

Using hand-crafted rules and machine learning to infer SciXML document structure

Ian Lewin

Computer Laboratory, University of Cambridge

Abstract

SciXML is designed to represent the standard hierarchical structure of scientific articles and represents a candidate common document representation framework for text-mining. Such a framework can greatly facilitate interoperability of text-mining tools. However, no publisher actually generates SciXML. We describe a new framework for inferring SciXML from a presentational level of description, such as PDF, using general purpose components such as Optical Character Recognition and expert hand-coded rules and then using supervised machine learning to provide the per-journal adaptation required for the different publication styles embodied in different journals. Adaptation via supervised machine learning can often be hampered by the effort involved in generating the necessary gold standard training material. In our framework, the effort required is substantially reduced by a) the initial processing by expert hand-coded rules which produces a reasonable “first draft” tagging and b) the intuitive nature of the SciXML tags which enables non-expert users to correct first drafts in a fast and efficient manner.

1 Introduction

SciXML is designed to represent the standard hierarchical structure of scientific articles. Its latest incarnation, SciXML-CB¹, is based on an analysis of XML actually generated by scientific publishers in the fields of Chemistry and Biology (Rupp et al., 2006). It is a candidate common document representation framework for text-mining in these fields. The interoperability of text mining tools could be greatly aided by standardization on such a common document framework. We advance our reasons for using SciXML in section 2; but the main focus of this paper is in demonstrating effective translation procedures for turning presentationally formatted text into SciXML through a combination of rule-based and machine learning techniques. Hand-coded rule-based processing generates draft structures for documents that are good enough that non-expert users, using the intuitive tags of SciXML, can easily and quickly correct them into gold standard material. This material can then be used for supervised learning of a tag correction procedure. The learnt procedure can then tag new documents automatically and much more accurately. Adaptability is therefore accomplished with the higher ac-

curacy rates made possible by supervised learning; but also at a relatively low cost in terms of effort and skills necessary for generating training material. We believe that the provision of an effective computational mechanism for generating SciXML from other formats such as PDF is a valuable addition to the text-miner’s toolbox.

The system currently uses Optical Character Recognition technology (OCR) to recover an XML encoding of the presentational structure, followed by the rule-based mechanism for deducing a draft document structure. This is followed by a journal-specific set of rules for correcting errors made by the earlier process. The journal-specific rules are learnt using Transformation Based Learning.

The derivation of SciXML from presentational formats such as PDF is valuable because many traditional scientific publishers remain somewhat reluctant to embrace an open access philosophy and release XML. One recent scholarly study estimated a mean of 9% for the fraction of open access publishing, from a sample of 10 biological and social science disciplines (Hajjem et al., 2005). Furthermore, even if full text is available, for example, as archive material, it is often preserved only in document formats such as PDF. Much text processing research has therefore concentrated on the (relatively) small amount of suitable material that is available, such as MedLine abstracts.

When full-text XML is available, then standard

¹SciXML is originally described in Teufel et al. (1999); software for SciXML-CB is available at the FlySlip project website www.cl.cam.ac.uk/~nk304/Project_Index

stylesheet technology can be used to convert publisher “own-brand” XML into SciXML. A number of conversion routines have been developed and are available with the SciXML-CB download package. This technology necessitates a new stylesheet for each new style, usually one per publisher. The use of presentational formatting such as PDF necessitates a new set of rules, usually one per journal; and it is this overhead that the use of machine learning is designed to overcome. Our objectives are to maximize accuracy whilst also reducing the need for further development effort to a minimum.

We report results which show that the “expert generated” general rules for inferring document structure produce highly variable results, with a mean tag-accuracy of 69.8% (results for individual journals vary from 54.2% to 87.8%). After per-journal error correction through transformation-based learning (TBL), the mean tag-accuracy rate improves to 84.6%, which is an error reduction of almost 50% and the range of accuracies for different journals varies from 76.5% to 94.6%. A further result shows that the first stage processing itself can be improved for previously unseen journals (by a mean of 10.7%) by simple use of TBL upon all the training material.

