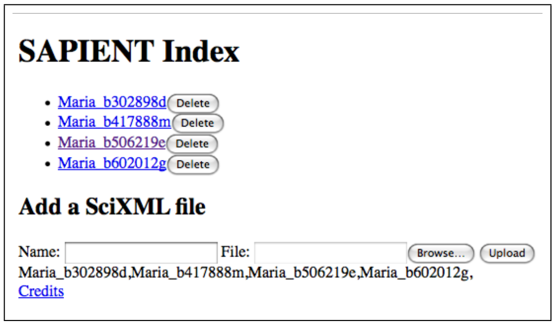

SAPIENT Architecture
User Input
Browser
Server
Paper in .xml
Page for paper upload & links to uploaded papers
Paper displayed in dynamic html
Javascript based annotation with CISP
XML Http request
response
Click on paper
Processing with .xsl
Click on Save
Paper saved as source.xml
1) Paper is split into sentences with SSSplit
2) Paper saved as mode2.xml
Annotations saved In mode2.xml
OSCAR annotations

Figure 1: Architecture of the SAPIENT System

SAPIENT has been designed to take as input full papers in XML, conforming to the SciXML schema (Rupp et al., 2006)(see Section 3).

To view or annotate a paper, a user must first upload it. The index page of SAPIENT shows a list of papers already uploaded (available as links) and an interface for uploading more papers (See Figure 2). Once the user selects a link to a paper, the paper is split into sentences using the XML-aware sentence splitter SSSplit which we have developed (See section 4) and is included in the server-side Java. The resultant XML file is stored alongside the original upload. Sentence splitting involves detecting the boundaries of sentences and, in this context, marking the latter by inline <s></s> tags added to the original XML. The <s></s> tags contain an id attribute enumerating the sentence.

After sentence splitting, the new XML file containing sentence boundaries marked by <s id=#NUM>< /s> tags is parsed by XSLT into HTML, so that it displays in the browser. In the HTML interface dynamically generated in this way, Javascript annotation drop-downs are available for



SAPIENT Index

- Maria_b302898d
- Maria_b417888m
- Maria_b506219e
- Maria_b602012g

Add a SciXML file

Name: File:

Maria_b302898d,Maria_b417888m,Maria_b506219e,Maria_b602012g,

Credits

Figure 2: Index page of the SAPIENT System

each sentence. The user can perform annotations by selecting items from the drop-downs and all the corresponding annotation information is stored in Javascript until a request to save is made by the user.

The Javascript drop-downs allow annotation at two levels (Figure 3), enabling a sentence to have a semantic label (type) with properties (subtypes) and an identifier (conceptID).

In the current implementation of SAPIENT, The **type** drop-down value corresponds to the selection of one out of 11 general scientific concepts (Liakata and Soldatova, 2008), namely (‘Background’, ‘Conclusion’, ‘Experiment’, ‘Goal of the Investigation’, ‘Hypothesis’,‘Method’, ‘Model’, ‘Motivation’, ‘Object of the Investigation’, ‘Observation’, ‘Result’). These labels originate from a set of meta-data (The Core Information about Scientific Concepts (CISP) (Soldatova and Liakata, 2007) which were constructed using an ontology methodology, based on an ontology of experiments EXPO (Soldatova and King, 2006). Because these labels map to ontology classes, they can also have properties. For example, ‘Method’ has the property ‘New’/‘Old’,‘Advantage’/‘Disadvantage’. These properties are dependent on the type selected and are expressed in terms of the **subtype** drop-down. The third drop-down, **concept ID** allows a user to provide a **concept identifier**. The latter is an entity formed by the name of a concept and a number (e.g. “Res2”). Concept identifiers uniquely identify an instance of a concept (e.g. the second Result), but not a sentence. That is, concept identifiers designate and link together instances of the same semantic concept, spread across different sentences, which can be in different parts of the paper. For example, the second result (“Res2”) can be referred to by 1 sentence in the abstract, 5 sentences in the Discussion and 2 sentences in the Conclusion sections.

