AUTOMATIC STRUCTURE AND KEYPHRASE ANALYSIS OF SCIENTIFIC PUBLICATIONS

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN THE FACULTY OF ENGINEERING AND PHYSICAL SCIENCES

2014

By
Alexandru Constantin
School of Computer Science

Contents

Word Count: 43870

List of Tables

List of Figures

Abstract

Purpose. This work addresses an escalating problem within the realm of scientific publishing, that stems from accelerated publication rates of article formats difficult to process automatically. The amount of manual labour required to organise a comprehensive corpus of relevant literature has long been impractical. This has, in effect, reduced research efficiency and delayed scientific advancement. Two complementary approaches meant to alleviate this problem are detailed and improved upon beyond the current state-of-the-art, namely logical structure recovery of articles and keyphrase extraction.

Methodology. The first approach targets the issue of *flat-format publishing*. It performs a structural analysis of the camera-ready PDF article and recognises its fine-grained organisation over logical units. The second approach is the application of a keyphrase extraction algorithm that relies on rhetorical information from the recovered structure to better contour an article's true points of focus. A recount of the scientific article's function, content and structure is provided, along with insights into how different logical components such as section headings or the bibliography can be automatically identified and utilised for higher-quality keyphrase extraction.

Findings. Structure recovery can be carried out independently of an article's formatting specifics, by exploiting conventional dependencies between logical components. In addition, access to an article's logical structure is beneficial across term extraction approaches, reducing input noise and facilitating the emphasis of regions of interest.

Value. The first part of this work details a novel method for recovering the rhetorical structure of scientific articles that is competitive with state-of-the-art machine learning techniques, yet requires no layout-specific tuning or prior training. The second part showcases a keyphrase extraction algorithm that outperforms other solutions in an established benchmark, yet does not rely on collection statistics or external knowledge sources in order to be proficient.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

Acknowledgements

Foremost gratitude is expressed to my supervisors, Prof. Andrei Voronkov and Dr. Steve Pettifer, for all their guidance and support throughout this work. I extend my thanks also to my colleagues of the Advanced Interfaces and Formal Methods Groups, for the many fruitful discussions and unwinding moments of comedy. Special mentions go to Dr. David Thorne for his valuable advice and calculating wit, and to Alec Tunbridge, Fabio Papacchini and Mohammad Khodadadi for their help, friendship and humour.

Appreciation is also shown to the fellow researchers, development teams and academic groups that have made data and procedures available for this dissertation, as well as to those who have supported the adoption of the solutions presented herein, by full or part integration within local workflows: the Utopia Documents team, the ScienceWISE team, the SemEval-2010 Task #5 and 2012 BioHackathon organisers and participants, the Public Knowledge Project team, Prof. Min-Yen Kan and the WING NUS team, Elsevier, CrossRef and the members of Force 11.

Finally, I would like to thank my mother, brother and Cami, for their limitless support, encouragement and understanding.

This research was funded by the EPSRC.

Dedication

To my father, the catalyst of a burning desire to get this far.

Chapter 1

INTRODUCTION

1.1 Motivation

The librarian and information scientist Patrick Wilson eloquently worded a now well-felt issue within scientific publishing, regarding the organisation and retrieval of textual information sources:

> *"How can the valuable be kept from oblivion?*
> *How can a man be sure of finding, in the great mass of writings, good and bad, pedestrian and extraordinary, the writings that would be of value to him?"*
>
> *'Two Kinds of Power: An Essay of Bibliographical Control'*
> Patrick Wilson (1968)

Members of academia as well as industry share insights on state-of-the-art research on a regular basis. Their primary medium of communication in this respect is the scholarly article, published in specialised journals or in proceedings of scientific conferences. In order to be published, submitted articles usually undergo the process of peer-review, in which the information they convey is examined by domain experts and weighed for the substantiality of its contribution to knowledge.

Historically, keeping up with the forefront of developments in a certain field meant periodically inspecting new issues of the few existing research journals. Nowadays, with electronic publishing having established itself as the prime medium of knowledge dissemination, publication rates across the scientific spectrum have skyrocketed, with

fields such as biomedicine effectively producing a new publication *every 20 seconds*[1]. Scientists are finding it increasingly hard to keep up with this substantial upsurge and to navigate growing digital libraries efficiently (Attwood et al., 2009; Czoski-Murray et al., 2012a,b).

In response to mounting demands for better solutions, there are now numerous research initiatives that target precisely the automatic processing of scientific articles. The predominant goal of these initiatives has been to reduce the search space for potentially relevant information, through means such as intuitive indexing and retrieval (Lourenço et al., 2010; Dinh et al., 2012), article recommendation (Beel et al., 2013), document summarisation (Teufel and Moens, 2002; Wan et al., 2010) or discourse annotation (Louis and Nenkova, 2011; Teufel and Kan, 2011). These services hold great value because they can offer clear overviews on the current focus of research efforts, as well as on future developments that can be expected. Still, their widespread adoption in scientific information systems has been very modest, mainly due to two recurrent shortcomings:

1. *Limited applicability due to input format restrictions*. This has represented a steadily escalating problem over the years. Most article processing tools require that the input be passed in a structured format that is machine-readable. The most common format for articles however, the camera-ready publication, is instead almost exclusively meant for printing and linear reading by humans, prioritising visual appearance over programmatic accessibility. The prime example format here is the PDF (Portable Document Format), often the sole article format released by publishers.

2. *Limited ability to approximate human relevance judgements*. Although information extraction is a highly-active field of research, solutions are generally quite limited in distilling meaningful, relevant content from article texts. Their common denominator is the identification of words or phrases that, taken together, best sum up the central ideas presented in the narratives – henceforth referred to as *keyphrases*. When keyphrases are selected solely from the text of the document being analysed, the procedure is known as *keyphrase extraction*. Keyphrase extraction is a valuable but intricate task, still largely improvable, according to

[1] This figure can be derived by inspecting the fact sheets of the National Library of Medicine (`http://www.nlm.nih.gov/pubs/factsheets/`). Yearly additions to the MEDLINE bibliographic database have been steadily growing to as much as one new citation every 40 seconds, whilst estimations are that MEDLINE currently covers only 40% of all existing biomedical journals.

recent research in the field (Kim and Kan, 2009; Kim et al., 2010). This is especially true for specialised scientific literature, which typically encompasses a wide range of rhetorical elements, with varying degrees of relevance to a specific information need. These may range from abstracts outlining the work, to figures and tables of results, experimental setups, mathematical proofs, etc.

An inspection of early publications reveals that the core structure of the peer-reviewed article has not changed much since the 19th century in terms of how data is presented. Its primary consumer, however, has. Whilst they used to only be catalogued and read by humans, scholarly works are now mostly filtered and processed by machines. This is an understandable shift, given the ever-growing number of available information sources. A first challenge for machines is then to support working with scientific articles as camera-ready, typeset publications, that have been optimised for printing and reading by humans. Only a noticeably small portion of all research output is made available in more machine-friendly, explicitly structured formats such as XML or LaTeX. The implication is that, while more research is being produced, a diminishing percentage of it is actually being discovered, referenced and built upon.

Once the contents of articles are reliably retrieved, the next challenge is to be able to automatically extract and represent their scientific contributions in ways that facilitate knowledge discovery and help the research lifecycle. In the biomedical sector, without adequate automated means of information management, institutions have turned to hiring professional curators to collect, annotate and organise scientific literature relevant to a certain topic, incurring heavy costs in the process. This phenomenon is wonderfully captured in a quote by the renowned bioinformatician Amos Bairoch:

"It is quite depressive to think that we are spending millions in grants for people to perform experiments, produce new knowledge, hide this knowledge in an often badly written text and then spend some more millions trying to second guess what the authors really did and found."

'The future of annotation/biocuration'
Bairoch (2009)

Fortunately, the issue of the classical article no longer serving the community to its full potential has not gone unnoticed. Initiatives such as Force 11 (Bourne et al., 2012) have started a march towards new ways of digital publishing. Some communities would embrace a structured database entry being added instead of a paper being published (Rebholz-Schuhmann et al., 2005). Until such realisations, however, it remains with

the text mining community to come up with smarter and more reliable solutions for literature management.

Research into document structure recovery and keyphrase extraction has meant to alleviate the above challenges, yet robust and easily integratable solutions are exceptionally rare. There are two general problem areas with the state-of-the-art in these fields. First, structure recovery has been addressed primarily with machine learning methods that are too specialised, requiring separate training for the different article layouts encountered. In practice, a literature review exercise will encompass a great variety of formatting styles, making a tailored solution unfeasible to employ. Second, even with a wide array of keyphrase extraction approaches having been proposed, existing implementations have made little progress towards extracting high-quality terms from article narratives (Kim et al., 2010). More recent studies address this shortcoming by relying increasingly more on external knowledge sources such as thesauri or ontologies, to limit extracted terms to those known to be valid, domain-specific concepts. The adoption of such solutions, however, has been quite modest. This may be due to the added complexity of the implementations, or the caveats that come with third-party dependencies, such as the inability to extract concepts that do not occur in these data stores.

1.2 Proposed Solution

This dissertation proposes a versatile, two-step approach for analysing the structure and content of scientific articles in PDF form. First, an article is processed with a structure recovery mechanism that identifies its organisation over logical units such as the title, sections, references, etc. The employed method is rule-based and directly-applicable to any conventional article layout, as it exploits only typographical conventions inherent in scientific literature. Afterwards, a novel keyphrase extraction algorithm is applied to the new, richly-structured representation of the article. In the first instance, article structure information helps filter out superfluous or noisy text such as headers, footers, tables and formulae. In addition, this feature also facilitates attributing more weight to rhetorically significant text, such as that from the abstract of the document, its section headings or the entire Bibliography. Because of its structure-aware nature, the procedure can more accurately predict the terms that human readers consider relevant, and has been found empirically to produce high-quality results.

When used together, these two processing steps can provide a highly-valuable metadata layer to the contents of vast PDF article collections, that would otherwise remain opaque and impervious to most text mining workflows.

1.3 Scope and Contributions

The two focus points of this thesis lie within the text mining fields of document and keyphrase analysis. More specifically, the work first deals with logical structure recovery from born-digital PDF scientific articles, as opposed to scans of paper documents. Next, it proceeds to keyphrase extraction – the selection of a small set of phrases from the text of a document, normally purposed to best characterise its content. This latter task is set aside from term assignment from sources other than the article text, named entity recognition (the classification of terms into predefined categories) and exhaustive terminology extraction.

The work also links at times to the neighbouring field of information science, for hermeneutic studies and arguments on the nature of textual information sources, their function, representation and organisation. Information science research provides a better understanding of the human perception of content and relevance, supporting its translation into machine code for structure recognition and keyphrase extraction.

The contributions of this research can be summarised as follows:

Theoretical considerations and analyses

- An examination of the nature of keyphrases and a remark of the interdependency between a keyphrase's intended purpose and its inherent quality as a keyphrase.
- An analysis of the impact of logical document structure on the human perception of content relevance.
- An analysis of the feasibility of using automatic part-of-speech taggers in the identification of keyphrase candidates.

Algorithms

- A fine-grained logical structure recovery algorithm, readily applicable to any conventional article layout and competitive with state-of-the-art machine learning techniques.

- A keyphrase extraction algorithm that achieves first place results in an established benchmark, amongst eighteen other systems and three baseline implementations.

Resources

- An implementation of the structure recovery algorithm as a freely-available PDF conversion service.
- A new dataset of open access biomedical articles, coming from nearly 2000 different journals, both in PDF and gold-standard XML form, made available for further research on structure recognition.
- An implementation of the keyphrase extraction algorithm, integrated within an open collaborative article management platform.
- An improvement of the keyphrase extraction benchmark, reviewed and accepted by its original authors, available as an attachment to the PDF version of this dissertation.

1.4 Thesis Outline

This dissertation is structured over six chapters that categorise its contributions.

Chapter 2: Keyphrases and Scholarly Works covers theoretical aspects of these two elements. It aims to answer questions about the definition and purpose of keyphrases and present a formalised set of features that discriminate them from other terms within a document. The scientific article is also examined as an information source with ample rhetoric hidden within its structure, that human readers readily respond to when making relevance judgements.

