Keyword and Metadata Extraction from Pre-prints

Emma Tonkin[1]; Henk L. Muller[2]

[1] UKOLN, University of Bath, UK
e-mail: e.tonkin@ukoln.ac.uk
[2]University of Bristol, UK
e-mail: henkm@cs.bris.ac.uk

Abstract

In this paper we study how to provide metadata for a pre-print archive. Metadata includes, but is not limited to, title, authors, citations, and keywords, and is used to both present data to the user in a meaningful way, and to index and cross-reference the pre-prints. We are particularly interested in studying different methods to obtain metadata for a pre-print. We have developed a system that automatically extracts metadata, and that allows the user to verify and correct metadata before it is accepted by the system.

Keywords: metadata extraction; Dublin Core; user evaluation; Bayesian classification

1. Introduction

There are two methods for obtaining metadata: the metadata can be mechanically extracted from the pre-print, or we can ask a person (for example the author or digital librarian) to manually enter the metadata. The former approach, automated metadata generation, has attracted a great deal of attention in recent years, particularly for the role that it is expected to play in reducing the *metadata generation bottleneck* [1] - that is, the difficulty of producing metadata in a timely manner. Much of this interest arises from prior work in *machine-aided indexing*, or *automated indexing* - that is, either software-supported or entirely software-driven indexing approaches. The difference between machine-aided or automated indexing and automated metadata generation or extraction approaches is, as seen by the authors, simply that the metadata is here seen as an end in itself; we aim to emulate well-formed metadata generation, and do not concern ourselves greatly here with the subsequent question - evaluation of the usefulness of this metadata for a given purpose.

Greenberg et al [2] describe two primary approaches to metadata generation, stating that researchers have experimented primarily with document structure and knowledge representation systems. Document structure involves the use of the visual grammar of pages, for example, making use of the observation that title, author(s) and affiliation(s) generally appear in content header information. Such metadata can be extracted via various means, for example using support vector machines upon linguistic features [3], a variable hidden Markov model [4], or a heuristic approach [5]. [6] describe an approach that primarily utilizes formatting information such as font size as features, and makes use of the following models: Perceptron with Uneven Margins, Maximum Entropy (ME), Maximum Entropy Markov Model (MEMM), Voted Perceptron Model (VP), and Conditional Random Fields (CRF): they note that an advantage of an approach that primarily makes use of visual features is the ease of generalisation to documents in languages other than English. This approach, however, focuses solely on the problem of extracting the document title.

The relevance of knowledge representation systems for Greenberg et al is the increasing availability of resources that can be useful to the process of metadata generation, or indeed the harvesting of existing

metadata registries; this is primarily of use in post-processing or enhancement, although such knowledge basis additionally provide a useful resource under many circumstances. For example, an authoritative but incomplete author name database can be used firstly for automatic name authority control, and secondly as an excellent basis for training of supervised machine learning systems in detection of fields containing author names. The issue of post-processing is, however, out of the scope of this paper, and will therefore be referred to only briefly.

Recent work on the Semantic Web and on classification and knowledge management has focused on the extent to which these methods lead to equivalent or stable results. Whilst the two approaches may have compatible outcomes in terms of the type of metadata output, they depend upon very different underlying mechanisms. Factual metadata such as title and author is usually unambiguous; but other metadata, such as keywords for classification, is of an interpretative nature. User entered classifications can be seen as based around a set of prototype *concepts* [7,8]. Mechanically generated classifications are generally built around an identified set of *features*. The features that are used by the mechanical system are meant to form a basis for making similar judgements to those given by a human, and hence are intended to emulate similar behaviour to the set of concepts recognised by the user; but they are in practice quite different, for they are based around a range of heuristics or learnt statistical measurements rather than a deeper understanding of the information within the data object. Because of this difference, care must be taken to ensure that the judgements are compatible, typically by choosing supervised methods, that may be trained and verified against reference data (ground truth).

2. Available metadata

An electronic copy of a document is potentially a rich source of metadata. Some of the metadata is presented in an obvious manner to the reader, for example the title of a document, the number of pages and the authors. Other metadata is less obviously visible. Attributes of the eprint such as format - intrinsic document properties - can be automatically detected with ease [9]. The class of a document - that is, whether it has been peer-reviewed, whether it appeared as a conference paper, article, journal article, technical report or PhD/Masters' thesis - is often unclear. The theme, subject matter and contributions contained within the document should be visible within the text, for this is after all the rationale behind making the document available at all, but a great deal of domain knowledge may be required to extract such information and recognise it for what it is.

We focused on five general structures that can be examined in order to extract metadata:

- The document may have structure imposed on it in its electronic format. For example, from an HTML document one can extract a DOM tree, and find HTML tags such as <TITLE>.
- The document may have a prescribed visual structure. For example, postscript and PDF specify how text is to be layed out on a page, and this can be used to identify sections of the text.
- The document may be structured following some tradition. For example, it may start with a title, then the authors, and end with a number of references.
- Documents that are interlinked via citation linking or co-authorship analysis may be analysed via bibliometric methods, making available various types of information.
- The document will have linguistic structure that may be accessible. For example, if the document is written in English, the authors may "conclude that xxx .", which gives some meaning to the words between the conclude and the full stop.