The distinction between sentence identifiers and concept identifiers is an important characteristic of the system. It means that the system does not necessarily assume a ‘1-1’ correspondence between a sentence and a concept, but rather that concepts can be represented by spans of often disjoint text. Therefore, SAPIENT indirectly allows the annotation of discourse segments beyond the sentence level and also keeps track of co-referring sentences.

### 2.1 SAPIENT Usability

Even though SAPIENT has been primarily designed to work with CISP concepts, it can be used to annotate papers according to any sentence based annotation scheme. Changes required can be easily performed by modifying the XSL sheet which dynamically generates HTML from XML and organises the structure of drop-down menus. Automated noun-phrase based annotation from existing ontologies is available to SAPIENT users through OSCAR3 (Corbett et al., 2007), since SAPIENT incorporates OSCAR3 functionality for chemical named entity recognition. The latter is implemented as a link which when selected calls the OSCAR3 workflow (integrated in the system) to automatically recognise chemical named entities (NEs) (See Figure 5).

When all annotations (both sentence based and chemical NEs) are saved to the server, a new version of the XML file is produced, which contains in-line annotation for sentences as well as extra in-line annotation for the semantic concepts and NEs embedded within <s></s> tags. These annotation tags are compliant with the SciXML schema (Rupp et al., 2006) and in the case of sentence-based annotations are of the form:

```
<annotationART atype=``GSC''
                 type=#TYPE
                 conceptID=#CONCEPTID
                 novelty=``Yes/No''
                 advantage=``Yes/No''
</annotationART>
```

(See Figure 4). The attribute **type**, stands for the CISP concept selected for the sentence in question. The **conceptID** attribute is an enumerator of the particular concept, which the sentence refers to. For example, two different sentences will have different sentence ids but if they refer to the same concept (e.g. the same “Conclusion”) , they will be assigned the same concept ID (e.g. “Con3”). The attributes **novelty** and **advantage**, are properties of the concepts assigned to a sentence and depend on the concept selection. They take boolean values or the dummy value “None” if the properties are not defined for a particular concept. For example, these attributes are relevant when the concept selected is a ‘Method’, in which case the method can be “New/Old” and/or have an “Advantage/Disadvantage”. The novelty and advantage attributes co-exist in the annotation (as can be seen in Figure 4) but they are not set by the system at the same time. For instance, if a sentence refers to a new method, it will be given the type ‘Method’ and the subtype “New”; this sets the novelty attribute in the underlying XML to “Yes” and leaves the advantage attribute set to the default “None”. The sentence will also be given a conceptID, e.g. “Met1”. If another sentence refers to an advantage of this method, then the new sentence will be assigned the type ‘Method’, the subtype “Advantage” (which sets the underlying advantage attribute to “Yes”) and the same conceptID “Met1”. The novelty attribute value is then inherited from the novelty attribute value of the first coreferring sentence, which in this case is “New”.

## 3 Input: Paper in XML

SAPIENT currently accepts as input papers in XML, especially ones compliant with the SciXML schema (Rupp et al., 2006). SciXML is ideally suited for this purpose as it was developed for representing the logical structure of scientific research papers. Tags used in the schema serve the purpose of paper identification (e.g. <TITLE>,<AUTHOR>), defining sections of the paper (e.g. <DIV>,<HEADER>), text sections with specific function and formatting (e.g. <ABSTRACT>, <EQUATION>), paragraph tags <P>, references, tables, figures and footnotes, lists, bibliography. SAPIENT operates only on the <TITLE>, <ABSTRACT> ,<BODY> and <P> tags, leaving out any list elements following the body, such as acknowledgements, figures or references at the end of the paper. This is because we make the assumption that only the abstract and the body contain sentences with semantic content of any importance to the research carried out in the paper. This would have been different if SAPIENT annotated figures as well, but such provision is not currently made. Tags such as <REF>, citations in the text, are included within the sentence boundaries.