2 SciXML

Standards are subject to a variety of different pressures. Sometimes these result in “lowest common denominator” standards; sometimes only the highest standards will do. We use SciXML for representing the structure of scientific articles for several reasons. Empirically, we have found the tags to be sufficient to identify the major constituents of scientific articles in a number of disciplines (chemistry, biology, computational linguistics). SciXML is being effectively used in several projects²: Citraz, Sciborg, FlySlip. SciXML is used to provide a common API into different publishers’ XML DTDs for all subsequent text mining operations. We have also used SciXML tags in order re-render scientific articles with additional NLP markup inside a Browser and have found that users find this re-rendering perfectly acceptable (Karamanis et al., 2007).

SciXML tags include TITLE, AUTHOR, ABSTRACT, DIV, HEADING, PARAGRAPH, CAPTION, FOOTNOTE, THEOREM, TABLE and FIGURE. Within these structural divisions, there is text which can be marked up with stylistic tags for italics, bold, superscript and subscript. Hierarchical structure is represented solely by the recursive section (or DIV) element, each of which may have

²see Acknowledgments for project information

a heading element.

3 Description of the System

The first phase of processing is OCR performed over electronic documents (in our case, all in PDF format). We currently use OmniPage Pro 14³ to deliver character level information. Some recent PDF text extraction technology⁴ also includes some stylistic information such as font size and styles.

The output of this phase is an XML representation of the visual or presentational layout. Pages are the top-most structure, followed by zones (these being reasonably graphically distinct units such as a column of text or perhaps a centred block containing the abstract), then paragraphs, lines and characters. The structures are associated with stylistic and positional information, for example, the font style, size and co-ordinates of an individual character. The zones are simply those delivered through the software in automatic mode. We use OCR so that text which is encoded through images remain recoverable; to ensure recovery of Greek characters and because, in the final re-rendering, it is essential to include images that were also present in the original document.

The second processing stage filters and simplifies the XML representation, for example, by removing character level xml structures and re-coding character level style changes using inline xml tags rather than through attributes on the character structure. It also attempts to correct certain characteristic OCR errors on zone boundaries.

The third stage of processing is the expert generated genre-specific tagging of the paragraphs. This is undertaken by a top-down left-to-right traversal of the zone and paragraph tags. During traversal, the program can inspect properties of the current structure including for example, its style, fontsize, the numbers of lines of text in it, and its textual contents. It can also store information for retrieval during subsequent processing. For example, zones that are near the top of the page and at most two lines long are simply declared to be page headers. A caption-initial paragraph is detected through a regular expression matching the leftmost substring of a paragraph. A subsequent paragraph is declared to be a part of the preceding caption if the line spacing remains the same.

The current set of rules in this stage of processing represents a development of the ruleset⁵ reported in

³www.nuance.com/omnipage

⁴See <http://www.jpedal.org> for one such PDF text extraction tool under GPL licence that offers stylistic information