Chapter 3: Structure Analysis of Scientific Articles details the hard problem of logical structure recovery given only a typographical layout. The proposed solution is presented and evaluated against the state-of-the-art as well as additional collections, diverse in layout and fields of study.

Chapter 4: Keyphrase Extraction scrutinises the progress made in 30 years of research on term weighting approaches, recounting the features that have worked best to help automated tools discriminate keyphrases from non-keyphrases. It then proceeds

to propose and evaluate a new extraction method that draws upon the gained insights to extract salient terms from articles in a highly proficient manner.

Chapter 5: Applications and Availability highlights the practicality of the proposed approaches, by presenting the real-world applications that have already adopted them, in addition also listing many highly-desired information services that can be realised with this functionality. The same chapter also details the availability of all the resources involved in this work, in terms of software, datasets, evaluation scripts and LaTeX sources of tabular data and formulae.

Chapter 6: Conclusions includes closing remarks on the knowledge gained from this research, provides a critical assessment of the methodology employed, and also outlines directions for future work.

Appendix A contains a verbose log and example output for the proposed structure recovery solution, to showcase the steps and decisions made by the system.

Appendix B provides analogous information (processing log and example output) for the keyphrase extractor presented in Chapter 4, along with expert relevance judgements on its output keyphrases in comparison to those of two other systems.

Appendix C aggregates the multiple keyphrase experiment results obtained in this thesis with those of existing efforts over the same data, for a quick overview of the achieved performances.

1.5 Terminology

As the terminology used in this thesis's fields of study is not always consistent across the literature, below is a list of the one adopted herein. The given definitions are considered sufficiently descriptive for the purposes of this research.

- **Word** or **Token**. A single character or group of consecutive characters – letters, numbers or symbols, not including whitespace.
- **Phrase** or **Term**. A single word or group of consecutive words that represents a conceptual unit, i.e. that has a known meaning.
- **Noun phrase**. A phrase formed of a noun plus any accompanying modifiers, such as adjectives or other nouns, that distinguish it by restricting or adding to its meaning.

- **Keyphrase**. A phrase that facilitates a particular operation, most commonly the description of textual content (accepted definitions are discussed in Chapter 2).
- **Keyword**. A single-word keyphrase.
- **Block**. A contiguous rectangular area of a typeset document, containing text or graphics.
- **Region**. A single block or group of blocks that forms a logical unit of discourse, possibly spanning multiple columns or pages.

Chapter 2

KEYPHRASES AND SCHOLARLY WORKS

In pursuit of better ways to manage the increasing amount of available literature, careful examinations of the structure and content of scholarly works have been conducted from many angles. Research in the areas of knowledge management, information science, library and archival studies, as well as text and data mining has been dedicated to analysing the different elements of scientific texts. A common objective has been to identify textual units of particular importance to potential readers, that could be exploited in information services such as search engines, article recommenders or summarisers. An element of particular interest in this respect has been the *keyphrase*. In this chapter, the concept of 'keyphrase' is examined for its perceived definition and purpose, in an attempt to formalise a set of features that best discriminate it from other terms within a document. The scientific article is also brought into context as an information source, considering functional aspects of its structure that facilitate the identification of keyphrases.

2.1 Definition and Purpose of Keyphrases

The term *keyphrase*, commonly, albeit incorrectly, also referred to as *keyword*, seems to be a well-understood concept nowadays, when web search engines mediate virtually all website access and all of them ask that the user input "a few descriptive words" to carry out the search. The usage of keyphrases in daily activities thus seems to be a common

occurrence, yet a concrete definition for the concept, that would allow its translation into machine code, has proven to be quite difficult to formalise. This shortcoming in the understanding of what a keyphrase is, may be the reason why the state-of-the-art in keyphrase extraction is still considered to have much room for improvement, as will be detailed in Chapter 4. Analysing some of the better-adopted definitions for 'keyphrase' may offer insight as to why this concept still seems to elude algorithmic methods of identifying it:

- *Keyphrases provide semantic metadata that summarize and characterize documents.* (Witten et al., 1999)
- *We define a keyphrase list as a short list of phrases (typically five to fifteen noun phrases) that capture the main topics discussed in a given document.* (Turney, 1999)
- *Keyphrases give a high-level description of a document's contents that is intended to make it easy for prospective readers to decide whether or not it is relevant for them.* (Frank et al., 1999)
- *We will call a small set of terms selected to capture the content of a document 'keywords'. 'Index terms' is an alternative term we also use; the choice mostly depending on what the set of words is used for: describing the document or facilitating its retrieval.* (Hulth et al., 2001)

The above definitions come from scientific literature dealing with keyphrase analysis. They vary in specificity, but invariably mention the application for which a set of keyphrases is to be used, i.e. summarisation, topic description, relevance assessment or indexing. Different use cases thus seem to influence the actual definition given for the general concept of 'keyphrase'. Complementary to the above, the following are some of the top hits retrieved by Google for definitions of the term 'keyword' (chosen over 'keyphrase' because of its more widespread use in the non-academic sector):

- *A significant or memorable word or term in the title, abstract, or text of an item being indexed, used as an index entry.* [1]
- *A significant word from a title or document used especially as an index to content.* [2]

[1] Dictionary.com – `http://dictionary.reference.com/browse/keyword`
[2] Merriam-Webster – `http://www.merriam-webster.com/dictionary/keyword`

- *A word, expression, or concept of particular importance or significance.* [3]
- *[...] terms that will help someone locate your chapter at the top of the search engine list using, for example, Google.*[4]
- *Any word that occurs in a text more often than normal.* [5]

These seemingly more popular definitions are quite inconclusive, linking to where a term occurs within a document, to its frequency of occurrence, or to a generic notion of relevance.

Overall, there is noticeable variation in the perception of what keyphrases are and what they can be used for. Consequently, there are very few concrete indicators of how they can be distinguished from other terms within a document. Problems arise because keyphrases are not limited to a specific type – they could represent the general topics being discussed or some very specific concepts. Moreover, the judgement of a term as a keyphrase is an inherently subjective measurement when considering the needs of individual users. Depending on a user's intended purpose for a set of keyphrases of a document, that set may actually vary or multiple different sets may serve the same purpose equally well. This fact is a recurrently overlooked problem in keyphrase analysis. Approximately 75% of this dissertation's literature review on keyphrase extraction does not seem to have a definite purpose in mind for the output terms. In common usage, keyphrases are not the end-goal themselves, but rather the means to an end. Their primary function seems to be that of *waypoints* to information sources, rather than the actual information that users seek. Both people and systems use keyphrases to better navigate the information space. Since some phrases might be better suited to the task than others, an important aspect in their automatic extraction is the measure of quality to attribute to a list of supposed keyphrases.

2.2 Keyphrase Quality Assessment

2.2.1 Comparison against Gold-Standards

The common procedure for evaluating keyphrase quality has been to compare the overlap between an automatically extracted set and a set manually extracted by humans,

[3] Oxford English Dictionary – `http://www.oed.com/view/Entry/312961`
[4] Springer Manuscript Guidelines – `http://www.springer.com/`
[5] Wiktionary – `http://en.wiktionary.org/wiki/keyword`

with the latter being considered the perfect or *gold-standard* output. For scientific articles, gold-standards have been customarily constructed by asking the original authors to provide a set of keyphrases for their articles, and supplementing these with a similar input from readers, in order to avoid bias. The advantage of the procedure is that it can be applied to virtually any document set and that, in most cases, the choices of humans are a desirable standard for keyphrase quality. When gold-standard outputs are available, performance is usually measured using van Rijsbergen's well-known F measure (van Rijsbergen, 1979; Lewis, 1995):

$$F_\beta = \frac{(\beta^2 + 1) * P * R}{\beta^2 * P + R}$$

where P denotes Precision – the fraction of extracted terms that match the gold-standard, R denotes Recall – the fraction of gold-standard terms that were extracted, and $\beta = [0, \infty)$. F_0 thus equates to Precision, F_∞ to Recall and the case when $\beta = 1$ represents the most common usage of the equation in the field of information retrieval, the F_1 measure, when equal weight is given to both Precision and Recall.

The F measure functions well to assess how proficient a system is in outputting a desired set of results. However, in keyphrase extraction, it is not uncommon for partial keyphrases matches to still be considered adequate representative terms for a document. The F measure is unforgiving in this sense, because it only considers exact matches. Another noteworthy aspect is that for many applications such as document indexing, two distinct sets of keyphrases may serve the same purpose equally well. The terms in the two sets do not have to overlap or even be synonymous, but merely be amongst a set of related terms that a human would consider using. An example would be different author names used with different terms from the title of their article in a document search. For example, the keyphrase combinations ("*Witten*", "*KEA*") and ("*Frank*", "*keyphrase extraction*") can hardly be distinguished in their ability to retrieve the publication of Witten et al. (1999): "*KEA: Practical Automatic Keyphrase Extraction*". A workaround to such issues would be to have humans rate the extracted keyphrases directly, but as the procedure involves substantial human effort, it is very limited in the size and scope of the dataset that can be evaluated.

Another disadvantage of the human-derived gold-standard approach concerns quality assessment against the choices of humans in general. Human choices for keyphrases are often inconsistent, as many works have noted (Witten et al., 1999; Barriere and Jarmasz, 2004; Paukkeri et al., 2008; Medelyan et al., 2008; Zesch and Gurevych, 2009).

The core problem lies with the human perception of relevance, which is intrinsically subjective. The knowledge readers have of certain subjects and the perspectives from which they analyse documents, ultimately lead to different appreciations of what content is important. This fact is evidenced when considering the rate of agreement that humans have amongst themselves with respect to the phrases representative of a document. Several studies place inter-annotator agreement between 26% and 43% (Barker and Cornacchia, 2000; Medelyan, 2009; Kim et al., 2010; Liu et al., 2011). This inconsistency is likely linked to the same inherent inability to formally define a keyphrase mentioned earlier. The shortcoming leaves annotators to their own personal views on what a keyphrase is, unless specific guidelines are given for the annotation task. With respect to indexing and classification of documents, Hjørland suggests that indexer disagreement might be systematic, due to underlying factors such as different theoretical views or paradigms (Hjørland, 2002a,b).

Lastly, the adequacy of human annotation also remains doubtful from a practical perspective, when considering the shear size of existing article repositories. In a real-world scenario, if the extraction goal is indexing for retrieval, an automatic procedure might be preferable because it can compile a lot of information not directly accessible to a human, such as corpus-wide statistics on term usage. A human indexer does not have direct access to the intricate statistics that an algorithm is able to derive. He or she is easily surpassed in computational ability to determine the set of terms with the best discriminating power for an article. What the indexer might still have over machines is better, albeit implicit, knowledge of textual semantics and of human search behaviour. This constitutes an advantage in predicting the most likely terms that users will search for when seeking a certain type of article, as they are hardly ever the optimal ones.

2.2.2 Proposed Alternatives for Quality Assessment

In an attempt to overcome the exact match limitation of the F measure and similar metrics, Barriere and Jarmasz (2004) relied on a terabyte-sized corpus to derive a measure of semantic similarity between automatically extracted terms and those chosen by authors. Pointwise Mutual Information (PMI) (Church and Hanks, 1990) was used to compute this similarity, drawing upon co-occurrence statistics of pairs of keyphrases over the corpus.

Paukkeri et al. (2008) developed a language-independent approach to keyphrase extraction and needed to evaluate extracted keyphrases for 11 languages. In order to avoid problems related to human evaluation in this case, the authors used Wikipedia articles as the evaluation dataset, and the anchor texts within them as potential keyphrases. Titles of articles linked from each article in the dataset, that in return also linked back to the article in question, were chosen as the gold-standard keyphrases.