Even though SAPIENT was developed with the SciXML schema in mind, it will work with any well formed XML document that has <PAPER> as the root node and which also contains an <ABSTRACT> and <BODY> node. Therefore, it is relatively easy to adapt SAPIENT to other XML schemas.

## 4 SSSplit: Sapient Sentence Splitting

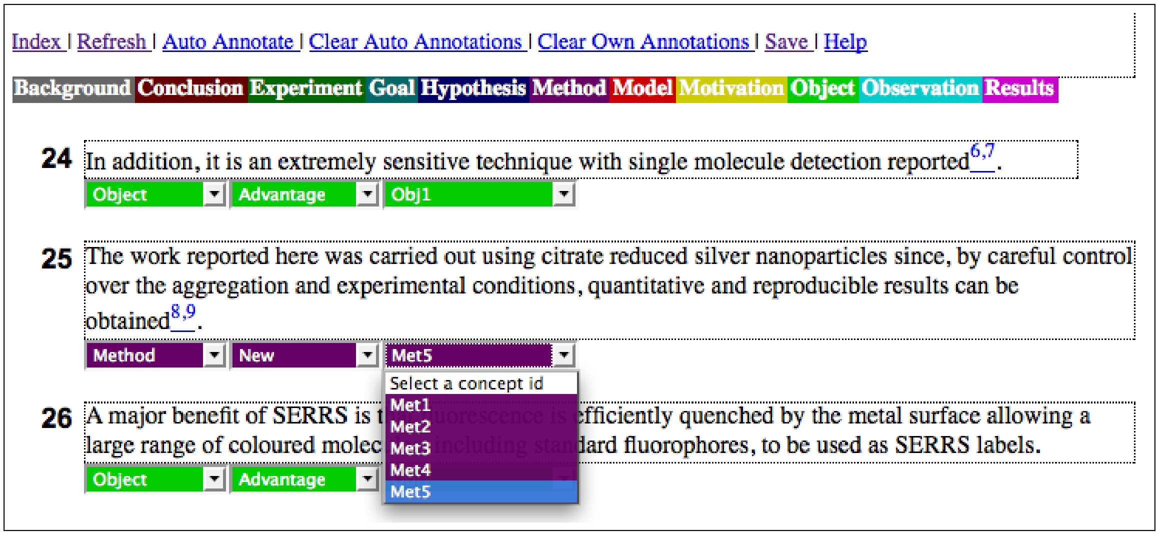
### 4.1 Sentence Matching

The reason for developing our own sentence splitter was that sentence splitters widely available could not handle XML properly. The XML markup contains useful information about the document structure and formatting in the form of inline tags, which is important for determining the logical structure of the paper. The latter is worth preserving for our purposes, since it can influence the annotation of individual sentences. XML markup (e.g. <ABSTRACT>,<REF>,<EQUATION>) needs to be combined carefully with tags designating sentence boundaries (<s></s>), so that the resulting document is in well formed XML. Current sentence splitters ignore XML markup, which means that any document formatting/information would have to be removed in order to use them. RASP (Briscoe et al., 2006), the sentence splitter used in the Sciborg project[4] at the University of Cambridge, can deal with XML but has to be compiled for different operating systems, which would result in compromising the platform independence of SAPIENT. A recent MPhil thesis (Owusu, 2008) has also developed an XML-aware sentence splitter but the code is in Microsoft C#.Net and therefore not platform independent.

We have written the XML-aware sentence splitter SSSplit in the platform-independent Java language (version 1.6), based on and extending open source Perl code[5] for handling plain text. In or-

[4] http://www.cl.cam.ac.uk/research/nl/sciborg/www/

[5] http://search.cpan.org/ tgrose/HTML-Summary-0.017/


Index | Refresh | Auto Annotate | Clear Auto Annotations | Clear Own Annotations | Save | Help
Background Conclusion Experiment Goal Hypothesis Method Model Motivation Object Observation Results
Select a concept id
Met1
Met2
Met3
Met4
Met5

24 In addition, it is an extremely sensitive technique with single molecule detection reported[6,7].

Object | Advantage | Obj1

25 The work reported here was carried out using citrate reduced silver nanoparticles since, by careful control over the aggregation and experimental conditions, quantitative and reproducible results can be obtained[8,9].