⁵Many thanks to Bill Hollingsworth for allowing us

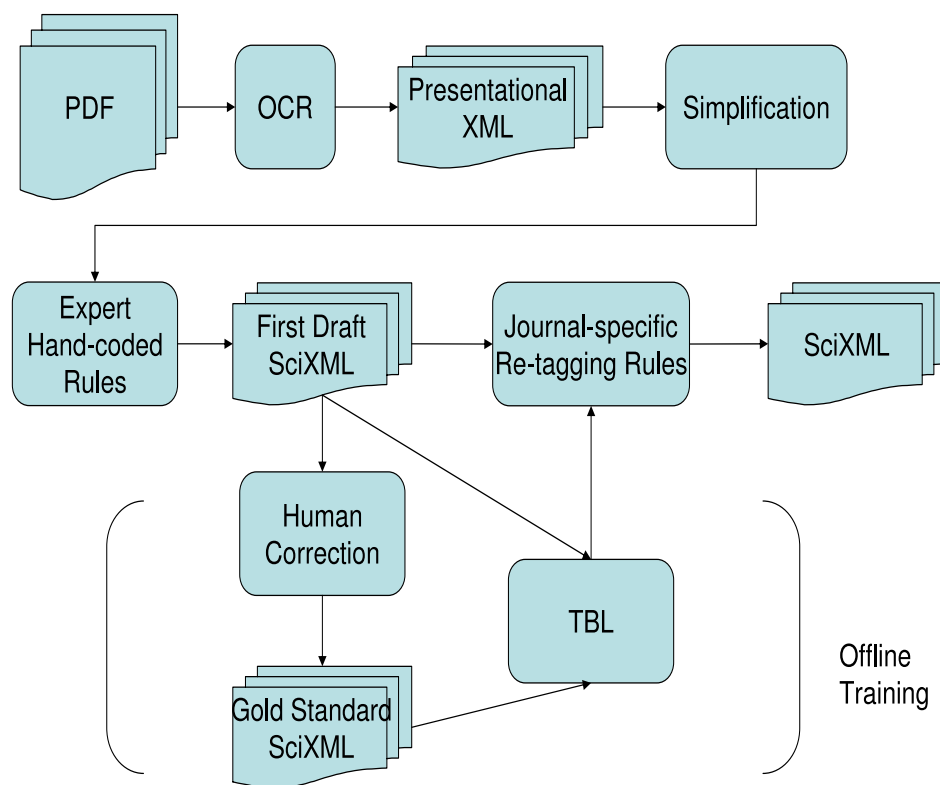


Figure 1: System Structure

Hollingsworth et al. (2005). In that system, there was just a single level of processing and a different set of expertly tuned rules for each journal. Very good results for highly tuned rule-sets were reported (with an F-score of 96.8 for the best); but the cost of development effort required in tuning the rule sets is quite prohibitive.

Rather than hand-tuning a complete rule-set for each journal, we decided to sub-divide the tagging into two stages. The first sub-part could use the “best parts” of the expertly tuned rules to derive what was expected to be a reasonably accurate initial tagging of the data. This expectation proved accurate. The task required an examination of the different expertly coded rule-sets and an extraction of the most generally shared parts of those rules. A new second tagging phase could then be attempted and this was intended to use a machine learning paradigm in order to remove the requirement for continual development effort for each new journal. TBL was chosen as a likely effective method in this

environment. TBL has proven successful in other areas, especially part of speech tagging in which tag-sequencing information can play a role, but one also wants to take into account other features of relatively *local* events (Brill, 1995). TBL also has the merit of using all available training data for each decision and delivering reasonably humanly interpretable outputs.

Consequently, we have a final fourth stage of processing in which a set of journal-specific tag transformations derived by TBL is applied. The transformations are learnt from gold standard annotated data. The accuracy of the initial tagging is sufficiently good to make this task not unduly onerous (on average there are 135 paragraphs per article to re-tag). Furthermore, the annotation task can be undertaken simply by someone with an understanding of the tags rather than by a programmer attempting to modify the use of various cues for paragraph classification during a tree traversal. The tags themselves (see below) are quite intuitive. In section 6 we report figures demonstrating the success of this strategy.

to use his PTX software for interpreting OmniPage XML outputs

4 Humanly Tagging OCR outputs

Although SciXML includes recursive structure through the DIV tag, for our purposes, we completely flatten even this minimal recursive structure by omitting “DIV” sections and tagging only HEADINGS and SUBHEADINGS. DIVs are therefore implicitly defined as the extents between HEADINGS (or SUBHEADINGS). We currently ignore subsections. It is advantageous to flatten the tags because, first, this makes the human generation of the gold standard tagging for a new journal much simpler. A document only consists of a sequence of tags, each being a tag over a simple paragraph of text. No decisions on the *extents* of tags have to be taken. Indeed, the hardest decision required is probably whether a (graphical) paragraph of ordinary text actually continues a linguistic paragraph started on a previous page or column. This can be particularly difficult when the graphical paragraph in question also begins with a new sentence. A further advantage is simply that the re-tagging problem becomes amenable to machine learning techniques for simple sequencing tasks.