In the work of Medelyan (2009), the main evaluation procedure was a comparison of the consistency of indexing between a human and a machine, with that of two humans. If an automatically extracted set of keyphrases had the same statistical overlap with a set extracted by a human, as two humans did with each other, the algorithm was considered human-competitive. The approach functioned well as a more tolerant appreciation of the quality of automatically extracted terms, but it can also be problematic, as remarked early on by Soergel (1994): the indexing of two approaches might be consistent, but consistently incorrect, therefore evaluating just this factor alone can lead to incorrect conclusions. Additional supporting evidence, coming from the analysis of indexing correctness in some way, should also be considered for drawing more meaningful conclusions. Medelyan used the F_1 measure to address this matter.

2.2.3 The Real-World Performance of Keyphrases

Regarding one of the first applications for keyphrases, document indexing, Anderson and Pérez-Carballo (2001) remarked that system evaluations were conducted on relatively small document collections, and that performance was based on judgements of persons other than real users with real information needs. Research into the true merits of automatic vs. human indexing had remained largely inconclusive. Cooper (1978) found that such an investigation was hampered to various degrees by "*almost insurmountable methodological obstacles*" involved in judging the overall quality of systems in their entirety. Repeated full-scale system evaluations were complex and time consuming, involving many empirical evaluations of different functionalities. This drawback existed regardless of whether humans or machines had carried out the indexing.

Because of this persistent inability to systematically compare the two approaches, the experiences of the actual users of such platforms seem to be a more reliable indicator of the quality of the underlying keyphrases. With a precise, well-defined application

in mind, the performance of the end-result can be easily judged, and can represent the success factor of the processing methods involved. As Hjørland and Nielsen (2001) also remarked, *"what counts as an information source is always relative to the question that it is supposed to answer"*. It can therefore be argued that keyphrases inherently need a purpose in order for their quality to be accurately evaluated and to even be properly defined. Without applications to test their usability in practice, the true quality of keyphrases remains questionable.

2.2.4 Keyphrase Quality Assessment in This Thesis

Given the wide acceptance of the gold-standard evaluation approach, despite its caveats, it is also employed on this thesis's proposed solution for keyphrase extraction, as it fosters an easy comparison with other approaches. However, given the complementary practicality of an evaluation in a real-world scenario, such an experiment was also put together, to foster a more thorough appreciation of the developed extractor's capabilities.

For the purposes of benchmarking the extractor against the state-of-the-art, the evaluation procedure of the SemEval-2010 keyphrase extraction challenge (Kim et al., 2010) was used. Nineteen systems took part in this challenge, which targeted precisely the extraction of keyphrases from scientific articles. The article collection of the competition comprised 100 scientific articles from various computing fields, annotated with author and reader keyphrases. A micro-averaged exact-match F_1 measure was computed at three keyphrase thresholds: the top 5, top 10 and top 15. In addition, a revision and improvement of the competition's evaluation script and gold-standard keyphrase sets was conducted in the course of this research, with the resulting modifications being proposed to the challenge organisers. The proposal was well-received, and with the support of the organisers and participants, the original submissions of 17 of the total 19 systems were retrieved and reprocessed with the updated procedure. One participant also contributed results for an updated (2014) system implementation, for an updated view of the state-of-the-art. Details regarding the competition and proposed modifications to the benchmark are given in Chapter 4, Sections 4.1 and 4.3.

Despite the SemEval benchmark, this chapter has thus far highlighted that human choices for keyphrases are not always dependable. A more reliable way of determining the quality of keyphrases seems to be measuring the success of real-word applications

that use them. This claim has received support from both the text mining and information science communities. A setting to judge the developed extractor's proficiency in practice was thus also put together within the ScienceWISE platform[6]. The evaluation was carried out over two tasks: article bookmarking for management of personal collections, and the enrichment of an ontology of scientific concepts. Details about this experiment are given in Section 4.3.4.

2.3 Rhetorical Roles of Article Components

In a study of the changes in scholarly article searching and reading patterns, Tenopir et al. (2009) remarked that, on average, the number of articles read by a person in academia was increasing, while the average time spent per reading was declining. The process was thus shifting towards a selective reading of just a few regions of interest, in order to determine an article's relevance quickly. The factors facilitating this content skimming were well-understood structural and stylistic cues that helped steer readers' attention, such as conventional names for certain sections (e.g. 'Abstract', 'References') or font emphasis (size and typeface) (Dillon, 1991; Zhang, 2012). For the purposes of keyphrase extraction, these rhetorical indicators could also be exploited to better outline the importance of certain terms over others.

As pointed out by Hulth (2004), humans are to a certain extent able to select keyphrases for unfamiliar documents and possibly even to appropriately select unfamiliar terms. It can thus be assumed that properties making some phrases keyphrases are also found on another level than word semantics. It is intuitive to consider that terms occurring in the title or abstract of an article are more likely to be important than those that do not, because of the established rhetorical functions of these elements. The decision on what weights to assign to different sections, however, should depend on the specific scientific domain being analysed, as well as on the desired particularities of the extracted keyphrases. For example, Shah et al. (2003) found the Introduction section to be a good source of gene and protein names within Genetics articles, while for Food Informatics and Computer Science, Hofmann et al. (2009) found the Discussion section most useful in extracting keyphrases that authors were likely to choose.

Within the text mining domain, many keyphrase features have been proposed, as the

[6]The ScienceWISE platform – `http://www.sciencewise.info`

next section will recount. However, only limited consideration has been given to underlying theoretical views on the nature of textual information sources and the way in which humans generally process them. Research conducted in the neighbouring field of information science provides valuable insight into how features such as visual cues fit within mental models for organising information, and are used by readers to assess the function, intent and importance of textual units. In analysing the usage of a digital library system, Bishop et al. (2000) conducted a user study aimed at assessing users' needs and practices with respect to different logical elements of articles. Through interviews, the authors examined how users utilised these elements in literature searches and in the decisions to actually retrieve and read certain documents. The find was that different components such as abstracts, references, captions and author affiliations were used for a multitude of tasks: from summarisation and predicting the impact of a document, to determining its authoritativeness and deciding what parts to skip when reading.

Zhang (2012) also conducted an analysis of the functions of article logical components. The author recommended the organisation and presentation of journal articles over *functional units* to enhance reading efficiency and the effectiveness of the reading outcome. A functional unit was defined as a chunk of information with a distinct communicative purpose, that subdivided the four major components of an article: the introduction, methods, results and discussion. For example, recent developments in a Related Work section could comprise one functional unit, while the indication of a gap in previous research could comprise another. Through experiments with a prototype article management system, the author found that functional units could support navigation, in-depth reading, comprehension and information use to various extents. Empirical evidence in Zhang's study supported the claim that different logical components and their subdivisions, functional units, had varying degrees of relevance to certain information needs.

2.4 Keyphrase Features and Their Formal Encoding

Pioneering works on term weighting methods such as those of Salton and Buckley (1988); Witten et al. (1999); Turney (2000); Barker and Cornacchia (2000), have paved the road to what is now a very active research field that targets the automatic extraction of *important*, content-bearing terms from electronic documents. As previously

discussed, what constitutes as *important* varies depending on the targeted application, therefore so too will the characteristics of the terms facilitating that operation. For their more widespread uses however, content description and indexing, some discriminating properties of keyphrases have emerged. The features considered for identifying them automatically range from simple frequency of occurrence and part-of-speech (POS) tags, to semantic relatedness and centrality within linguistic dependency graphs (Csomai and Mihalcea, 2008; Joorabchi and Mahdi, 2013).

This section summarises the most common features that have been used in keyphrase extraction systems. Both observable and non-observable features are covered, such as the number of tokens a keyphrase has, versus its total frequency within a document collection. A feature's type and scope are taken as a basis for categorisation. The type generally falls under one of four categories: *Statistical, Linguistic, Structural* and *Semantic*. The scope denotes the level at which a feature is derived. For example, information about the term's length is available at the phrase level, whereas term frequency can be computed from the document itself, or over an entire collection. The four scopes used in the following categorisation are *Phrase, Document, Collection* and *External*. Table 2.1 presents the considered features over these categories.

2.4.1 Feature Types

Statistical Features

Possibly the first statistical remark made about keyphrases was recurrence, i.e. that the above-norm repetition of a certain term inevitably conveyed it greater significance over other terms. From the frequency within a document (TF) came the idea of scarcity within other documents (IDF) as a complementary measure of the significance of a term (Spärck Jones, 1972). The TF-IDF combination (Salton, 1975; Salton et al., 1975) was soon to follow, and established itself as an efficient way of identifying possible indexing terms for documents.

Other observations regarding term statistical emphasis were that important ones had a tendency to be introduced early on in the narrative (hence the *first occurrence* feature), to be repeated throughout the text (*last occurrence*) and their constituent words would display a better co-occurrence stability than other words (*phraseness*).

Table 2.1: Common features used in automatic keyphrase extraction, categorised by their type (Statistical, Linguistic, Structural, Semantic) and usual scope (Phrase, Document, Collection, External).

Feature	**Scope**			
	Phr.	**Doc.**	**Col.**	**Ext.**
Statistical				
Length (in number of tokens)	X			
Term Frequency (TF)		X	X	X
Inverse Document Frequency (IDF)			X	X
Cluster Term Frequency (CTF)		X	X	
Phraseness (collocation statistics, e.g. GDC [a], PMI [b])		X	X	
First occurrence (within the document or a paragraph)		X		
Last occurrence		X		
Shorter term subsumption		X		
Longer term promotion		X		
Informativeness (e.g. TF-IDF[c])			X	
Linguistic				
IsStopword	X			
HasPunctuation	X			
Suffix sequence	X			
Part-of-speech (POS) tags (e.g. IsNounPhrase)	X	X		
Acronymity	X	X		
Structural				
Font emphasis (size, typeface)	X	X		
InTitle		X	X	X
InSection[d]		X	X	
InHeading		X	X	
TF in Title / Section / Heading		X	X	X
SF-ISF (i.e. TF-IDF over individual sections)		X	X	
Semantic				
Keyphraseness (known to be a topic/keyphrase)			X	X
Relatedness / keyphrase cohesion			X	X
Synonymy (e.g. Wikipedia redirect links)				X
Occurrence in a controlled vocabulary / thesaurus				X
Term variant conflation				X

[a] Generalised Dice Coefficient (Park et al., 2002)
[b] Pointwise Mutual Information (Church and Hanks, 1990)
[c] The TF-IDF measure (Salton, 1975; Salton et al., 1975)
[d] Any one of Abstract, Introduction, Related Work, Discussion, Conclusion, or Bibliography.

Linguistic Features

Another set of features soon followed statistical ones, once language-specific extraction solutions started being developed. Among the first linguistic considerations was the use of POS tags to restrict the set of keyphrase candidates to e.g. terms containing only nouns and adjectives (Barker and Cornacchia, 2000; Hulth, 2004). The exclusion of exceptional terms that did not bear any useful information (*stopwords*) was also employed. Abbreviations likewise emerged as a useful way of identifying salient terms, as significant concepts would often be associated with a shorter phrase, for easy referencing throughout the text (Nenadić et al., 2002; Nguyen and Kan, 2007; Pianta and Tonelli, 2010).

Structural Features

As the breadth of features used for keyphrase extraction started to grow, a recurrent remark concerned the rhetoric of the texts being analysed: the way in which information was presented within them, both stylistically and logically, carried implied term importance as well.

Programmatic access to an article's structure in terms of logical regions was found to improve extraction performance irrespective of the other considered keyphrase features (Shah et al., 2003; Hofmann et al., 2009; Nguyen and Luong, 2010). The recent uptake in the availability of publications in machine-friendly formats such as XML has expanded possible feature use to also include term occurrence in section headings or in titles of other articles as indicators of high-value phrases. This has led to a number of information extraction solutions coupling themselves to data stores that have such semi-structured versions of publications. The amount of information available in these stores varies from simple metadata such as the title, authors and publisher information (e.g. DBLP[7]), to author-provided keyphrases (e.g. HAL (Baruch, 2007)), to abstracts and citations (e.g. Scopus (Burnham, 2006)), to rich annotations of the full-text, down to paragraph and even equation-level (e.g. PMC[8], arXiv[9]). The main limitation in the adoption of structural features alongside statistical and linguistic ones is the scope and availability of corpora annotated with such information.