Method | New | Met5

26 A major benefit of SERRS is t uorescence is efficiently quenched by the metal surface allowing a large range of coloured molec including standard fluorophores, to be used as SERRS labels.

Object | Advantage

Figure 3: Example of SAPIENT annotation through selection from drop-down menu.

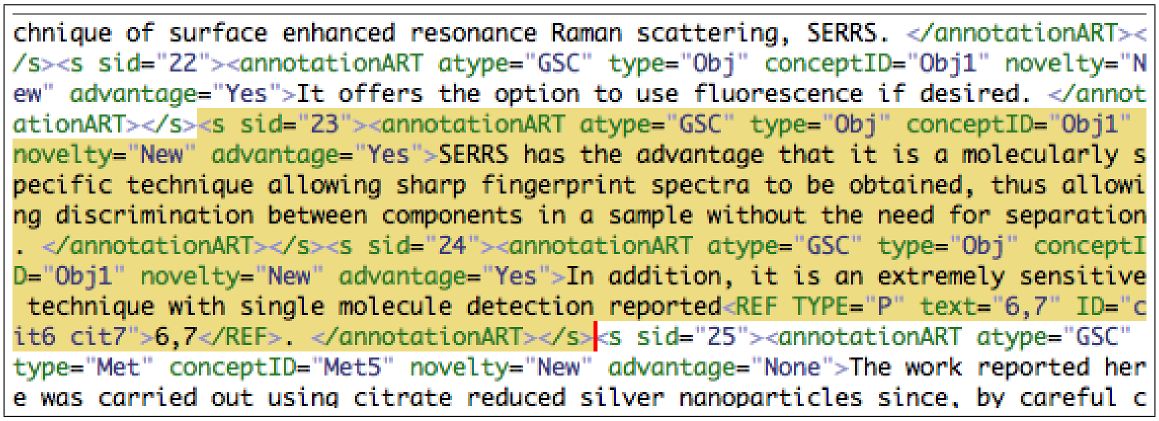
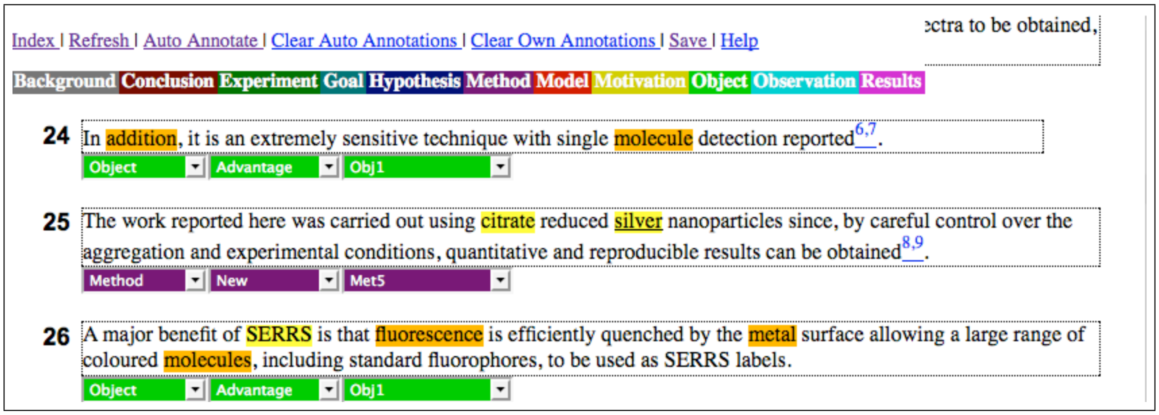

```
chnique of surface enhanced resonance Raman scattering, SERRS. </annotationART><
/s><s sid="22"><annotationART atype="GSC" type="Obj" conceptID="Obj1" novelty="N
ew" advantage="Yes">It offers the option to use fluorescence if desired. </annot
ationART></s><s sid="23"><annotationART atype="GSC" type="Obj" conceptID="Obj1"
novelty="New" advantage="Yes">SERRS has the advantage that it is a molecularly s
pecific technique allowing sharp fingerprint spectra to be obtained, thus allowi
ng discrimination between components in a sample without the need for separation
. </annotationART></s><s sid="24"><annotationART atype="GSC" type="Obj" conceptI
D="Obj1" novelty="New" advantage="Yes">In addition, it is an extremely sensitive
 technique with single molecule detection reported<REF TYPE="P" text="6,7" ID="c
it6 cit7">6,7</REF>. </annotationART></s><s sid="25"><annotationART atype="GSC"
type="Met" conceptID="Met5" novelty="New" advantage="None">The work reported her
e was carried out using citrate reduced silver nanoparticles since, by careful c
```