Owing to the graphically based output of OCR, the tags also include some tags not included in SciXML, in particular PATTACH, PAGEHEADER and PAGEFOOTER. PATTACH denotes a graphically based “paragraph” which logically continues a previous paragraph. In order to generate SciXML, a sequence of PATTACH elements following a PARAGRAPH element is collapsed into a single PARAGRAPH element. PAGEHEADER and PAGEFOOTER elements could simply be omitted from the final SciXML document, although in practice we keep the contents so that, when papers are re-rendered, all the original textual content appears on the screen.

5 Experimental Evaluation: Method

Our framework is currently deployed in the context of the FlySlip project which has generated an NLP enhanced Browser for scientific articles for the FlyBase curation team. FlyBase is the world’s leading genetic database for *Drosophila* (Drysdale et al., 2005). In order to test our framework, we chose 5 journals⁶ from the “top ten” list of the FlyBase curation team watch-list and then randomly selected 25 articles from these for the years 2004 and 2005. We passed the 25 articles through the first two stages of our system (OCR and initial simplification) and this generated 3396 paragraphs of text to be classified. We then manually recorded the correct classification for each paragraph (the

Gold Standard). The correct classification included 4 tags not generated by the first stage tagging program: ABSTRACT, ABSTRACT-HEADING, SUBHEADING and BAD. A paragraph is classified as BAD if OCR destroys the real paragraph structure in a document by joining areas in distinct paragraphs together in such a way as to destroy the reading order of the text. For example, if double column text is treated as single column text by OCR then this is classified as BAD, as are strings of characters labelled as text by OCR but actually extracted from inside images or tables. If OCR merely overlooks a paragraph break however, then this is not classified as BAD. Also, a paragraph is not BAD merely in virtue of character recognition errors within it. We also removed anything appearing within the references section from our input documents as the section itself was reliably detectable but OCR made many errors on the paragraph structure within it.

To carry out TBL, we used $\mu - TBL$, configured with its implementation of the “Brill” algorithm (Lager, 1999). The algorithm calculates which possible transformation rule reduces the error rate most effectively, adds this rule to its ordered rule set, applies this rule to the training data and then iterates. Termination occurs when no more effective transformations are possible. $\mu - TBL$ also has two parameters which determine how effective a candidate rule has to be in reducing error rates before it is accepted. The scoring threshold (*ST*) indicates how many positive instances of a rule are required for a rule to be acceptable and the accuracy threshold (*AccT*) indicates what proportion of successful re-taggings are required. 100% accuracy would mean that only rules that never cause a correct tag to be changed are acceptable. A scoring threshold of 1 would mean that even a rule that has only one instance in the training data should be accepted.

The training data consisted of the gold standard annotation of each paragraph plus the following features

fontsize One of 5 categories ranging from *huge* to *tiny*, depending on the fontsize being equal or greater than one of 1700, 1200, 1000, 800 or 0 points.

fontstyle One of: italic, bold, bold+italic or unstyled.

zoneboundary One of 3 categories indicating whether the paragraph terminates, begins or is in the middle of a zone.

pageboundary One of 3 categories indicating whether the paragraph terminates, begins or is in the middle of a page.

⁶Development Biology, Development, The EMBO journal, Genetics, Journal of Biological Chemistry

indent A binary value indicating whether the paragraph is indented or not

lines One of 4 categories: 0, 1, 2 or 3+ indicating the number of lines in the paragraph

The “Brill” algorithm operates over a space of candidate rules for correcting tag assignments. Our candidate rules considered rules according to the following templates

Change tag X to Y ...