There exist in practice a huge number of features by which to describe a complex object such as an eprint. Readers effortlessly identify and use relevant subsets and combinations of these on a daily basis, but not all of those features are actually intrinsic to the document or the specific instance of the document (the file).

2.1 Formatting structure

Certain document types contain structural elements with relatively clear or explicit semantics. One of the potential advantages of a language like HTML that stresses document structure over a language such as Postscript that stresses document layout, is that given a document structure it is potentially feasible to mechanically infer the meaning of parts of the document.

Indeed, if HTML is used according to modern W3C recommendations, HTML is to contain only structural information, with all design information contributed in CSS. This process of divorcing design from content began in the HTML 4.0 specification [10]. Under these circumstances, a large amount of information can potentially be gained by simply inspecting the DOM tree. For example, all headers H1, H2, H3, ... can be extracted and they can be used to build a table of contents of the paper, and find titles of sections and subsections. Similarly, the HEAD section can be dissected in order to extract the title of a page, although this may not contain the title of the document.

However, given that there are multiple ways in HTML to achieve the same visual effect, the use of the tags given above is not enforced. Many WYSIWIG tools use alternative means to produce a similar visual impression – for example, generating a <P class='header2'> tag rather than a H2 tag. Since the semantics of these alternatives are less clear, this makes extraction of data from HTML pages in practice difficult. A technical report by Bergmark [5] describes the use of XHTML as an intermediate format for the processing of online documents into a structure, but concedes that, firstly, most HTML documents are 'not well-formed and are therefore difficult to parse'; translation of HTML into XHTML resolves a proportion of these difficulties, but many documents cannot be parsed unambiguously into XHTML. A similar approach is proposed by Krause [11].

In this paper we ignore any context markup, and we have focussed on documents that are not presented in a structure language. On examination of Bergmark's metadata extraction algorithm, it seems likely that a robust metadata extraction from XHTML makes relatively little use of formatting information.

Figure 1: Visual structure of a scientific paper

2.2 Visual structure

In contrast to HTML, other methods to present documents often prescribe visual structure rather than document structure. For example, both Postscript and PDF specify symbol or word locations on a page, and the document consists of a bag of symbols or words at specific locations. Document structure may be inferred from symbol locations. For example, a group of letters placed close together are likely to be a word, and a group of words placed on the same vertical position on the page may be part of a sentence in a western language.

The disadvantage of those page description languages is that there are multiple ways to present text, for example, text can be encoded in fonts with bespoke encodings; the encoding itself has no relation to the characters depicted, and it is the shape of the character which conveys the meaning. In circumstances like this it is very difficult to extract characters or words, but the visual structure itself can still be used to identify sections of a document. For example, Figure 1 shows a (deliberately) pixelated image of the first page of a paper, and even without knowing anything about the particular characters, four sections can be highlighted that almost certainly contain text (red), authors (green), affiliation (yellow) and abstract (blue).

Indeed, it turns out that visual structure itself can provide help in extracting sections of an image of, for example, legacy documents that have been scanned in. However, it is virtually impossible to distinguish between author names above the title and author names below the title, if the length of the title and the length of the author block are roughly the same.

We have performed some experiments that show that we can extract bitmaps for the title and authors from documents that are otherwise unreadable — 3-6% of documents on average in a sample academic environment [12]. An approximately 80% degree of success is achievable using a simple image segmentation approach. These images, or indeed the entire page, may alternatively be handed to OCR software such as gOCR for translation into text and the resulting text string processed appropriately. An account of the use of appearance and geometric position of text and image blocks for document analysis and classification of PDF material may be found in Lovegrove and Brailsford [13], and a rather later description of a similar ‘spatial knowledge’ approach applied to Postscript formatted files is given by Giuffrida et al [13].

In this paper we focus on documents from which we can extract the text as a simple stream of characters.

2.3 Document structure

From both structured description languages (such as HTML) and page description languages (such as PDF) we can usually extract the text of the document. The text itself can be analysed to identify metadata. In particular, author names usually stand out, and so do affiliations, and even the title and journal details.

The information that can be extracted from the document structure includes:

1. Title
2. Authors
3. Affiliation
4. Email
5. URL
6. Abstract
7. Section headings (table of contents)
8. Citations
9. References

10. Figure and table captions eg. [15]
11. Acknowledgments [16]

Extracting these purely from the document structure is difficult, but together with knowledge about words likely found in, for example, author names or titles, the extraction is feasible. A detailed discussion on the methods that we use can be found later on in this paper.