Figure 4: Behind the scenes: Example XML fragment of a paper annotated using SAPIENT.


ectra to be obtained,
Index | Refresh | Auto Annotate | Clear Auto Annotations | Clear Own Annotations | Save | Help
Background Conclusion Experiment Goal Hypothesis Method Model Motivation Object Observation Results

24 In addition, it is an extremely sensitive technique with single molecule detection reported[6,7].

Object | Advantage | Obj1

25 The work reported here was carried out using citrate reduced silver nanoparticles since, by careful control over the aggregation and experimental conditions, quantitative and reproducible results can be obtained[8,9].

Method | New | Met5

26 A major benefit of SERRS is that fluorescence is efficiently quenched by the metal surface allowing a large range of coloured molecules, including standard fluorophores, to be used as SERRS labels.

Object | Advantage | Obj1

Figure 5: Incorporation of OSCAR3 annotations in SAPIENT, after selecting the link “Auto Annotate”

der to make our sentence splitter XML aware, we translated the Perl regular expression rules into Java and modifed them to make them compatible with the SciXML(Rupp et al., 2006) schema. We then further improved the rules, by training on a set of 14 papers in SciXML. This involved displaying the papers, checking whether the XML was well formed and making corrections accordingly. We would observe cases of oversplit and undersplit sentences and amend the rules while keeping them as general as possible. The rules in SSSplit were evaluated by comparing the system output against a gold standard of 41 papers, where sentence boundaries had been provided by human experts (See section 4.2). The sentence splitter is integrated within the SAPIENT system but is also available as a separate package ("SSSplit"). This should enable any future work to easily incorporate or extend it. It is currently trained for splitting papers in SciXML, but can be easily ported to any other kind of XML, as discussed in section 3.

### 4.2 SSSplit Evaluation

SAPIENT and SSSplit have been have been employed by more than 20 different users to successfully display 270 full papers. For a more accurate evaluation of the quality of the sentences produced by SSSplit, we used a Perl script which compared the sentence boundaries (start and end) generated by SSSplit, to sentence tags in a set of 41 papers (SciXML files) annotated manually by human experts. If both the start and end of a sentence matched up in the generated and manual versions, we considered this a true positive result. In the case where a sentence did not match in the two versions, we first searched for a matching end in our generated set of sentences and then in the hand annotated version. If the 'true' end of the sentence (as defined by the manual annotation) was found in later sentences in the SSSplit version, this meant that the system had split a sentence too early, or "oversplit". This we considered to be a false positive, since we had detected a sentence boundary where in reality there was none. This would result in the following sentence being matched at the end only, which also counts as a false positive. In the case where the end of the SSSplit sentence was found in a later sentence, within the set of 'true' sentences, it meant that our sentence spanned too wide, or that the system had "undersplit". These cases we considered to be false negatives, as we had failed to detect a sentence boundary where there was one.

| | RASP | Owusu | SSSplit |
|---|---|---|---|
| Precision | 0.994 | 0.996 | 0.964 |
| Recall | 0.983 | 0.990 | 0.994 |
| F-measure | 0.988 | 0.992 | 0.978 |

Table 1: Comparison of sentence splitters in RASP, Owusu and SSSplit.