[7]The DBLP Computer Science Bibliography – `http://dblp.uni-trier.de/`
[8]The PubMed Central Archive – `http://www.ncbi.nlm.nih.gov/pmc/`
[9]The arXiv.org e-Print repository – `http://arXiv.org/`

Semantic Features

This last feature category follows the development the article stores just mentioned, as well as of thesauri and ontologies that curate terminologies used in certain fields. Knowledge of valid scientific concepts relevant to an article's domain of study is very useful in limiting the set of all candidate keyphrases to those known to be meaningful. Medelyan (2009) and Lopez and Romary (2010) for example, use the Wikipedia Miner tool (Milne and Witten, 2013) to derive the probability of a term being an anchor across Wikipedia articles, referred to as *keyphraseness*. Likewise, a *relatedness* feature has been proposed on the assumption that keyphrases are often associated with each other, being semantically related to the topic of the document (Kim and Kan, 2009; Zhang et al., 2013). This feature is also referred to as *keyphrase cohesion*.

The main drawback of this feature category is, as with the structural one, the availability of data stores that can provide the required information. Depending on how semantic features are used in an implementation, another shortcoming might also be the inability to extract novel, emergent concepts that have yet to be catalogued by these external sources.

2.4.2 Feature Scopes

A feature's scope is important in acknowledging the amount of information required to derive it. Narrow-scoped features are generally easier to compute and have consequently been more often integrated in keyphrase extraction solutions. The different scopes of features generally follow their types:

- Linguistic features are mostly phrase-scoped, as they examine the terms' morphology. Others extend to the document scope and use contextual information, such as the POS tags of neighbouring words, or the prior introduction of an abbreviation.
- All structural and most statistical features fall within the document scope. Extractors that do not go beyond this scope do not have any external dependencies and can therefore be readily applied to any document. The keyphrase extractor of this dissertation is document-scoped. Its evaluation in Chapter 4 will therefore highlight the achievable performance over existing methods, with only a careful re-engineering of document-intrinsic features, such as an article's structural rhetoric.

- All document features are also extendable to the collection scope for garnering more contextual information, although some, such as the first occurrence or font emphasis are rarely considered beyond the input document itself. The prime example of a feature bound to the collection scope is the IDF measure.
- Semantic features rely on prior knowledge of the meaning and function of certain terms. Consequently, they are at the very least collection-scoped. Some directly consult external sources of information compiled in advance, either from collective human input (e.g. Wikipedia) or from various article corpora of reference.

2.5 Summary

Although keyphrases are pivotal in highly-accessed services such as search engines, the formal description of their features seems to have been hampered by a limited understanding of the way in which humans assess term relevance. A keyphrase can be deemed good when it helps fulfil an information need with minimal effort, but without this information need, the keyphrase is merely a phrase. For this reason, *quality considerations for keyphrases should always be dependent on their intended purpose*.

With respect to descriptiveness, a set of keyphrases can be judged as high-quality when the impression it leaves regarding the article's contents is well in tune with the impression a reader is left with after inspecting the article. This judgement is best derived in an empirical setting with real users and real information needs. However, for the purposes of benchmarking different extraction solutions, gold-standard sets of manually-selected terms are still in use, as the evaluation procedure is easy to set up. Regarding discriminating qualities for retrieval, keyphrases are meant to function as waypoints in a large information space. Getting a 100% match against a list of human-extracted terms might not matter in this case, as long as the documents 'pointed at', i.e. the information sources, are roughly the same and can be retrieved with comparable ease.

This chapter has provided a theoretical appreciation of the function of keyphrases, of the specialised structure of scientific articles, and of how humans use these to navigate content. The implied rhetoric within the different elements of a publication stands out as one of the most helpful features for identifying keyphrases, yet it has been only modestly adopted in extraction software. This is due to the limited overall availability of such metadata, and of adequate means of extracting it from article corpora. The

following chapter will detail a novel method for recovering an article's rhetorical structure that is layout-independent, yet competitive with state-of-the-art machine learning techniques. Afterwards, Chapter 4 will proceed to exemplify how the recovered structure can be integrated within a keyphrase extraction solution to achieve top-performing results.

Chapter 3

STRUCTURE ANALYSIS OF SCIENTIFIC ARTICLES

An important precursor of high-quality information extraction is the ability to provide the algorithm with high-quality input. With regards to camera-ready scientific publications, the content is carefully positioned when displayed, but noisy, unordered and unstructured when extracted to be processed. As highlighted in the previous chapter, the structure of an article is very influential in a human reader's assessment of relevant content. In addition to coherent text extraction, structure recovery is therefore also highly beneficial.

Being a specific genre of articles, scientific papers can be said to comprise *"a distinctive type of communicative action, characterized by a socially recognized communicative purpose and common aspects of form'* (Yates and Orlikowski, 1992). On this premise, that meaning and visual appearance are intrinsically linked, it follows that information about one could be used to inform about the other. It is indeed true that, when looking at an article, a human can likely discern the logical roles of different elements with minimal effort, even if the article is written in an unfamiliar language.

Dillon (2000) writes that "*as well as identifying placement and layout, users directly recognize and respond to content and meaning*". This chapter tests the capability of an automated method to achieve the same – use an article's geometrical layout to identify logical units of discourse and reconstruct the narrative flow. Interdependencies such as those between a figure caption, the figure it describes, and the in-text references that point to that figure are carefully considered, in an attempt to approximate some of the rule sets that humans use to distinguish certain elements.

3.1 Background and Related Work

3.1.1 A Growing Format Issue

The accelerated increase in volume of published scientific research has given rise to numerous document processing initiatives aimed at reducing the search space for potentially relevant information. Efforts in this respect have been successful in carrying out tasks such as intuitive indexing and retrieval (Lourenço et al., 2010; Dinh et al., 2012), document summarisation (Teufel and Moens, 2002; Wan et al., 2010) or discourse annotation (Louis and Nenkova, 2011; Teufel and Kan, 2011). The added-value brought by such services can be quite significant when considering the very high publication rates in certain domains, for instance biomedicine and the life sciences.

A problem exists, however, in that many document processing tools work exclusively on structured machine-readable content rather than camera-ready, typeset publications. This makes their performance dependent on data sources containing noise-free, accurate representations of article narratives. Though some publishers and digital libraries now make content available in machine-readable form, many do not, and much legacy content exists only as PDF documents designed primarily for human reading and not programmatic access.

The state-of-the-art in article structure analysis employs machine learning or template matching approaches (Hollingsworth et al., 2005; Luong et al., 2011) only able to process a limited set of same-style articles at a time, and that require repeated human intervention in a conversion process. Because of these drawbacks, text analysis tools often choose to couple themselves to data stores that have semi-structured representations of articles available, such as PubMed Central (PMC)[1], DBLP or arXiv[2]. The approach has proven successful, but only to a certain extent, as this compromising dependency deters the tools' widespread adoption. Much of the information sought by researchers is made available solely within PDF publications with no alternative representation. Without means of readily expanding their reach to this highly popular format, many promising natural language processing and text mining solutions remain either undiscovered, or of limited use to potential users.

The PDF is still the de-facto standard for distributing published scientific work and

[1]The PubMed Central Archive – `http://www.ncbi.nlm.nih.gov/pmc/`.
[2]The arXiv preprint database – `http://arxiv.org/`.

continues to present many challenges to attempts of converting it into computationally amenable formats. Too often, formatting embellishments such as headers and footers or columnated layout hamper the correct extraction of content, providing noisy or erroneous input to subsequent processing stages, and invariably degrading the quality of the end-result. The PDF's persistent popularity and the vast, diverse catalogues of legacy material available only in this format ask for versatile conversion solutions if the knowledge contained within them is to be harnessed.

The primary technical challenge of accurate content retrieval from a PDF article is to determine its rhetorical structure given only a typographical layout. The distinction between a title, section headings, tabular data, etc. can, for the most part, be easily understood by a human reader from a page's physical layout and the use of fonts. Interestingly, this is often possible even if the document is written in an unfamiliar language. This suggests that generally accepted typesetting conventions hold important cues for determining elements' rhetorical functions. For example, the most emphasised element of the front matter, aside from drop capitals and exceptionally large headers, is probably the title. Different publishers abide to such conventions to varying degrees and may at times require readers to make use of finer-grained stylistic and linguistic cues in order to logically distinguish units of text[3]. For a machine, this distinction needs to be made explicit, that is, marked-up in some declarative way. The PDF has supported the inclusion of structural metadata since version 1.4 in 2001, but so called 'tagged' PDF documents are exceptionally rare. Without this metadata layer to describe the document's logical structure, PDF text extraction tools see the input simply as an arbitrary stream of symbols. The symbols come with individual style and positioning information, but given the PDF's display-oriented purpose, there is no precise order among them and no indication of their rhetorical function. Different components thus become intertwined when extracted, running headers and page numbers intrude into body text and column boundaries disappear. This makes and already difficult text mining task significantly harder or simply impossible in some cases.

Notable existing efforts that target the recovery of some structure from the flat, machine-unfriendly PDF are expanded upon in the next section. As will be illustrated, except for the SectLabel system (Luong et al., 2011), current tools focus on geometrical analysis, aiming to output groupings of words into lines, blocks or columns, and either do

[3] As an illustration of such cases, Figure 3.1 depicts some examples of less common layouts.

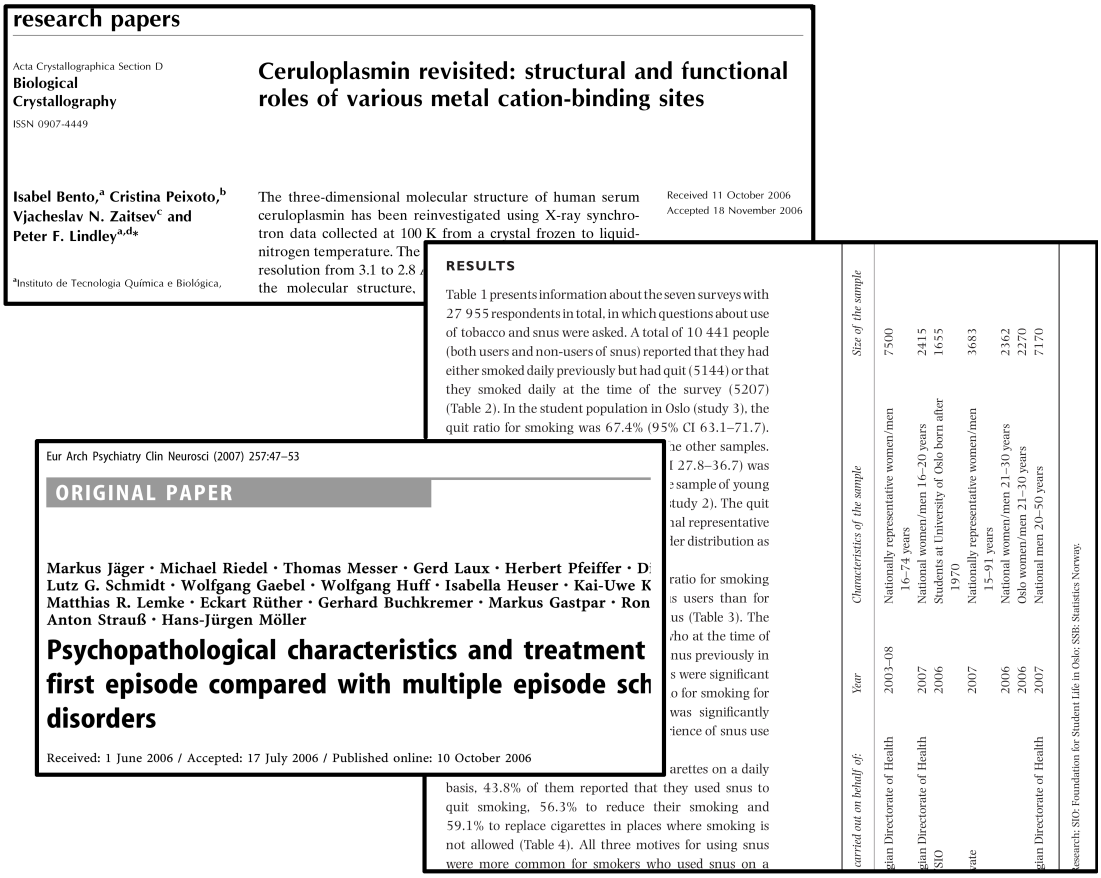


Figure 3.1: Non-standard article layout examples. The two examples on the left show an atypical positioning of the author names (to the side, respectively above the article title). The example on the right depicts a page with horizontal body text on the first column and a vertically-oriented table on the second.

not handle or are in their preliminary phases of logical structure recognition. Complementary, the PDFX system described in this chapter focuses on logical structure, but handles its geometrical baseline as well[4]. It aims to identify, extract and link these two structures together in order to facilitate an enhanced level of interaction with the article's contents. The method employed is rule-based, iterative and unrestricted with respect to the set of formatting templates that input articles need to adhere to. The only requirement is that they be full-text born-digital documents in PDF form, as opposed to PDF images such as scans of paper documents. The 19 logical element types that the described version of the system (v1.9) can differentiate are listed in Table 3.1. These elements cover the principal parts of a typical research article. They are ultimately

[4]The name 'PDFX' should not to be confused with the PDF/X ISO 15930 set of standards meant for graphics exchange via PDFs.