- 1 if the previous (next) tag is Z
- 2 if the current paragraph is a zone (page) boundary
- 3 if the value of the current fontsize (fontstyle, indent, lines) is Z
- 4 any binary (ternary) combination of the above

In order to assess the benefits of TBL in this environment, we first assessed the “raw” performance of our expert coded rules for each of our 5 journals. Secondly, we ran TBL separately on each of the journals and assessed the improvement (if any) that occurred as a result of TBL. This measures the expected utility of using journal specific retagging. Since the data sets are smaller for each individual journal, we used “leave one out” cross validation in which, for a data set of 5 articles, we carried out 5 separate tests in each of which one article is treated as test data and the remainder as training data. The results are then averaged.

Finally, we pooled training data from different journals and assessed the improvement (if any) that occurred when testing the result on data from a previously “unseen” journal. Again, this was carried out using “leave one out” cross validation.

6 Experimental Results

Figure 2 (pre-training) shows the baseline performance for each of our 5 journals (5 articles each) with no journal specific training having been applied. The mean across all 5 journals is 69.8% with a standard deviation of 12.8, which is high.⁷

Figure 2 (post-training) shows the performance achieved after journal specific training has been applied at the best performing thresholds (AccT = 0.4; ST = 2). Thus, 91.8 is the average accuracy

⁷These baseline figures are not directly comparable with figures reported in Hollingsworth et al. (2005) as the evaluation methodology differs. They use a gold standard where the extents of text that are tagged (as well as the tags themselves) may differ from those that the system delivers

	dbio	dev	emb	gen	jbc	Mean
pre	73.8	61.3	54.2	72.0	87.8	69.8
post	91.8	83.3	76.9	76.5	94.6	84.6

Figure 2: within-journal pre & post training accuracy (AccT=0.4,ST=2)

achieved in 5 “leave one out” tests from 5 articles in *Developmental Biology* and represents a 24.4% improvement over the baseline. On average, performance shows an increase of just over 20%. A one-tailed paired t-test (*i.e.* pairing journals pre and post-training) gives a P-value of 0.009 for the null hypothesis that training has no effect. The table also shows how the spread of results changes as a result of training. The standard deviation reduces to 9.0. (Inference about standard deviations are almost certainly not reliable in this context Moore and McCabe (1989))

Varying the thresholds for rule acceptance does have an effect although not a terribly strong one. Figure 3 shows data for a varying scoring threshold, again for journal-specific training. Requiring at least 2 positive instances to justify a rule, has a positive effect; but more than this is somewhat detrimental. Varying accuracy thresholds appeared not to have a significant effect (data not shown). However, at all thresholds, there was some positive effect. Furthermore, the greatest effects apply to the least well-performing journals, although naturally there is less room for improvement in the journals for whom the first stage processing already succeeds reasonably well.

Figure 4 shows the improvements resulting from pooling training data and testing on data from a journal not included in the pooled data. The results demonstrate that a positive effect can be obtained for articles from previously unseen journals by simply training on all existing data from other journals. Interestingly, in this scenario not all thresholds deliver improvements. Candidate rules with less evidence in training data from *other* journals are less likely to generalize well. Variations in accuracy thresholds also had a greater impact on test performance.

In a further experiment, we also tested the impact of journal specific training, having first carried out the general training. Perhaps unsurprisingly, the results were nearly identical to the earlier results obtained simply from using a model trained on journal-specific data only, although the rule-sets that resulted naturally differed somewhat.

To illustrate the types of rules derived, Figure 5 shows the first rules derived for two different journals *Development* and *Developmental Biology*.