Our training consisted of 14 papers in the fields of physical chemistry and biochemistry. A different set of 41 papers distinct from the training set but from the same thematic domain was used as a test set. Out of these 41 papers, 36 feature as a test set (with n-fold validation) also for the sentence splitters RASP (Briscoe et al., 2006) and the XML-aware sentence splitter developed by (Owusu, 2008). The results for all three systems, obtained as medians of Precision, Recall and F-measure for the 36 papers are shown in Table 1.

Precision is the proportion of true positives over all end and start tags returned, giving a measure of the number of boundaries identified correctly. Recall is the proportion of true positives over all the relevant start and end tags in the hand-annotated papers, giving a measure of the number of boundaries actually found. F-Measure combines Precision and Recall to give a more balanced view on the system performance.

In comparison with RASP and the XML-Aware splitter of (Owusu, 2008), SSSplit performed well, though it did not outperform these systems. Their highest result for precision was 0.996 (vs 0.964 for SSSplit) and for recall 0.990 (vs 0.994 for SSSplit). We can explain their higher results somewhat by their use of n-fold cross-validation on 36 out of the same 41 papers that we used, which can allow information from the test set to leak into the training data. We did not perform n-fold cross-validation, as this would have involved going through each of the papers and removing any potential influence on our regular expression rules of the sentences included within, which is a non-trivial process. Our test data was completely unseen, which meant that our eval-

|  | Training | Testing |
|---|---|---|
|  | (1979 sentences) | (5002 sentences) |
| Precision | 0.961 | 0.964 |
| Recall | 0.995 | 0.994 |
| F-measure | 0.96875 | 0.978 |

Table 2: Comparison of SSSplit on the training and testing papers. The training set consisted of 14 papers (1979 sentences) and the testing set of 41 papers (5002 sentences).

uation is stricter, avoiding any influence from the training data.

In addition to the comparison between SSSplit and the other two XML-aware sentence splitters, we also performed a comparison between our training and testing sets, depicted in Table 2.

As can be seen in Table 2, recall was only slightly better on the training set than the test set, but precision was worse on the training set, presumably because of lack of attention being paid to the oversplitting in a particular paper ("b103844n"). This shows that we have not overfitted to the training set in developing our splitter. Our recall is particularly high, indicating that our splitter makes very few false negative errors. We can attribute many of the false positive errors to our somewhat small set of abbreviations considered, resulting in oversplit sentences. We would like to incorporate a more sophisticated approach to abbreviations in the future.

## 5 Performing CISP Annotations

Within the context of the ART project (Soldatova et al., 2007), SAPIENT has been used by 16 Chemistry experts to annotate 265 papers from RSC Publishing journals, covering topics in Physical Chemistry and Biochemistry. Experts have been annotating the papers sentence by sentence, assigning each sentence one of 11 core scientific concepts and linking together sentences across a paper which refer to the same instance of a concept. The aim is to create a corpus of annotated papers (ART-corpus) with regions of scientific interest identified by CISP concepts ("Result","Conclusion", "Observation","Method" and so on).

A preliminary evaluation of the experts' agreement on the ART Corpus, based on a sample of 41 papers, annotated by the 16 experts in non-overlapping groups of 3, shows significant agreement between annotators, given the difficulty of the task (an average kappa co-efficient of 0.55 per group). The details of this work are beyond the scope of the current paper, but the preliminary results underline the usability of both the CISP meta-data and SAPIENT. In the future, we plan to further evaluate the ART Corpus by incorporating existing machine learning algorithms into SAPIENT and automating the generation of CISP meta-data. This would make SAPIENT a very useful tool and would indeed add a lot more value to the meta-data, since training and paying annotators is a costly process and manually annotating papers is incredibly time consuming.