Table 3.1: The 19 logical element types that PDFX v1.9 can differentiate. Labelled formula recognition has been added on top of the capabilities reported in Constantin et al. (2013).

Front Matter	Body Matter	Back Matter / Others
title	body text	bibliographic item
author	(sub)section	URI
abstract	(sub)section heading	email
author footnote	image	side note
	table	header/footer
	caption	page number
	figure/table reference	
	bibliographic reference (in-text citation)	
	labelled formula	

stored in an XML file with a tag hierarchy that closely follows the ANSI/NISO Journal Article Tag Suite standard (JATS)[5]. The semi-structured nature of the XML serves as a convenient, quick access route to any of the articles's components.

To exemplify its utility, the PDFX public web service hosted at `http://pdfx.cs.man.ac.uk/` further transforms the XML into HTML, presenting the core content of the original article as a single-column stream of text, free from elements such as headers, footers and side notes. Also, because the positioning of tables and figures on paper is mostly just aesthetic, these are extracted and repositioned side-by-side with the core text, so as to not obstruct the reading flow. An illustration of this functionality is available at `http://pdfx.cs.man.ac.uk/example`.

3.1.2 Related Work

What follows is an account of the work done so far in recognising the two structures of PDF scientific articles (geometrical and logical) and their alignment to the presented approach.

A clear picture of the research conducted on the subject prior to 2003 is given by Mao et al. (2003). The authors analysed 17 document structure recovery algorithms in detail and derived four general limitations of the surveyed algorithms:

[5]The ANSI/NISO Standard Z39.96-2012 – JATS: Journal Article Tag Suite (`http://jats.niso.org/`) formalised from the U.S. National Library of Medicine (NLM) Archiving and Interchange Tag Suite.

1. *Assuming that physical layout analysis has already been performed*. The argument here was that the physical structure analysis procedures relied upon for logical structure inference were not flawless, and should not have been assumed as such. Mao et al. targeted visual analysis via optical character recognition (OCR) with this remark, but correctly reconstructing words, lines and blocks of text is also an important step of the content retrieval method used in this dissertation: direct access to the PDF object model. The particularities of these two text extraction methods are discussed in Section 3.2.2.

2. *Making use of deterministic models that fail in the presence of noise or ambiguity*. This remark was more specific to the analysis of document images. It aimed to highlight the possibility of uncertainty in the logical structure recognition phase due to erroneous physical layout analysis results or document noise coming from e.g. printing, photocopying, faxing, etc. Rigid deterministic models were unable to cope with such inconsistencies.

3. *Neglecting quantitative performance evaluation issues*. This limitation regarded the lack of a soundly designed experimental methodology. A point was made that above meaningful performance metrics, error analyses and well-defined ground-truth, representative datasets should also be decided upon and reused across different algorithms, to foster comparative evaluations. As will be detailed in Section 3.3.1, the evaluation of this dissertation's proposed method was conducted over four datasets that covered a wider spectrum of layout types and subject domains than previous works. Care was also taken not to skew results in favour of a few, more popular layouts. Three of the four datasets are freely available.

4. *Not basing the work on trained models for specific classes of documents*. The point made here was that document structures vary greatly in complexity and that appropriate techniques should be used for each. This remark targeted the precision of tools in recognising different elements correctly and made a valid point that following a template-matching paradigm was likely to yield better results for the targeted layout. It can be argued, however, that in order to be practical, a conversion solution needs to retain layout independence. In a very common literature review scenario for example, a researcher will most likely not want to limit his or her analysis to just a few formatting styles for which trained models exist. Styles vary across journals and conference proceedings, within the same journal or proceedings over time, and there is no certainty that they

will not continue to change in the future. Unless sustained effort is invested, on the account of end-users, to cater to the specifics of all relevant layouts, systems employing a trained model approach will likely have limited impact in real-world scenarios.

A rule-based automated labelling module was presented in Kim et al. (2001). The underlying algorithm used 120 hand-crafted rules derived from page layout analysis of medical journals and features extracted from OCR output. The authors first categorised a paper into a specific layout template and then used OCR features together with *cue-word* lists to perform element recognition. For example, inclusion of the word 'Diabetes' in a text block in the upper half of the first page was taken to suggest the presence of a journal name. The authors also mentioned that devising a single rule-based algorithm that can handle all journals is unlikely and that individual rules have to cater for each particular layout type. In what follows, this statement is challenged with the presentation of a more versatile rule system that makes use of stylistic and contextual information to overcome this limitation.

The work presented in Hollingsworth et al. (2005) was amongst the first to deal specifically with PDF articles. The issue of general-purpose PDF converters performing poorly on full-text articles was mentioned. It was believed to be due to the lack of knowledge about the article's subject domain and of the publisher's typesetting rules, as these set scientific articles aside from other forms of literature. With this belief, the authors motivated the need to generate well-developed journal templates for recovering text structure, afterwards proceeding in a similar manner to Kim et al. (2001).

A paper by Ramakrishnan et al. (2012) documented an approach to PDF text extraction proposed as a baseline for further experiments into more advanced methods. The methodology considered was also rule-based, but the rule sets were user-defined. The authors made their **LA-PDFText** tool customisable for specific purposes by the use of Drools[6] rule files. These files needed to be created by users for article layouts of interest, and used as input parameters to the system. The rules that had to be defined were for block classification into 5 rhetorical categories (title, author, abstract, section and section heading). In their paper, the authors showcased and linked to examples of Drools files meant for two epochs of the PLOS Biology journal[7].

[6]The Drools business logic integration platform – `http://drools.jboss.org/`
[7]The PLOS Biology journal – `http://www.plosbiology.org/`

The **PDFExtract** system presented in Berg (2011); Berg et al. (2012) is a parameterisable toolkit also aimed at high-quality text and geometrical structure extraction from born-digital PDFs. It takes after related OCR visual analysis techniques for tasks such as whitespace detection and block identification, having adapted them to the PDF processing task. It then proceeds to use heuristics to assign one of several logical roles to blocks. The possible roles are *title, abstract, section, section heading, body text* and *footnote*.

The work documented in Luong et al. (2011) addresses logical structure recovery in scholarly articles with rich document features. The authors extend an existing platform for reference string parsing called **ParsCit** (Councill et al., 2008) that uses the machine learning methodology of conditional random fields for model training. This confers the devised **SectLabel** module the ability to process both metadata-rich XML produced by the Nuance OmniPage OCR engine[8] as well as plain-text input, albeit with reduced accuracy. A significant increase in performance was noted when rich spatial and stylistic cues were considered, such as absolute and relative positions, font faces and special line attributes (e.g. bullet points or tabular data). In terms of the logical structure it extracts, the SectLabel system is the best-aligned related work to this dissertation's proposed solution, but uses machine learning rather than rule sets to achieve its goal. The article collection used in evaluating SectLabel was freely available, providing the opportunity for an interesting comparison. The comparison is detailed in the evaluation part of this chapter (Section 3.3).

Table 3.2 provides a summary of the capabilities of the systems that were found available for use, in terms of the geometrical and logical structure elements that can be extracted. Other well-known, freely available solutions are also included for a more complete overview of existing PDF processing software.

3.2 Proposed Solution: PDFX

The proposed approach addresses structure recovery with the initial generation of a geometrical model, on which a rule-based element identification sequence is employed. Identification stages exploit only typographical conventions inherent in published scientific literature and do not require domain- or layout-specific tuning such as template

[8]Use of the Nuance OmniPage OCR Engine (`http://www.nuance.com/for-business/by-product/omnipage`) is a preprocessing requirement of SectLabel for analysing PDFs.

Table 3.2: Existing tools for structure recovery from PDF articles and their capabilities, alongside the PDFX tool described in this chapter.

Tool	Geometrical Structure	Logical Structure
pdftotext -bbox [a] (Output: XHTML)	pages, words (with coordinates)	-
pdftohtml -xml [a] (Output: XML)	fontspecs, pages, lines (with coordinates, font info), emphasis	-
pdftohtml -c [a] (Output: HTML+CSS)	paragraphs; CSS positioning instructions	not explicit
pdf2xml (1) [b] (Output: XML)	pages, lines, words (with coordinates, font info, rotation, emphasis)	-
pdf2xml (2) [c] (Output: XML)	pages, font blocks (size, face, colour), lines (with coordinates), images	-
pdftohtmlEX [d] (Output: HTML+CSS)	fontspecs, lines, words; CSS positioning instructions	not explicit
pdfextract [e] (Output: XML)	pages, columns, lines, regions (with coordinates, font info, implicit font face)	title, header, footer, body, reference
LA-PDFText (Output: Text/XML)	text blocks (with font, line number, height)	title, author, abstract, section, section heading
PDFExtract (Output: XML)	fontspecs, pages, paragraphs (with coordinates), lines (with font info)	title, abstract, section, section heading, body, footnote
SectLabel (Output: XML/HTML)	provided by third-party tool	title, address, affiliation, author, footnote, category, keyword, copyright, body, (sub)section, (sub)section heading, figure, table, caption, construct, equation, list_item, note, reference, email, page
PDFX (Output: XML/HTML)	logical elements with page and column attributes and block, column and page break markers	title, author, abstract, author footnote, body, (sub)section, (sub)section heading, figure, table, caption, figure/table reference, citation, reference, URI, email, side note, header/footer, page

[a] The Poppler PDF library – `http://poppler.freedesktop.org/`.
[b] The pdf2xml project – (Déjean and Meunier, 2006)
[c] Mobipocket.com pdf2xml – `https://launchpad.net/pdf2xml/`
[d] The pdf2htmlEX project – `http://coolwanglu.github.io/pdf2htmlEX/`
[e] CrossRef Labs pdfextract – `https://github.com/CrossRef/pdfextract/`

matching or model training. The approach is implemented in PDFX, a system designed to output the recovered article structure in an XML format. The resulting XML describes the document's organisation over logical units, and also links it to geometrical typesetting markers in the original PDF, such as column or page breaks. The performance evaluation of the system has been conducted in two settings: a comparison against the SectLabel system mentioned previously, and on articles from PubMed Central, Elsevier[9] and ACM[10] , against gold-standard XML representations.

3.2.1 Design Principles

The implementation of PDFX carries out a two-stage process in order to address the task of structure recovery. The first stage constructs a geometrical model of the article's contents to determine the spatial organisation of textual and graphical units on page. The second stage draws upon the first to identify different logical units of discourse based on their interdependencies and discriminative features.