2.4 Bibliographic citation analysis

There exists a widespread enthusiasm for bibliometrics as an area, which depends heavily on citation analysis as an underlying technology. Some form of citation extraction is a prerequisite for this. As a consequence, a number of methods have been identified for this approach, making use of various degrees of automation. Harnad and Carr [17] describe the use of tools from the Open Journal Project and Cogprints that can, given well-formed and correctly specified bibliographic citations, extract and convert citations from HTML and PDF. Citation linking is of interest to many as a result of the potential of this data in analysis of impact and, arguably, value of scientific papers, but other uses of the information exist, in particular in the area of interface design and support for information-seeking practices.

The nature and level of interlinking between documents is a rich source for information about the relations between them. For example, a high rate of co-citation may suggest that the subject area or theme is very similar. In this instance, we extracted citations via our software; these could potentially be used for various purposes. For example, Hoche and Flach [18] investigated the use of co-authorship information to predict the topic of scientific papers.

The harvesting of acknowledgements has also been suggested as a measure for an individual's academic impact [16], but may also carry thematic information as well as information on a social-networking level that could potentially be useful for measuring points such as conflict of interest.

Along with content classification, this constitutes part of a toolkit for 'similarity search' [9].

2.5 Linguistic structure

Finally, the document can be analysed linguistically, inferring meaning of parts of sentences, or relationships between metadata. For example, citations in the main text may be contained within the same sentence, indicating that the two citations are likely to be related in some way. The relation may be a positive relationship or a negative relationship, depending on the text around it: *In contrast to work by Jones (1998), work by Thomas (1999)...*

Analysing linguistic structure depends on knowledge of the document language, and possibly on domain knowledge. Using linguistic analysis one can attempt to extract:

1. keywords
2. relations between citations

Other than Bayesian statistics across term appearance, we do not use explicit linguistic information in the work presented below, but instead focus on the document structure, guided by simple probabilistic information.

3. Uncertainty and metadata

Potential discrepancies between mechanically generated metadata and user-generated metadata may not

be a big problem, because there is also considerable variation in metadata generated by users. There are three principle sources of variation in metadata as generated by humans: typographic errors, different interpretation of the document, and different interpretations of the metadata descriptions. Below we give a description of those three, and a discussion on the consequences of metadata uncertainty.

3.1 Differences in document interpretation

Differences in document interpretation come to light in, for example the consistency of classifying pre-prints using keywords. Neither humans nor computers can index with 100% accuracy. If the same article is indexed by each author and a librarian in turn, then they will probably suggest different indexing terms, stemming from different interpretations of the work, background of the person, knowledge about classifications, and in-depth knowledge of the subject matter. Indexing consistency is a well-known problem of interest to researchers in the domain of information science [19].

Indeed, it is doubtful that there is a "gold standard" classification, for even the author of the article may not agree with appropriate classification keywords. Differing interpretations of the work undoubtedly exist; for example, censorship is generally seen as a primary theme of Bradbury's classic work, *Fahrenheit 451*, an interpretation that the author does not accept. That is, the relevance of a document changes over time, and may not coincide with the author's intention; as this occurs, the keywords associated with a document change over time too. This suggests that either keywords have to be kept up to date, or the interpretation of keywords must depend on the context in which those keywords were assigned.

3.2 Typographic errors in metadata

A common failure mode for a human entering metadata is typographic errors. The frequency of typographic errors depends on system interface, feedback, user profile and the type of metadata. In high-grade metadata that is entered by professionals who are being paid to, say, index scientific works contains very few errors. But low-grade metadata, entered by for example on-line users may contain a significant number of errors. Upper bounds for this value on the tagging system Panoramio was less than 10%, with other tag systems showing far higher numbers.

These errors are not limited to incorrect spellings, but include errors where the metadata value is selected using a drop-down menu the user may select the keyword "above" or "below" the chosen keyword, or spell checkers that have "corrected" a typographic error and have, for example, replaced recking with racking (rather than wrecking). The latter can be a big problem with people who write documents in a non-native language.

In citing other authors, errors in orthography are common, stemming from typographical error, misreading, cultural misunderstanding (such as the inversion of first and last names), as well as from other sources such as issues with citation management software or, indeed, error propagated from replicating prior mis-citation of the document. An overview and typology of features found in online orthography can be found in Tavosanis [20].

Automatically generated metadata does not contain any typos, other than those copied from the original document and those introduced during the extraction process. However, computer generated metadata is subject to different failure mode. In the simplest case, an incorrect keyword is suggested because it appears appropriate on the basis of the *features*, but turns out to be one that is inappropriate to a human who understands that identical words may refer to different *concepts*.

3.3 Different interpretations of metadata schemas

A third common variation in metadata is due to the interpretation of metadata schemas. This expresses itself commonly in the way in which author names are interpreted. Different parts of names have different meanings, and in some cultures the first part of the name may be a family name, whereas in other cultures the first name may be a given name, and there are languages where there are "middle parts" that are part of the surname.