	1	2	3	4
dbio	92.54	91.84	91.8	90.76
dev	82.72	83.26	83.06	82.04
emb	75.26	76.9	72.54	72.54
gen	76.28	76.52	75.18	75.24
jbc	94.28	94.58	93.32	93.62
All	84.216	84.62	83.18	82.84

Figure 3: within-journal accuracies at scoring thresholds 1 to 4 (AccT=0.4)

	dbio	dev	emb	gen	jbc	All
1	6.5	14.2	2.6	-9.7	-10.1	0.7
2	9.1	15.4	6.0	1.5	0.3	6.5
3	7.8	19.6	7.0	3.1	0.9	7.7
4	7.3	30.6	11.8	3.1	0.9	10.7

Figure 4: cross journal % changes at varying scoring thresholds (AccT=0.4)

Column 1 shows the number of instances changed by the rule. Column 2 shows the rule accuracy. Column 3 shows the rule itself using the syntax ‘ $X \rightarrow Y$ IF $f_1:v_1@[p_1]$ & ...’ which is to be read as: change tag X to Y if the value of feature f_1 is v_1 at tag position p_1 , and so forth. Tag position 0 is the current tag, the previous tag is -1, the next tag is +1 and the tag position is 0, unless stated otherwise.

The rules show that, for both journals, the first rule is to detect subheadings, which is a tag not output at all by the first stage processing. In *Development*, the style of subheadings is bold; whereas indentation is the cue for *Developmental Biology*. In fact, subheadings are italicized in this journal but this cue is no more effective (possibly less effective) than that using indentation. In *Development*, the text immediately following a subheading is not indented and, since the subheading itself is not punctuation terminated, earlier processing has incorrectly classified this text as a PATTACH. This “feature” becomes part of the cue for spotting the subheading itself (other bold items may not be so followed) and is also immediately corrected by the next rule which changes all PATTACH that follow a subheading into a P. In both sets of rules, page-headers are the target of the next rules to apply for which, in *Developmental Biology*, fontsize is a good cue.

7 Related Work

The idea of a standard document format for text-mining has been noted before, for example IeXML (Rebholz-Schuhmann et al., 2006). IeXML however only considers standardization at the level of

sentence structure and below. The Genia project⁸ has also generated a document format which consists of text *excerpts* that are extracted from the original document and then linked back to it. In this way, text can be mined and marked up but links to the original document maintained. In addition, if the original document is updated, then required updates to the additional markup can be discovered and made efficiently. However, the relation of the extent of an excerpt to the document structure is not altogether clear. Some structural information may be available within excerpts (and possibly not in a standardized format); for others, one may have to traverse links between excerpts and links back to the repository. Whether SciXML really is an acceptable API for all text-miners, or whether, for example, one should always be prepared to handle all constructs of the multi-faceted NLM archive DTD, remains an open question.

Several other research efforts have considered the problem of Information Extraction from text encoded in (semi) proprietary formats such as PDF, e.g. Multivalent (2006); Thoma (2001); Mathiak and Eckstein (2004); Corney et al. (2005); Miller et al. (2004). The quality of text extraction required naturally depends partly on the intended end application. For example, if Information Retrieval (document selection) is the intention, then it remains an open question whether NLP techniques can add value to word-based metrics (Lewis and Jones, 1996). In such a case, preserving the reading order of the text and recovering document structure is not so important. BioRAT is designed for research scientists who may navigate their topic using extracted facts only, or who may revert to the original source when an interesting fact is detected. Consequently, document structure is not of great importance but reading order is very important as IE patterns are defined over it. For the reasons given earlier, we wish to preserve reading order and as much document structure as possible.