3.2.2 Geometrical Model Baseline

Text and layout information from PDF documents has, in the past, been obtained through the use of OCR tools (Esposito et al., 1995; Kim et al., 2001), with this still being the case in some more recent approaches (Hollingsworth et al., 2005; Luong et al., 2011). OCR tools attempt to infer text and font information features through a visual analysis of each page as a static image, rather than by accessing the physical structure information of the PDF itself. Considerations for using OCR for this task usually arise from the following aspects:

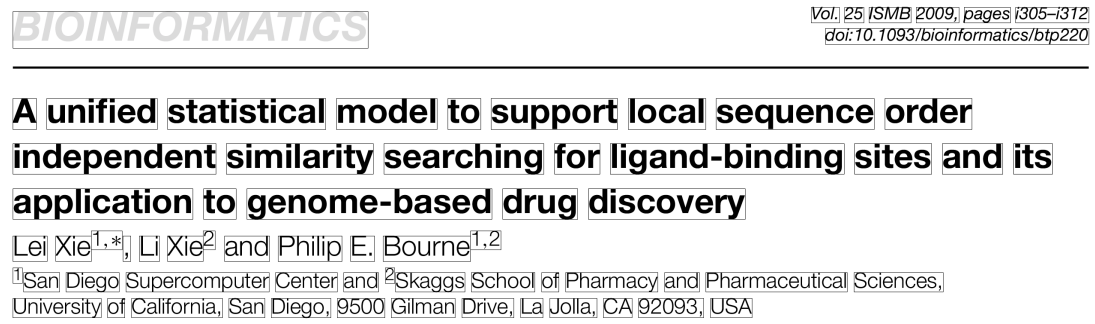
- A need to analyse legacy articles that are mostly scans of paper documents with no selectable text, i.e. no objects in the PDF model apart from page-sized bitmap images;
- The unavailability of reliable PDF rendering libraries;
- The availability of commercial OCR software that are proficient enough to yield satisfactory results and readily applicable over diverse input.

[9]The Elsevier publishing company – `http://www.elsevier.com/`
[10]The Association of Computing Machinery – `http://www.acm.org/`

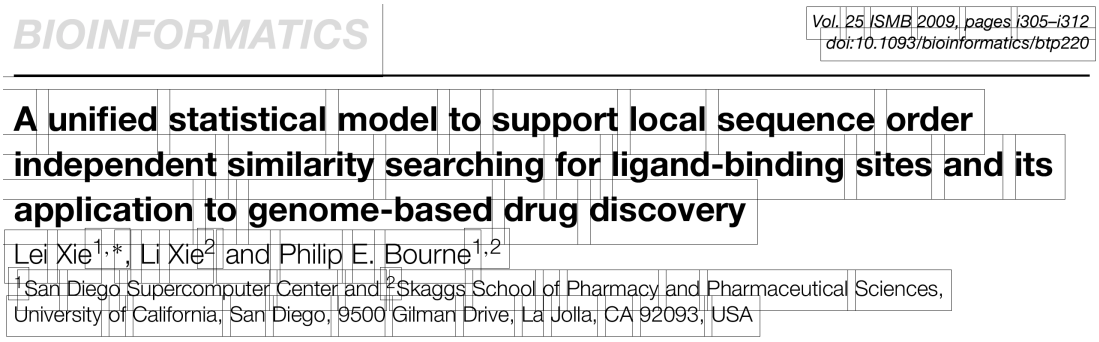
The fact that the PDFs made routinely available by publishers are born-digital, motivated the exploration of a new approach that would alleviate this current reliance on commercial software. Born-digital PDFs readily encapsulate text and positioning information in their object model. Working directly with this type of article has the advantage of font and positioning information being 100% accurate. However, it does not alleviate the need for text and reading order reconstruction, because the PDF is optimised for printing documents to paper, not for machine analysis. Word composition and line segmentation are thus seldom explicitly captured in the PDF, as only the appearance, not the organisation of elements matters when printing. Operations such as de-hyphenation and Unicode normalisation are required for reconstructing the textual content. Additionally, the internal sequencing of symbols from the PDF text stream is not required to follow any particular order, as long as all elements are present prior to rendering; it needs to be disregarded and the intended reading order reestablished.

PDFX does not have its own PDF reading library to access the PDF object model. It uses a component from the Utopia Documents PDF reader (Attwood et al., 2010) to retrieve the positioning information of all words and bitmap images of each page and, additionally, the text content, text orientation and font name for each word. With this basic information, PDFX proceeds to construct a geometrical model of the whole document, and to gather document- and page-wise statistics to guide the selection of constituent blocks for different logical elements in the following steps. Font frequency maps suggest common versus rare features (such as those of the core body text vs. those of a possible title), while a font difference between two neighbouring words functions as an initial indication of two distinct logical units.

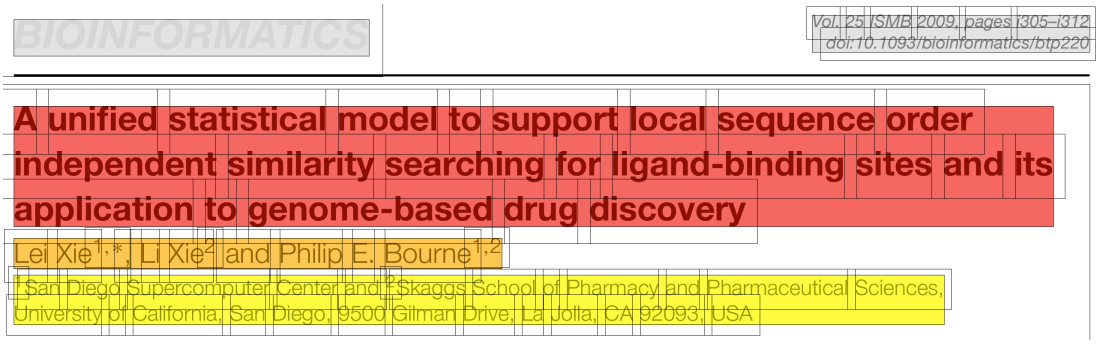
Adjacent words of similar font characteristics are then merged together to form a first set of blocks (contiguous rectangular areas of text) with which logical structure inference will commence. An important aspect is that the merging parameters used are defined relative to the font size and font face of each word, as well as to the spacing between consecutive words and lines. Figure 3.2 shows an illustration of the block construction steps. This approach facilitates tailoring for any logical element type and any article layout automatically, being significantly more flexible than approximating hard-coded numerical parameters.



(a) Initial word boxes, as retrieved from the PDF.



(b) The expansion of word boxes with respect to individual font sizes.



(c) Merging of expanded word boxes with respect to font similarity and intersections.

Figure 3.2: Illustration of the initial construction of blocks. The word boxes in (3.2a) are expanded with font-specific radiuses to yield (3.2b). From these, horizontally-intersecting words of the same style are grouped to form lines, intervening boxes (e.g. superscripts) are subsumed in their respective lines, and same-style lines are then grouped to form blocks (3.2c). Different block colours signify ultimately different logical roles.

3.2.3 Logical Structure Recovery

With the geometrical model and statistics in place, this stage attempts to determine the semantic roles of the newly created blocks, possibly merging them into *logical regions* that may span multiple columns or pages. A single pass through a sequence of steps aims to identify one logical element type at a time, across the whole article, by tagging regions with certain characteristics. The sequence is given in Table 3.3.

Table 3.3: The sequence of element identification steps taken by PDFX for logical structure recovery, along with their average relative processing times.

#	Element type	Time (%)
1.	Body text and reading order	8.5
2.	Bitmap images	1.3
3.	DOI	4.0
4.	Authors	7.1
5.	Title	0.3
6.	Outsiders: headers, footers, side notes, page numbers	3.0
7.	Top-level headings	20.0
8.	Abstract	3.7
9.	Captions	10.3
10.	Lower-level headings	26.4
11.	Author footnotes	3.3
12.	Bibliography and individual bibliographic items	1.2
13.	Remaining body regions	0.1
14.	Tables	1.8
15.	In-text references, URIs and emails	7.8
16.	Labelled formulae (preliminary)	1.2

The relative processing times shown in Table 3.3 highlight the varying levels of computational or logical difficulty associated with the identification of each element type, in the current implementation of the system. For example, captions require multiple attempts to merge their candidate words without collisions with other regions, as they are often part of floating bodies. Similarly, section headings often require several fallback strategies to be identified because of only loosely-defined formatting conventions for these elements. A trade-off between precision and processing time was made in the system design, in that the sequence of identification steps does not reiterate. Instead of employing multiple passes until no more new information is gained, PDFX confers each tag assigned to a region a confidence level. Tags and confidence levels of elements identified up to a certain point are used to make new tagging decisions in the upcoming steps. The confidence level can be either `confident`, to mean that the

region adheres to concrete rules of a specific element type or `possible`, to signify a partial conformation with these rules. Then, as region identification progresses and new `(tag, confidence)` information becomes available, two types of events may occur:

- Certain regions may have their tags or confidence levels changed to reflect their most likely function in the current context. E.g. a `possible` body block may change into a `confident` figure caption;
- Increasingly more difficult element types become identifiable because of new structural and semantic cues. E.g. in-text citations are recognisable only after individual bibliographic items have been identified.

The identification sequence in Table 3.3 is thus carried out in a prioritised manner, the order being dictated by elements' requirements for prior information. The elements considered harder to tag confidently are identified towards the end of the sequence, only after certain helpful identification steps have already been carried out. This creates a chain of dependencies between different elements, as illustrated in Figure 3.3. Apart from the direct dependencies indicated by the arrows in the figure, there are also implicit ones, in that progressively fewer words remain available for the construction of latter elements. This has the beneficial effect of reducing the amount of potential noise in the subsequent steps.

At each step in the sequence, region candidates are constructed in a manner tailored to the particular element type being identified. For example, words belonging to potential captions are allowed to jointly span more than a column in width, whilst words of potential headings are not. In addition, candidate words for the abstract, captions, headings and bibliographic items are also considered among words previously assigned to other elements because of possible font similarities. For example, it is not uncommon for captions or bibliographies to share the same predominant font with the body text. The sequential element identification approach makes it easy to confer such precedence rights over words to element types identified in later stages.

Body text and reading order. The first step in the sequence is to identify the core body text along with the reading order of the blocks. The most frequently used font of lowercased alphabetic words is assumed to be the body font. This is a convenient way of omitting most of the text of large bibliography sections and tabular content that might dominate font statistics in some cases. Out of the set of blocks, those containing