It is virtually impossible to design a metadata scheme that both allows **all** names to be stored in a single canonical format, and that at the same time is unambiguous and easy to use for authors from all different cultural backgrounds.

One strategy around this is that authors names are just opaque strings of characters that warrant no interpretation. These are difficult to match because authors are frequently inconsistent in providing their names, preferring perhaps in certain cases to provide middle initials and in others to give only an initial of one of their given names. Indeed, it is a strategy that is often used consciously by authors to separate their publications in one field from those in another. This strategy may even be deliberately applied to "fool" automatic indexing [20]. Even where authors are consistent, errors in data extraction or journal style guide clashes may cause errors in author name extraction. For example, some article styles require "first" names in citations to be abbreviated to a single letter.

3.4 Propagation of errors

In the general case, we consider metadata generation as an inherently uncertain operation. This implies that metadata should not necessarily be seen as a discrete set of values, but it could be better to represent it as a probability distribution [21,22]. Representing the metadata as a distribution gives us the opportunity to communicate the uncertainty in the suggested metadata to the user. For example, we can select a number of possible keywords based on features of a publication, and communicate which of those keywords are more probable than others.

Once errors in metadata exist, they *propagate*, reinforce similar errors on future pre-prints, introduce seemingly unrelated extra errors, and obfuscate the data presented to the user.

Firstly, a system will normally use previous classifications in order to classify future papers. In our system, paperBase, author-names, title, abstract, and classification of previous pre-prints are being used to predict the classification of new pre-prints. Once a pre-print has been misclassified, future papers may be misclassified in a similar manner.

Secondly, a system typically uses the metadata found in pre-prints in order to establish *connections* between pre-prints. Connections can be made because two pre-prints are written by an author with the same name, because they cite each other, or because they cover a similar subject matter according to the keywords. Those connections can be used to, for example, disambiguate author identities. A missing link or an extraneous link would make the process of reasoning about clusters of related papers increasingly difficult.

Thirdly, the answers of search queries are diluted when errors are introduced. Cascading errors cause a disproportional dilution of search results. This is also true of user-contributed systems in which users may infer the use of classification terms through examining available exemplars.

When machine-generated classifications are provided, they are generally represented as unitary facts; either a document may be described via a keyword, or it may not. Consider the following example of a machine-generated classification:

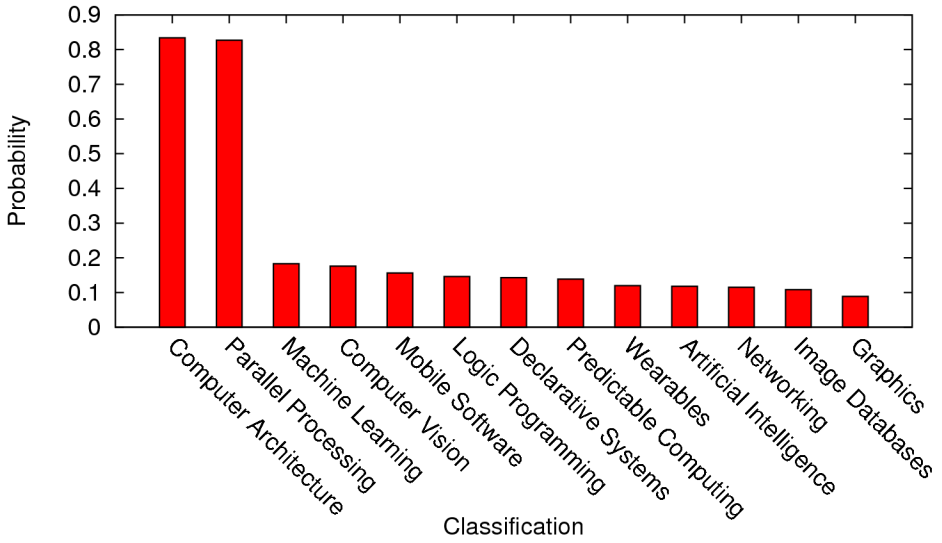

Figure 2: Candidate keywords with associated probabilities

In this case, a document is considered almost certain to be about "Computer Architecture" or "Parallel Processing", and to have a diminishing likelihood of being classifiable as about "Machine Learning" or any of the other terms. In general, a threshold is placed, or the top classification accepted by default, when the result is presented, but it is this distribution that describes the paper with respect to others. The shape of this distribution is very relevant in establishing the nature and relevance of the classification. There may be no clear winner if there are many keywords with similar probability, and then our confidence in the clarity of the results may be shaken absent human evaluation of that judgement.