A variety of previous work discusses the potential advantages of using OCR versus PDF text extraction techniques. Generally, PDF text extraction techniques have often been considered poor at recovering one or more of: foreign characters, text stylistics, column detection in multi-column documents, text encoded via graphics and positional information. However, clearly this is a moving field. A rather different use of TBL for document structure determination was described by Curran and Wong (1999). Here, the input documents are structured HTML and the outputs are structured XML. For example, an HTML rendering of a bibliogra-

⁸www-tsujii.is.s.u-tokyo.ac.jp/GENIA/

Rules for *Developmental Biology*

43 1.00 p→subheading IF tag:p@[1] & indent:0
14 1.00 pattach→pageheader IF lines:1 & firstfont:small
14 1.00 heading→pageheader IF firstfont:tiny
9 1.00 subheading→pattach IF lines:'3+' & firststyle:u
7 1.00 p→pageheader IF lines:1 & firstfont:small
4 1.00 subheading→pfloat IF firststyle:u & firstfont:small
4 1.00 pfloat→author IF firstfont:large
4 1.00 p→pattach IF lines:'3+' & indent:0
4 1.00 heading→abstractheading IF firstfont:small

Rule for *Development*

44 1.00 p→subheading IF tag:pattach@[1] & firststyle:bold
33 0.88 pattach→p IF subheading@[-1]
28 1.00 pattach→pageheader IF lines:1
10 1.00 pfloat→pageheader IF zoneboundary:left
5 1.00 p→pattach IF indent:0 & zoneboundary:left
4 1.00 pattach→abstractattach IF firststyle:bold
4 1.00 p→abstract IF heading@[-1] & firststyle:bold
4 1.00 heading→abstractheading IF tag:pfloat@[-1]

Figure 5: Example rules learnt for two different journals (AccT=0.4,ST=2)

phy (in which journal titles might be italicized and volume numbers shown in bold) might be translated into suitable XML description. The algorithm begins from a (text) alignment of example HTML renderings and corresponding XML structures and learns a general translation algorithm. Overall, the scheme represents a very different scenario where the input and output tag vocabularies are actually disjoint. Furthermore, since the structures may differ substantially, this instantiation of TBL requires not just a re-tagging of existing tags, but insertion and deletion of tags and the ability to handle several tags beginning at one textual point. No evaluation of tagger accuracy appears to have been published. However, this line of research represents one possible way to tackle the overhead involved in generating new stylesheets for converting publisher XML into SciXML.

8 Conclusions and Further Work

We have described a system for generating SciXML structured text from PDF documents. SciXML is a promising candidate for a common document structure framework that could aid the interoperability of text-mining components. The system includes a cascade of processing in which more and more domain-specific processing is applied. First, fully general OCR software is used to obtain a visual or presentational description of the document. Subsequently, a hand-coded set of rules designed for the genre of interest (here, scientific articles in biology) is deployed. Finally, in order to modify system behaviour for the varying styles of differ-

ent journals within a genre, we use a supervised learning technique, transformation based learning, to derive a journal specific set of rules. The latter stage of processing has proven highly effective at correcting errors from the previous stages; without requiring a prohibitive cost of expert tuning or training data collection. The judicious use of a markup (SciXML) that contains easily understood tags and of an expert coded rule-set that can provide a reasonable first draft tagging enables an effective adaptation strategy for previously unseen document styles.

There are a variety of possible extensions to our work. Our current dependence on OCR is clearly one we would like to remove. OCR is not always perfect although it is not entirely clear to us how much performance would improve if it were. Re-display of a document scanned by imperfect OCR would clearly be directly impacted by quality of OCR. However, structure determination can also be affected. For example, OCR can sometimes detect an extremely large number of very small paragraphs even though, to the human eye, there is just one. Although the PATTACH tag discussed above provides a mean to reconstruct the text, and generally, this mechanism succeeds, the number of these “corrections” may have effects elsewhere during training. The correction of “bad” material could undoubtedly be improved and probably ought to be handled prior to structure determination.

One attractive possibility is to use the stylistic information that is now available from PDF text extractors in order to generate SciXML.

We are also interested in whether we can detect natural sub-classes amongst journals within our domain and whether we could reasonably reliably classify unseen articles into those styles. Again, by doing this we might be able to improve overall performance on new styles; and also thereby reduce even further the amount of training that users have to perform. We are of course aware that the “one journal-one style” rubric that we have adopted is only a temporary working hypothesis. Journal styles change over time. Furthermore, some articles, such as review articles often exhibit a markedly different style from standard publications of research results.