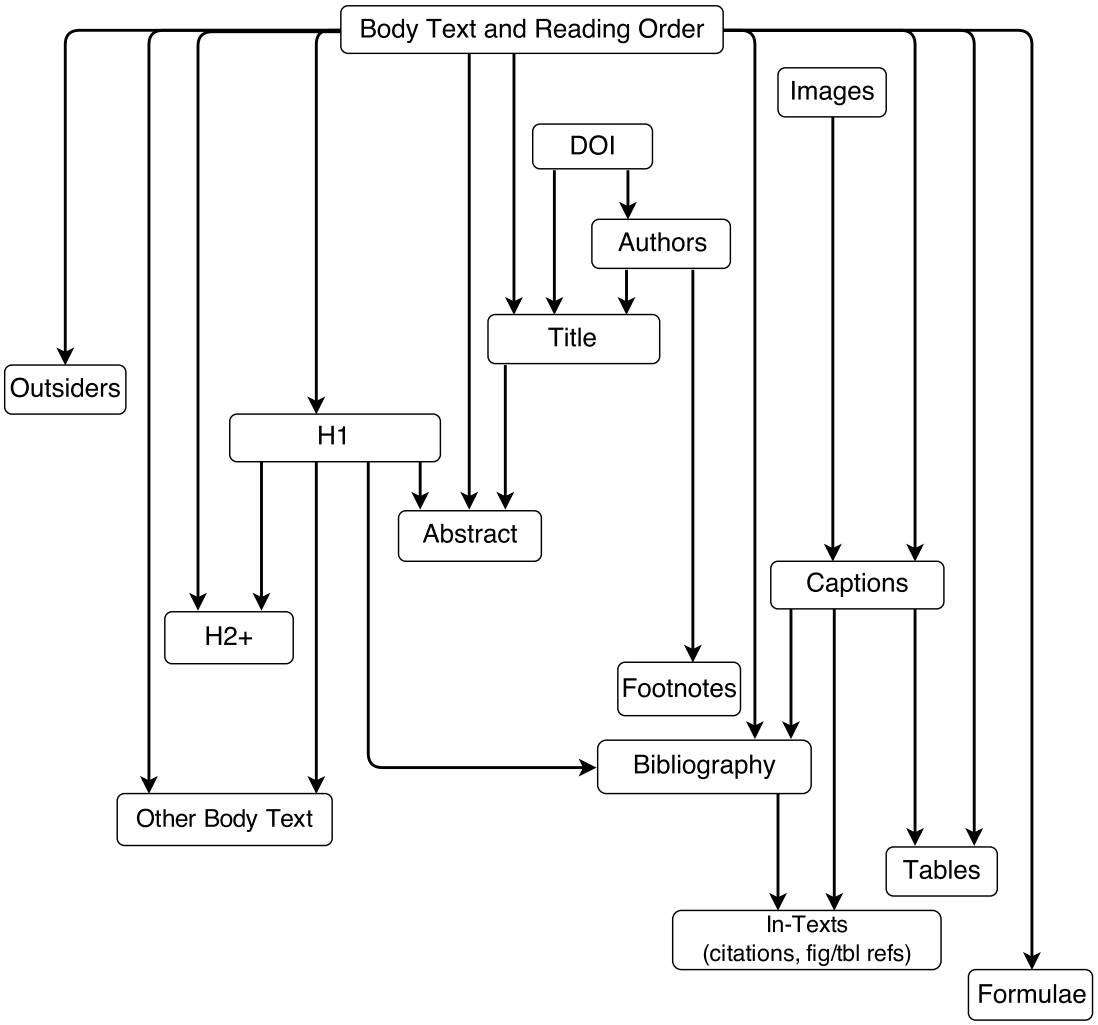


Figure 3.3: Workflow diagram of PDFX's identification sequence, showing the order of processing and the prior information dependencies of different rhetorical elements, that are required for their confident recognition. Identification proceeds from top to bottom, progressively reducing noise for subsequent steps through the assignment of words to previous regions. The nature of the dependencies between elements is explained throughout this section.

primarily words in the body font are tagged as `body` regions and the dominant body region shape is used to determine the columnar layout and the intended reading order of the article.

Bitmap images. As another atomic component retrieved from the PDF, bitmap images are identified and extracted next. Overlapping images are merged and then exported in PNG format. Because plain-text may be overlaid on bitmap backgrounds in figures, any intersecting words are also rendered on top of the graphics prior to exporting.

DOI. A regular expression search for a DOI number is conducted on the first page. If found, the article's metadata it attempted to be retrieved from CrossRef[11] or PubMed via their DOI lookup services. If the lookup was successful, the retrieved metadata will serve as a sanity check for the identification of the title and authors in the following steps. This DOI lookup is optional and the only stage at which PDFX will make use of external services for processing. The speed of the lookup is below half a second on average.

Authors. To help author identification, PDFX makes calls to a local name service to identify words on the first page that are human names. The service implementation is an in-memory hash table of names and a listener for connections. The list of names was compiled from freely available resources such as open access publications and censuses. PDFX will send one string at a time to the name service and expect a response of '1' if that string was found to be a name (i.e. if it occurs in the hash table). If any of the identified names either match DOI metadata or are superscripted, the largest of these will dictate the font for all possible author regions on page. If this information (DOI or superscription) is not available, then the author region exemplar is assumed to be the one having the highest density of names out of all the candidates.

Title. If DOI metadata is available and the DOI title string occurs on page, the respective region is taken to be the title region. Otherwise, title candidates are regions in the top-3 largest fonts on the first page, that contain at least two words not already assigned to author regions. A list of common stopwords such as 'Communication' or 'Original Research' is used to filter out possible large headers. The position on page and the size in number of words are then used to select the most probable candidate for the article's title.

Outsiders. Headers, footers, side notes and page numbers are all considered `outsiders` because of their positioning relative to the core text. These elements mark the end of the ones considered easy to identify. A rectangular outline of the union of all confidently tagged body regions in the article, irrespective of page number, is treated as an overall *document frame* that bounds the narrative's core text. With an added check for a possible odd/even page pan, the document frame facilitates easy recognition of outsider regions – elements notorious for adding noise to plain-text conversions or to attempts of copying cross-column or cross-page content.

[11] The CrossRef DOI Registration Agency – `http://www.crossref.org/`

Top-level headings (`h1`). Section headings of scientific literature are non-trivial to identify robustly because of their high variability in typesetting across publishers. In PDFX, these elements are handled with a combination of linguistic, stylistic and contextual features. For top-level headings, two lists of cue words are used, one for typical, standardly-named headings such as 'Introduction' or 'Conclusions' and one for back-up, variable-sized ones such as 'References' or 'Acknowledgements'. If any contiguous regions contain only typical cue words, the largest of them is taken as a confident `h1` exemplar. Afterwards, all other regions sharing the font and alignment characteristics of the exemplar are also tagged as `h1`s. Since only one such exemplar is required to identify all remaining headings, the procedure is versatile enough to deal with non-standard articles such as those of the life sciences domain, where section naming conventions are not strictly imposed. In such cases, it will be sufficient for the article to just mention e.g. "Background" or "Discussion" for PDFX to retrieve an exemplar `h1` font and search for this throughout the article to identify the remaining elements. In the case of no confident hits using standard cue words, there is one fallback to variable-sized cues, and if this method fails as well, there is a second fallback to regions of the largest remaining font in the document. The set of remaining fonts is built from words not yet assigned to any `confident` region. The PDFX architecture externalises its entire reliance on cue words in the form of a command-line argument, allowing the system to be equally as proficient for any language other than English. In this respect, the Public Knowledge Project[12] (PKP) has kindly contributed cue lists for German, Spanish and Portuguese. PDFX aims to conduct automatic language recognition of the input article as a preprocessing step, in future releases.

Abstract. At this point in the sequence, abstract candidates are built from words that do not belong to any `confident` region and are positioned between the identified title and the first `confident` top-level heading, in the reading order. Confident abstract recognition is steered by the occurrence of the cue words 'Abstract' or 'Summary' (for English) at the beginning of the candidate regions. Abstracts that are not marked specifically in text are disambiguated from neighbouring regions such as institutions, subject descriptors and 'Keywords' sections by inspecting their textual content for length and punctuation.

[12]The Public Knowledge Project – `http://pkp.sfu.ca/`

```
- page 3 - caption candidate 'confident' -  Table 1. PDB chains u...
- page 3 - caption candidate 'confident' -   Fig. 1. Fitting of t...
- page 4 - caption candidate 'confident' -   Fig. 2. The derived ...
- page 4 - caption candidate 'possible'  - Figure 2 shows the der...
- page 5 - caption candidate 'confident' -   Fig. 3. Percentage o...
- page 7 - caption candidate 'possible'  -  table 2 are amongst t...
- page 7 - caption candidate 'confident' -  Table 2. Top 18 most ...
-
- 7 total caption candidates
- Caption font: ('Times-Roman', 8.0)
- Label/text delimiters: {'. ': 5, ' ': 2}. Chosen: '. '
- 5 total captions
- page 3 - Removing intersecting region:
             (body, possible) Table 1. PDB ch...
- page 3 - Removing intersecting region:
             (body, possible)  Fig. 1. Fittin...
```

Figure 3.4: PDFX debug log excerpt of the caption identification stage.

Captions. Captions are identified through the use of regular expressions for finding typical (`<label> <number> <delimiter>`) cues at the beginning of constructed regions. The most frequent font of compliant regions, along with the most frequent `<delimiter>` pattern, help differentiate real captions from plain body text simply beginning with a reference to a figure or a table. Figure 3.4 shows an excerpt from PDFX's debug log that exemplifies this behaviour. It shows how two of initially seven caption candidates were marked as `possible` because their potential labels were only delimited by whitespace and not punctuation. They were afterwards left out because a whitespace delimiter was uncommon amongst all candidates. Some of the words belonging to the remaining candidates had been previously assigned to two `possible body` regions. These intersecting regions were thus discarded in favour of the newly created captions, because of the precedence captions take over body text.

Lower-level headings **(`h2+`)**. If the top-level headings were found to follow a numbering pattern, this pattern is used to recognise lower-level ones. It is not uncommon for first- and second-level headings to share the same font name and size, therefore, if no `h2` candidates were found via a numbering pattern, an attempt is made to identify them amongst existing `h1` regions, possibly separating them by a difference in case (i.e. UPPER CASE vs. Title Case) or emphasis (i.e. **bold** vs. regular or **bold** vs. *italics*). In addition, similarly to the top-level heading identification step, there are two fallback mechanisms. The first exploits knowledge of the current context, examining yet-unidentified regions positioned in between a confident top-level heading and a

`body` region. Font difference and region size in number of words are used to identify possible candidates. Finally, a check of the beginning of `body` regions is also conducted, in case lower-level headings were typeset inline with the body text, but with a different font face.

Author footnotes. Author footnotes are identified through a superscript pattern applied to the end of words making up author regions. Any identified superscripts are searched for elsewhere on the first page and text blocks found to start with the superscripts are marked accordingly.

All words left unassigned to regions by this point in the identification sequence are merged together loosely, regardless of a difference in font. A first goal of this procedure is to help recognise the bibliography and tables correctly, as these elements are well-known for having diversely-styled contents that do not fit within typical stylistic constraints. A secondary goal is to have neighbouring words ultimately left unassigned encapsulated into contiguous `unknown` regions in the output XML.

Bibliography and bibliographic items. These elements are given special attention because they are particularly valuable in scientific literature analysis systems. The bibliography section itself is fairly easy to identify because of its positioning within the article and of common section heading cues, e.g. 'References' or 'Bibliography', for English. Contiguous body-like blocks are taken to be bibliography block candidates if they occur between the found heading and the next top-level heading in reading order, if any. Spatial alignment and textual cues are then gathered to help segregate individual bibliographic items (`bib_items`). Spatially, `bib_items` can be separated by a difference in indentation level or vertical gaps. Textually, they can be separated by one of three delimiters: a *numeric* delimiter, a *bracket* delimiter or a *name* delimiter. The first two cases are straightforward to handle because consecutive cues are easily identified. Name delimiters in PDFX are considered to be lines with identified proper names in them, or with the common cue 'et al.'. Once a `bib_item` beginning has been identified, words in the reading order are collected for this element until the next delimiter or the end of the section is encountered. As a sanity check, and to help identify column- or page-split elements, `bib_items` are required to have a year in them if they are to be considered confidently tagged.

Remaining body regions. This step means to simply tag any yet-untagged elements structurally similar to body regions as `possible` body regions themselves. A case in which this step is useful is the formatting of the 'Methods' section differently from the

rest of the body text, usually in a smaller font. This typesetting choice is customary in biomedical literature and can easily confuse structure recognition software.

Tables. Table recognition is a well-documented, still active field of research in its own right because of the inherent difficulty of the task (Zanibbi et al., 2003). In PDFX, this step is carried out in a standard way, using heuristics for cell construction and the constraint that tables have accompanying captions. The step was placed towards the end of the processing sequence in order to also narrow down the set of possible words that make up tabular structures. For each identified caption, multiple attempts are made at building table variants, from neighbouring words having a combination of the following features:

- Coming before or after the table caption in reading order;
- On the same column as the caption or ignoring the column boundaries;
- Currently unidentified or currently assigned to `possible` regions.

Up to 8 table variants may thus be built per table candidate, depending on the words adjacent to its caption. An alignment score is computed for each variant to suggest the probability of their constituent words following a tabular arrangement. The `left`, `centre` and `right` coordinates of each word are first aggregated to yield table-wide statistics. Afterwards, each word is considered to contribute to the table's alignment score with its "best fit" in these statistics. For example, if a word's `centre` coordinate is seen more often overall than its `left` and `right` coordinates, the word is assumed to be centre-aligned within this table candidate. Its `centre` coordinate is then collected in a new aggregation of "best fit" statistics. The alignment score is given by the ratio between the total number of words with their "best fit" coordinate occurring more than once in these statistics (to leave out outliers), and the number of distinct "best fit" coordinates. This computation will yield a higher score, for example, for a single-column table variant than for a two-column one that incorrectly subsumes a paragraph of justified text. Even though the paragraph has two extra column candidates of left- and right-aligned words, its remaining words will contribute more negatively to the alignment score. The table variant with the highest score is then considered to be the table region, and heuristics are finally applied to its words to identify probable headers and cells. Whitespace rectangles spanning the height and width of the table are constructed to identify cells, and rows of unusual sizes are marked as headers[13].