In the case of classifications, many options may be acceptable, but this is less the case in other situations where uncertainty exists. Consider the following citation parses taken from a sample paper (bold text denotes the title and italic text denotes the author):

- **Confirmation**-*Guided Discovery of First-Order Rules, PETER A. FLACH, NICOLAS LACHICHE*
- **Confirmation-Guided** *Discovery of First-Order Rules, PETER A. FLACH, NICOLAS LACHICHE*
- …
- **Confirmation-Guided Discovery of First-Order Rules, PETER A. FLACH,** *NICOLAS LACHICHE*
- **Confirmation-Guided Discovery of First-Order Rules, PETER A. FLACH, NICOLAS** *LACHICHE*

The likelihood for the correct parse is much higher than the likelihood for all other parses. Unlike the prior example of a classification, only one of these parses can be valid. Whilst it is the most likely, we do not have total confidence in this, but we are able to generate a probability of its accuracy (our level of confidence, a value between 0 and 1). Hence, it is possible to provide some guidance as to the validity of this datum as a "fact" about the document.

The danger of reasoning over data in which we, or the system, have low confidence, is the risk of propagating errors. If we retain a Bayesian viewpoint, we may calculate any further conclusions on the basis of existing probabilities via Bayesian inference. If, however, we treat a probability as a fact and make

inferences over inaccurate data without regard to degree of confidence, the result may be the production of hypotheses over which we have very little confidence indeed.

As a consequence, an extension of DC metadata to include estimates of confidence, as described in [23] is useful, as in the case of classification would be an estimate of the number of classifications considered "plausible"; the breadth or range of likely classifications, which could also be described in terms of variation or level of consistency in judgement - a similar value to that which might be generated in any other situation in which generated or contributed classifications may be treated as "votes", such as collaborative tagging systems.

If the nature and extent of the error are known, further functions that employ these values may apply this information to estimate the accuracy of the result or that of derivative functions. We note that for certain types of metadata, this problem is well-investigated. For example, author name disambiguation has received a great deal of interest in recent years, eg. Han et al [24,25].

4. Prototype

We developed a system for the automated extraction of metadata from pre-print papers known as paperBase. The extractor makes use of the structure that is inherent to scientific papers and Bayesian classifiers in order to identify the metadata. We have captured the structure of scientific documents in a probabilistic grammar that produces most known forms of papers. More details on this grammar are given in Tonkin and Muller [12].

The grammar is used to parse the text of a paper, and this produces a collection of metadata with associated probabilities. The parser takes the path through the grammar that results in maximal probabilities for authors, title, affiliation, email addresses. The individual probabilities can then be used later on to decide how to use the metadata. We extended DC with appropriate attributes for the encoding of those confidence measures, so that, for example, a user interface might visually encode the confidence and highlight fields that are likely to contain errors.

4.1 Visual interface

The interface displays the metadata in a tabbed form, one tab for each type of pre-print. The extracted metadata such as author names, title, journal-name, and suggested keywords are displayed in the tab. The uncertainty that is assigned to each of the suggested keywords is shown by ordering the keywords based on the certainty, and by using a graded colour-coding to indicate probable keywords, providing clear and consistent interface semantics. The keywords are shown in a list with scroll-bar, with the five most likely keywords visible.

4.2 Extension to Dublin Core

The metadata extracted, or in some cases generated, from the document object may be retrieved as an XML document via the paperBase API. The DC metadata itself is encoded into XML using the DC XML guidelines [26] as a basis. Additional terms, including confidence values (probabilities of accuracy) where appropriate for this interface, were included in this document. A fragmentary example of an Open Archives Initiative/Dublin Core XML record as generated by paperBase is below:

<oai_dc:dc>
<dc:type>e-print</dc:type>
<dc:title>An Evaluation Study of a Link-Based Data Diffusion Machine</dc:title>
<dc:creator canonical='Muller HL'>Henk L. Muller</dc:creator>
<dc:creator canonical='Stallard PWA'>Paul Stallard</dc:creator>
<dc:creator canonical='Warren DHD'>David HD Warren</dc:creator>
<dc:description> .... Abstract deleted...</dc:description>
<dc:subject probability='834'>Computer Architecture</dc:subject>
<dc:subject probability='827'>Parallel Processing</dc:subject>
<dc:subject probability='183'>Machine Learning</dc:subject>
<dc:subject probability='176'>Computer Vision</dc:subject>
<dc:subject probability='156'>Mobile Software</dc:subject>
... More keywords ...
</oai_dc:dc>

All keywords are given with a number indicating a calculated probability that the keyphrase is applicable to this document. In this instance, the top two keywords, Computer Architecture and Parallel Processing are good choices, with a high probability (the maximum value is 1000). The next three are less likely, and are, indeed, inappropriate.

The probabilities given are not normalised into confidence values; at this time, there exists no consensus on how confidence values should best be encoded. Therefore, the structure of this record may well change in future.

4.3 Deployment Workflow

As a first trial, we have integrated the system in the institutional repository that stores papers written by members of the Department of Computer Science at the University of Bristol. We adapted the workflow so that authors first have to upload an electronic version of the paper, prior to providing any metadata. When the paper is uploaded the user is presented with a form in which the user can enter the meta-data for that publication.