Acknowledgments

This work was primarily supported under BB-SRC grant reference BBS/B/16291 FlySlip (see www.cl.cam.ac.uk/~nk304/Project_Index). Associated software has been developed under two EPSRC projects: Sciborg and Citraz (see www.sciborg.org.uk & www.cl.cam.ac.uk/~sht25)

References

- Brill, E. 1995. Transformation-based error-driven learning and natural language processing: a case study in part-of-speech tagging. *Comput. Linguist.* 21(4): 543–565.
- Corney, D., D. Jones, and Buxton B. 2005. *BioRAT: A search engine and information extraction tool for biological research*. <http://bioinf.cs.ucl.ac.uk/biorat>: World Wide Web.
- Curran, J. R., and R. K. Wong. 1999. Transformation-Based Learning in Document Format Processing. In *Proc. AAAI 1999 Fall Symposium on on Using Layout for the Generation, Understanding or Retrieval of Documents*, 75–79. AAAI Technical Report FS-99-04.
- Drysdale, R.A., M.S Crosby, and The Flybase Consortium. 2005. FlyBase: genes and gene models. *Nucleic Acids Research* 33: D390–D395.
- Hajjem, C., S. Harnad, and Y. Gingras. 2005. Ten-Year Cross-Disciplinary Comparison of the Growth of Open Access and How it Increases Research Citation Impact. *IEEE Data Engineering Bulletin* 28(4): 39–47.
- Hollingsworth, B., I. Lewin, and D. Tidhar. 2005. Retrieving Hierarchical Text Structure from Typeset Scientific Articles - a Prerequisite for E-Science Text Mining. *Proc. 5th E-Science All Hands Meeting (AHM2005), Nottingham* 267–273.
- Karamanis, N., I. Lewin, R. Sealy, R. Drysdale, and E Briscoe. 2007. Integrating Natural Language Processing with FlyBase Curation. In *Proceedings of the Pacific Symposium in Biocomputing 2007*, vol. 12, 245–256.
- Lager, Torbjrn. 1999. The μ -TBL System: Logic Programming Tools for Transformation-Based Learning. *Proc. 3rd Intl Wkshp on Computational Language Learning, Bergen* 190–201.
- Lewis, David D., and Karen Sparck Jones. 1996. Natural Language Processing for Information Retrieval. *Communications of the ACM* 39(1): 92–101.
- Mathiak, B., and S. Eckstein. 2004. Five Steps to Text Mining in Biomedical Literature. *Proc 2nd European Wkshp on Data Mining and Text Mining in Bioinformatics*.
- Moore, D., and G McCabe. 1989. *Introduction to the Practice of Statistics*. W.H. Freeman and Company.
- Multivalent. 2006. [Http://multivalent.sourceforge.net](http://multivalent.sourceforge.net).
- Miller, H. M., E. E. Kenny, and P. W. Sternberg. 2004. Textpresso: an ontology-based information retrieval and extraction system for biological literature. *PLoS Biol* 2(11).
- Rebholz-Schuhmann, D., H. Kirsch, and G Nenadic. 2006. IeXML: towards a framework for interoperability of text processing modules to improve annotation of semantic types in biomedical text. In *Proc.BioLink, ISMB 2006, Fortaleza, Brazil*.
- Rupp, C.J., A. Copestake, S. Teufel, and Waldron B. 2006. Flexible Interfaces in the Application of Language Technology to an eScience Corpus. *Proc. 6th E-Science All Hands Meeting (AHM2006), Nottingham*.
- Teufel, S., J. Carletta, and M. Moens. 1999. An annotation scheme for discourse-level argumentation in research articles. In *Proc. EACL*.
- Thoma, G. 2001. Automating the production of bibliographic records for MEDLINE. Internal R&D report, CEB, LHCNCB, NLM.