## 6 Conclusion and Future Work

We have presented SAPIENT, a web-based tool for the annotation of full papers, sentence by sentence, with semantic information. We have also discussed how these annotations result in the indirect definition of regions of interest within the paper. The system has been already tested in a systematic study and has been employed for the creation of a corpus of papers annotated with CISP concepts (ART Corpus). In the future we plan to extend SAPIENT so that the system can itself suggest annotation labels to users. We also plan to target the needs of particular users such as authors of papers, reviewers and editors.

SAPIENT, SSSplit and their documentation are both available for download from http://www.aber.ac.uk/compsci/Research/bio/art/sapient/.

## Acknowledgments

We would like to thank Peter Corbett, Amanda Clare, Jem Rowland and Andrew Sparkes for reading and commenting on earlier versions of this paper. We would also like to thank the anonymous reviewers for their useful comments. This work was part of the ART Project (http://www.aber.ac.uk/compsci/Research/bio/art/), funded by the U.K. Higher Education Joint Information Services Committee (JISC).

## References

E. Briscoe, J. Carroll and R. Watson 2006. The Second Release of the RASP System. *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions, Sydney, Australia.*

P. Corbett, P. Batchelor and S. Teufel. 2007. Annotation of Chemical Named Entities. *Proc. BioNLP*.

Nikiforos Karamanis, Ruth Seal, Ian Lewin, Peter McQuilton, Andreas Vlachos, Caroline Gasperin, Rachel Drysdale and Ted Briscoe. 2008. Natural Language Processing in aid of FlyBase curators. *BMC Bioinformatics*, 9:193.

Maria Liakata and Larisa N. Soldatova. 2008. Guidelines for the annotation of General Scientific Concepts *JISC Project Report*, http://ie-repository.jisc.ac.uk/.

Jimmy Lin 2009. Is Searching Full Text More Effective Than Searching Abstracts? *BMC Bioinformatics*, 10:46.

Ben Medlock and Ted Briscoe. 2007. Weakly supervised learning for hedge classification in scientific literature. *45th Annual Meeting of the Association for Computational Linguistics*, 23-30 Jun 2007, Prague, Czech Republic.

P. Ogren. 2006. Knowtator: a Protégé plug-in for annotated corpus construction. *Proceedings of the 2006 Conference of the North American Chapter of the Association For Computational Linguistics on Human Language Technology: Companion Volume: Demonstrations*, New York Press, New York, June 04 - 09, 2006.

Lawrence Owusu. 2008. XML-Aware Sentence Splitter. *MPhil thesis*, Cambridge, UK.

CJ Rupp, Ann Copestake, Simone Teufel and Ben Waldron. 2006. Flexible Interfaces in the Application of Language Technology to an eScience Corpus. *Proceedings of the UK e-Science Programme All Hands Meeting 2006 (AHM2006), Nottingham, UK*

Hagit Shatkay, Fengxia Pan, Andrey Rzhetsky and W. John Wilbur. 2008. Multi-dimensional classification of biomedical text: Toward automated, practical provision of high-utility text to diverse users. *Bioinformatics*, 24(18):2086–2093.

Larisa N. Soldatova and Maria Liakata. 2007. An ontology methodology and CISP - the proposed Core Information about Scientific Papers. *JISC Project Report*, http://ie-repository.jisc.ac.uk/137/.

L. Soldatova, C. Batchelor, M. Liakata, H. Fielding, S. Lewis and R. King 2007. ART: An ontology based tool for the translation of papers into Semantic Web format. *Proceedings of the SIG/ISMB07 ontology workshop.*, p.33–36.

Larisa N. Soldatova and Ross D. King. 2006. An Ontology of Scientific Experiments. *Journal of the Royal Society Interface*, 3:795–803.

S. Teufel and M. Moens. 2002. Summarizing Scientific Articles – Experiments with Relevance and Rhetorical Status. *Computational Linguistics*, 28(4). (preprint)

W. Wilbur, A. Rzhetsky and H. Shatkay. 2006. New Directions in Biomedical Text Annotations: Deifinitions, Guidelines and Corpus Construction. *BMC Bioinformatics*, 7:356.