4.4 Technical details

The extracted data is provided to the end user via a web service. The service is engineered to use web standards common in the *Web 2.0* environment, including REST, Dublin Core and XML. The client interface for the user's web browser makes use of ECMA JavaScript and XML (AJAX) to retrieve the analysed data and place it into a web form.

The webserver has a dedicated thread that interprets metadata. This thread decodes postscript and PDF files, and extracts text from those using the public domain PDFbox Java library (www.pdfbox.org). When the text is extracted it is interpreted in a probabilistic grammar, and the results are stored in a database. Various web services make use of this database, including an independent browse interface along the lines of CiteSeer, and a machine-to-machine REST interface that is used to support AJAX applications requiring document metadata. Others, such as an OAI harvesting interface, can be built against the same database backend - however, as mentioned above (in "Propagation of errors") it is useful for client services to be aware of the data origin and constraints on its use.

An AJAX application embedded into the repository's web interface polls the webserver for metadata, and fills the form in when metadata becomes available. Typically, metadata is available within a few seconds

of submitting the form.

The form will then be filled in asynchronously when the web server has extracted the data. One might regard a synchronous implementation as ideal, where the form comes back when the file has been uploaded. However, since we only have limited computational resources on the web server, and it may take a few seconds for a paper to be completely analysed, we must queue all papers on the server and deal with them one at a time, in order to control congestion. The way in which the queue is handled can be optimised to limit the impact of likely causes of congestion, such as a batch file upload (a usage pattern supported by the service's own internal interface).

As a result, users may have to wait for a few seconds before their form is filled in with the relevant information - however, we think this is beneficial because the user can use this time to familiarise themselves with the form. Providing and filling the form as two asynchronous steps is preferable over a user looking at a spinning hour-glass, and then being taken to a filled-in form. In addition, our system fails gracefully, in that if the decoder service is not working for whatever reason, the form will simply stay with all fields blank and report that no metadata could be extracted.

The accessibility of the resulting software represented a primary concern. As such, care has been taken to ensure that the system functions across multiple browser platforms, including IE, Firefox, Safari, Opera and other Gecko- or KHTML-based browsers such as Konqueror. Cross-browser compatibility is, however, a moving target; hence it is expected that this will impose a small ongoing maintenance cost. The non-availability of JavaScript simply means that the user must fill in each field manually, as was the case before this service became available.

One further accessibility concern for us was the way in which screen readers and similar assistive technologies reacted to the dynamic content placement. The dynamic content proved not to be an issue in practical use; however, the presence of a (non-AJAX) "SELECT MULTI" element fell foul of a known showstopper bug in the screen reader, which meant that we could not complete the evaluation. It seems that fully supporting screen readers would involve at least the level of customisation and maintenance required for cross-browser compatibility, and furthermore this requires additional investment in developing or procuring a software base for testing purposes.

5. Evaluation

We have performed two trials of our system. In one trial we have rebuilt our entire repository, logging suggested keywords together with keywords that were assigned by the author. We show that 80% of the keywords selected by the authors are in the top-five list of keywords. This is a conservative figure, since we expect that some authors would not have picked the other 20% of the keywords if they weren't suggested - see also our discussion earlier on reliability of human indexing.

Throughout the development process, sets of informal think-aloud trials were conducted, that resulted in user feedback; applicable results were included in latter phases of the iterative design process. We then performed a more formal evaluation study on 12 subjects, presenting them with a set of six papers to be entered into a repository. The participant were presented with a form to enter the data, and sometimes this form would be pre-filled in with automatically extracted metadata.

Half the participants had their first three papers entered without assistance, and had automatically extracted metadata for the last three papers. The other half of the participants were presented with automatic data for the first three papers, and had no assistance for the latter three papers. The papers were selected in order to cause maximum trouble for the metadata extractor, in particular, an author name would be extracted twice on one paper, there would be one missing author on a second paper, other papers would have missing ligatures, or mathematical formulas in the abstract.

In order to quantify what a true user would see we have manually judged the quality of title and author extraction on 186 papers. We found that 8% of the titles was completely wrong, and 8% was not completely correct, with the remaining 86% of the titles being right. For the experiment above, this would mean that a participant might have seen one bad title, with a probability of less than 50%. Three bad titles in a row has a probability of 0.4%.

For the authors, 13% of the authors was wrong, of the remaining 87%, 32% included the right authors but had extraneous text that was misconstrued as authors. Our sample was not sufficiently randomised and had many papers by a Slovenian author with a diacritical mark in both surname and first name, which skewed our statistics. In addition, another author's affiliaton was at the "George Kohler Allee" which was misrecognised as an author name.

A detailed analysis of the quantitative results of those study are published in another paper[12]; in short, it was found that the assistive effect generally caused participants to take less time overall in depositing papers. Here, we report on the qualitative feedback that the participants provided. At the end of the trial participants were given a form with four questions, asking which system they preferred, whether they thought that system was faster, whether they thought that system was more accurate, and an "any other comments" box.

A most interesting observation was that the participants were divided on the question of whether manually entered data had fewer errors. Many participants had spotted errors in the automatically corrected data, and had corrected them, and had concluded that the manual data must have been more accurate - however, analysing the errors it turns out that manual data contains more errors. The reasons for this is two-fold. First, there are people who take manual entry literally: they type the title in again (rather than using a copy-and-paste feature). Typing is an error-prone process. Second, people who use the copy-and-paste feature seem to assume that this is by definition error-free - hardly any of the participants spotted that when they had used copy-and-paste ligatures had gone missing during the process, or that hyphenation had been introduced because a word had been broken across two lines in the abstract.

Instead, participants accepted copy-and-paste as a *ground-truth*, and corrected the errors only when the copy-and-paste had been performed by the meta data extractor. We postulate that people have a limited amount of time to perform tasks such as entering publication data, and that they either spend it on manual entry, or on correcting automated entry - the latter leads to more accurate results. There is also a possibility that this is related to the "proofreader blindness" effect - it is known to be more difficult to proofread one's own work than work by others in one's own domain [27]. It is possible that the same effect plays a role in this instance.

Many of the comments that were passed on using the last open question related to features that people would like to see in the system; in particular, we requested a month and a year, and many participants rightly complained that they had to give a month, even if they didn't know it.

A number of comments gave qualitative feedback on the use of paperBase. A telling comment was *Just adding a few fields makes the task of adding your publication much less boring and time consuming. I'd prefer to see it try and occasionally fail as opposed to it be removed because it occasionally failed.*. This has been observed in other studies - users are aware that the task they are doing should be done automatically, and they appreciate any help that a system will give them [28].

Another user commented that *I particularly liked the ordered keywords*. The suggested keywords are either ordered alphabetically, or in some order of likelihood. The latter works really well if the *right* keywords are somewhere in the top-5; if they are not in the top-5 they are very hard to find back, because the ordering is related to the perception of the extraction algorithm, and no longer related to the user entering the publication. Even though people liked it in general, we should have an option to sort the

keywords alphabetically (or have an assisted keyword search) for situations in which the algorithm fails.

One very interesting comment read: *Abstracts are a nuisance; I would remove those from the database.*. Indeed, this participant had blanked out all abstracts - they had not entered any abstract manually and had erased the automatically extracted abstracts. We postulate that they are only a nuisance because of the work involved in entering abstracts - from a search and user interface perspective abstracts are highly valuable and should be available. We think that automated extraction will aid in making meta data more complete - as long as people will not delete valuable information wholesale.

6. Conclusion

Semi automatic meta data entry offers many advantages. From the limited study that we performed, we observed an increased accuracy, faster entry time, and most important, buy-in from the participants unambiguously preferring the semi automated entry system. The evaluation that we performed is limited in that we studied only a single domain (computer science papers), with participants who were very computer literate (postgraduates in computer science), and with only a small number of participants. In future evaluations we would like to include different domains.

The current version of the interface only uses a small amount of the data that could be used. In particular, we do not use links between papers (as found in the form of citations) yet to, for example, disambiguate author identities. The number of file formats supported by the system could be increased, and methods found for the user to correct other metadata such as citations which are also extracted by the system. Equally, the provision and use of error margins may have some promise in providing cross-site, hybrid search operating across a number of resource and metadata types.

One feature of interest within the study results is the reminder that the quality of metadata, whether semi-automated or not, depends on the level of interest of the participant. Individuals who simply do not see the point of providing a given metadata element will at best put little thought into the process, and at worst will actively remove extraneous elements despite the best efforts of an automated metadata extraction service. The ultimate arbiter in any system that is not fully automated is the individual contributor, despite any scaffolding that the system may provide, and any mismatch between the contributor's needs and the aims of the system designer should be identified and allowed for in design and development.

7. Notes and References

[1] Liddy, E. D., Sutton, S. Paik, W., Allen, E., Harwell, S., Monsour, M., Turner, A. and Liddy, J. Breaking the metadata generation bottleneck: preliminary findings. Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries, Roanoke, Virginia, United States, 2001. pp. 464

[2] Greenberg, J., Spurgin, K. and Crystal, A. Functionalities for automatic metadata generation applications: a survey of metadata experts' opinions. Int. J. Metadata, Semantics and Ontologies, Vol. 1, No. 1. 2006

[3] Han, H., Giles, C. L., Manavoglu, E. and Zha, H. Automatic Document Metadata Extraction using Support Vector Machines, Proceedings of the Third ACM/IEEE-CS Joint Conference on Digital Libraries, ACM Press, New York, 2003. pp.37–48

[4] Takasu, A. 'Bibliographic attribute extraction from erroneous references based on a statistical model', Proceedings of the Third ACM/IEEE-CS Joint Conference on Digital Libraries, ACM Press, New York, 2003. pp.49–60.

[5] Bergmark, D. Automatic Extraction of Reference Linking Information from Online Documents. CSTR 2000-1821, Cornell Digital Library Research Group, November 2000.

[6] Hu, Yunhua, Li, Hang, Cao, Yunbo, Teng, Li, Meyerzon, Dmitriy and Zheng, Qinghua. Automatic extraction of titles from general documents using machine learning. Information Processing & Management. Volume 42, Issue 5, September 2006, Pages 1276-1293

[7] Rosch, E.. Natural categories. Cognitive Psychology 4, 1973. pp. 328-350.

[8] Labov, W. The boundaries of words and their meanings, New ways of analysing variation in English. Washington: Georgetown University Press C-J. N. Bailey and R. W. Shuy, 1973. pp 340—373

[9] Olivié, H., Cardinaels, K. & Duval, E.. Issues in Automatic Learning Object Indexation. In P. Barker & S. Rebelsky (Eds.), Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2002, Chesapeake, VA: AACE, 2002. pp. 239-240.

[10] Austin, Daniel, Peruvemba, Subramanian, McCarron, Shane, Ishikawa, Masayasu and Birbeck, Mark. XHTML™ Modularization 1.1, W3C Working Draft, 2006. Retrieved April 30th, 2008, from http://www.w3.org/TR/xhtml-modularization/xhtml-modularization.html

[11] Krause, J. and Marx, J. . Vocabulary Switching and Automatic Metadata Extraction or How to Get Useful Information from a Digital Library, Proceedings of the First DELOS Network of Excellence Workshop on "Information Seeking, Searching and Querying in Digital Libraries". Zurich, Switzerland, 2000.

[12] Tonkin, E. and Muller, H. L. Semi Automated Metadata Extraction for Preprints Archives. Proceedings of the Eighth ACM/IEEE Joint Conference on Digital Libraries, ACM Press, New York. 2008

[13] Lovegrove, W. S. and Brailsford, D. F. Document analysis of PDF files: methods, results and implications. Electronic publishing, Vol. 8(2&3), 207-220 (June and September 1995).

[14] Giuffrida, G., Shek, E.C. and Yang, J. Knowledge-based metadata extraction from PostScript files. DL '00: Proceedings of the fifth ACM conference on digital libraries. pp. 77-84, ACM, NY, USA, 2000. DOI: http://doi.acm.org/10.1145/336597.336639

[15] Liu, Y., Mitra, P., Giles, C.L. and Bai, K. Automatic extraction of table metadata from digital documents. Proceedings of the Sixth ACM/IEEE-CS Joint Conference on Digital Libraries, 2006. pp 339-340.

[16] Giles, C. L. and Councill, I. D.. Who gets acknowledged: Measuring scientific contributions through automatic acknowledgment indexing. PNAS, Vol. 101, no. 51, 2004. pp. 17599-17604

[17] Harnad, S. and Carr, L.. Integrating, navigating and analysing open Eprint archives through open citation linking (the OpCit project). Current Science. 79(5), 2000. 629-638

[18] Hoche, S. and Flach, P. Predicting Topics of Scientific Papers from Co-Authorship Graphs: a Case Study. Proceedings of the 2006 UK Workshop on Computational Intelligence (UKCI2006), pp. 215–222. September 2006

[19] Olson, H & Wolfram, D. Indexing Consistency and its Implications for Information Architecture: A Pilot Study. IA Summit 2006.

[20] Tavosanis, M. A causal classification of orthography errors in web texts. Proceedings of AND 2007.

[21] van Rijsbergen, C. J. The Geometry of Information Retrieval. Cambridge University Press. 2004

[22] Widddows D. Geometry and Meaning. (CSLI-LN) Center for the Study of Language and Information. 2004

[23] Cardinaels, Kris, Duval, Erik and Olivié, Henk J., A Formal Model of Learning Object Metadata. EC-TEL 2006. pp. 74-87

[24] Han, Hui, Lee, Giles, Zha, Hongyuan, Li, Cheng, and Tsioutsiouliklis, Kostas. Two supervised learning approaches for name disambiguation in author citations. Proceedings of the Fourth ACM/IEEE Joint Conference on Digital Libraries, ACM Press, New York, 2004. pp. 296-30

[25] Han, Hui, Zha, Hongyuan, Giles, C. Lee. Name disambiguation in author citations using a K-way spectral clustering method. In Proceedings of JCDL'2005, ACM Press, New York, 2005. pp.334~343

[26] Powell, A and Johnston, P. Guidelines for implementing Dublin Core in XML. DCMI Recommendation. April 2003. http://dublincore.org/documents/dc-xml-guidelines/

[27] Daneman, Meredyth and Stainton, Murray. The generation effect in reading and proofreading. Reading and Writing, Vol. 5, no. 3, 1993. pp. 297-313. DOI - 10.1007/BF01027393

[28] Berry, Michael W. and Murray Browne. Understanding search engines; mathematical modeling and text retrieval. SIAM, 2005