[13]The cell construction heuristics were contributed by Florian Thomas (`florian.thomas@live.fr`).

In-text references. In-text references to figures and tables are taken to be all (<label> <number>) parts of caption candidates that were not ultimately tagged as captions, such as the two examples from Figure 3.4. Grammars for detecting citations based on bibliographic cues have been constructed for each of the three reference delimiter types mentioned earlier (numeric, bracket and name). If the elements of the bibliography were separated by numeric or bracket delimiters, the bibliographic cues of their respective citations in text are the delimiters themselves. In the attempt to identify citations pointing to name-delimited bib_items, the method presented in Powley and Dale (2007) is used, based on (<first-author> <year>) patterns to construct each cue. The system can identify single, compound, as well as ranged citations, e.g.

- [Altschul 1997(a)]
- (Altschul et al., 1997; Claverie, 1994)
- [2–10]

A caveat of the approach is that, because citation finding is based on bibliographic cues, only citations for identified bib_items will be recognised. However, the ones that are recognised will always link back to the bib_items they refer to, through an rid attribute in the output XML. For example, the citation

- <xref ref-type="bibr" rid="R1">Altschul et al., 1997</xref>

corresponds to

- <ref rid="R1">Altschul S.F. et al. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res., 25, 3389-3402.</ref>

Labelled formulae. Labelled formula identification is the last step in the sequence and the most recent addition to PDFX's functionality. The constructs are identified by searching for probable bracketed labels at either end of each line, and inspecting the line's density of digits and mathematical symbols. All similar neighbouring lines to the one with the label are merged together, in an attempt to cover multi-line formulae as well. Presently, the system limits itself to identifying the formula regions in the PDF and rendering them as images to capture their original layout. There is no attempt to reconstruct MathML or LaTeX representations of the formulae, like for example in (Yoo and Kim, 2013). In a similar fashion to figures, formulae in the XML will appear encapsulated in <disp-formula> tags that contain their respective labels and rasterisations.

3.2.4 Output

The end result is constructed using the tags that regions were left with at the end of the processing sequence. The initially identified contiguous blocks, now encapsulated in logical regions, jointly reconstruct the rhetorical structure of the article. Information about the different regions and their organisation is represented using an XML format very close in schema to JATS XML[14]. Logical `section` elements implied by the heading hierarchy are added in and populated during the XML construction. As regions can span multiple blocks, columns or pages, their respective XML elements may contain tags that act as physical position markers in the original text. Figure 3.5 shows an XML excerpt of a region spanning two blocks.

```
<region class="DoCO:TextChunk" page="55" column="1">
  <xref ref-type="fig" class="deo:Reference">Figure 3.5</xref>
  shows excerpt of a region spanning two blocks.
  <marker type="block"/>
  The class attribute of each element in the XML [...]
</region>
```

Figure 3.5: Example of a PDFX XML region spanning two blocks.

The `class` attribute of each element in the XML has been added to facilitate further interoperability with other services. This attribute is derived from the tags of regions and set in accordance with DoCO[15]. DoCO is an ontology of both physical and logical components of bibliographic documents, well-suited for linking structure recognition output such as that of PDFX to other text processing pipelines. A multitude of different-purpose workflows can treat the PDF-to-DoCO-compliant-XML conversion as a preprocessing step, to greatly widen their application domain in terms of accepted input. This is particularly useful for tasks that require only specific parts of the document as input. Several real-world use cases for which PDFX is currently being used are given in Chapter 5.

For an overview of the entire structure recovery process, Appendix A provides an example of PDFX's analysis of a scientific article, in the form of a verbose, human-readable log that marks the transitions between the aforementioned steps. An excerpt

[14]The Public Knowledge Project (PKP) has generously contributed an XSL solution to transform PDFX XML into NLM 3.0 XML (a precursor of JATS). This script has been revised and made available at `http://pdfx.cs.man.ac.uk/usage`, in hopes of fostering the rapid integration of PDFX's functionality with the many existing NLM XML processing services.

[15]DoCO, the Document Components Ontology – `http://www.purl.org/spar/doco`

from the XML output produced for the same article is also given.

Lastly, some additional features of PDFX found to be of value to users are:

- The dehyphenation of line-split words to reconstruct the reading flow. If an end-of-line hyphen is detected, the candidate fused word formed from the two sides of the hyphen is searched for anywhere in the document. If encountered, the fused word will be displayed in the output XML. This procedure can also be extended to include a check if the fused word occurs in a dictionary, for example, in the standard Unix file `/usr/share/dict/words`.
- Unicode normalisation, to reduce inconsistencies between the visual appearance of characters, and the way in which they were encoded in the PDF. For example, the Angstrom sign ('Å') might be displayed in the PDF as the letter 'A' with the ring operator ('˚') drawn directly above it. Even though it may look like an adequate Unicode character on page, the respective text stream extracted from the PDF will be a 3-character sequence '`A ˚`' or '`˚ A`'. PDFX attempts to remedy such cases by reconstructing characters when possible.
- The optional ability to output sentence-level tags in the XML, using the Punkt tokenizer (Kiss and Strunk, 2006) to conduct the sentence splitting.
- The optional DOI resolution of individual bibliographic references, using CrossRef's Metadata Search service[16].

3.3 Evaluation of PDFX

The standard evaluation procedure was to obtain Precision (P), Recall (R) and F_1 measures for the output regions, given manually created gold-standard XML versions of articles. These files, considered to be perfect conversions, helped determine which tag assignments made by PDFX were correct (true positives - TP), which were incorrect (false positives - FP), and which elements had been left unidentified (false negatives - FN). The well-known P/R/F_1 combination has also been used by previous efforts concerned with structure recovery. The formulae for these metrics are recounted below, for convenience:

[16]CrossRef Metadata Search – `http://search.crossref.org/`

$$P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{TP + FN} \qquad F_1 = \frac{2 * P * R}{P + R}$$

A direct string comparison for exact matches was infeasible, because of possible small discrepancies between texts of the PDFs and those of the gold-standards, such as cases, hyphens, typos or punctuation. In addressing this issue, the Ratcliff/Obershelp string similarity metric (Ratcliff and Metzener, 1988) was used to count as a correct match any extracted element found to be at least 95% similar to its gold-standard counterpart. Similarity computation works by treating the longest matching subsequence of two strings as an anchor, counting its characters, and recursively repeating the procedure to the left and the right side of the anchor, until there is nothing left to examine. The output similarity ratio is twice the total number of characters counted in this way, divided by the total number of characters in the two strings.

3.3.1 Datasets

A total of four datasets were used in evaluating PDFX's performance, in order to cover a more representative sample of scientific research output, both in terms of layout variety and domains of study. The demographics of the datasets are given in Table 3.4. Results for two of these datasets, namely Luong et al.'s and PMC sample, were also presented in Constantin et al. (2013) and processed with PDFX v1.5. In the evaluation presented in this dissertation, the collections have been reprocessed with the latest version of PDFX at the time of writing (v1.9).

Table 3.4: Demographics of the article collections used in the performance evaluation of PDFX. Consecutive columns represent the sizes of the datasets progressively filtered for what was considered outside the scope of this study.

Dataset	**Initial size**	**+gold**[a]	**+type**[b]	**+page**[c]	**+valid**[d]	**+lang**[e]
Luong et al.	40	40	40	40	39	39
PMC sample	2135	2135	1957	1943	1943	1943
PMC	346555	306131	267353	265582	246707	246668
Elsevier	368777	362591	220759	220250	217947	215517

[a] PDFs with associated gold-standard XMLs.
[b] Of in-scope article type, e.g. 'research article', 'case report'.
[c] Within a 2–50 page range.
[d] PDFs born-digital, correctly encoded characters, full-length gold-standards.
[e] In English, if language information was present in the XML.

The first dataset facilitates the comparative evaluation between PDFX and the state-of-the-art machine learning system SectLabel introduced previously. It was taken from the original SectLabel publication (Luong et al., 2011) and comprised 40 articles from the field of Computer Science. As mentioned in the original article, the collection *"includes 20 ACM papers spanning various years and venues, 10 papers from the 2009 Proceedings of the Association for Computational Linguistics Annual Meeting, and 10 papers from the 2008 proceedings of the ACM Conference on Human Factors in Computing Systems"*. Annotated versions of the articles were manually created by Luong et al. and made available at `http://wing.comp.nus.edu.sg/parsCit/`. Processing this dataset offered an interesting view of how PDFX's rule set matched up against a trained model solution. One article of this dataset was not retrievable in born-digital form and was consequently left out of the evaluation.

The next two datasets were chosen for the variety of their articles' layouts. They were compiled from a May 2011 snapshot of the PMC Open Access Subset (OAS)[17]. PMC is a full-text archive of biomedical and life sciences journal literature of the NLM. Each publication in the OAS is available in both PDF and XML formats. NLM has created the Journal Archiving and Interchange Tag Suite[18] to define XML elements and attributes that describe the content and metadata of journal articles. This permitted a straightforward aligning of the NLM tag set to the simplified tag set devised for PDFX output. The entire OAS, at the time the snapshot was captured, comprised nearly 350k documents. After filtering out what was considered outside the scope of the study, 246k articles remained (please see Table 3.4).

To facilitate a thorough evaluation, a sample of the full PMC dataset was also extracted and considered separately. This smaller collection consisted of the latest publication of every distinct journal in the OAS snapshot, so that the many different document styles would be represented as evenly as possible. This fostered the inspection of PDFX's performance on PMC articles without skewing the results in favour of styles of journals with high throughput. This sample dataset contained 1943 articles in total. The collection is available for download at `http://pdfx.cs.man.ac.uk/serve/PMC_sample_1943.zip`. The archive contains the PDFs, the gold-standard NLM XMLs and PDFX's corresponding output (1.5GB in size).

The fourth and last dataset was chosen for being considered a representative part of

[17]The PubMed Central Open Access Subset – `http://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/`

[18]The NLM Journal Archiving and Interchange Tag Suite – `http://dtd.nlm.nih.gov/`

yearly published research. It was taken from all of Elsevier's publications from the year 2008, kindly provided under research license by the publishers. It was filtered in the same manner as the PMC collections, ultimately comprising 215k articles. In contrast to the PMC datasets, style change was not as common here (being the output of a single publisher), but the topic coverage was significantly wider. This in turn implied variety in terms of sectioning, figure and table use, and overall writing style. Since Elsevier is the world's leading publisher of scientific literature, processing its output was considered relevant to the present study. The company was found to have a market share of around 25% in the field of science, technology, and medical publishing, being about three times as large as its closest competitor (Corty, 2010). Its document styling is therefore amongst the most frequently encountered, making a measurement of performance over it befitting.

A remark made during the evaluation phase of PDFX was that the manually constructed XMLs did not necessarily match the content of their respective PDF versions. Either because of character encoding issues, human error or intent (such as changing the title of the camera-ready PDF at the last minute), the two variants differed at times. The Discussion section of this chapter (Section 3.3.3) exemplifies such inconsistencies. Differences in content were a caveat of the two representations being generated and maintained independently of each other. An analysis of arXiv.org LaTeX-generated PDF articles was considered for alleviating this issue, but available LaTeX parsers were found to be inadequate in recovering a sufficiently rich rhetorical structure for a thorough assessment. The shortcoming was instead addressed by evaluating the results with two similarity thresholds: 0.95 and 0.8, respectively. Differences in reported accuracy were then inspected to help identify the elements that PDFX might still have extracted correctly.

3.3.2 Results

The XML outputs for the comparative evaluation were manually inspected for the 14 elements that made up the intersection between the SectLabel system's capabilities and those of PDFX. The elements were *title, author, abstract, top-, second- and third-level headings, references (bibliography), body text, page numbers, emails, figures, tables,* and their respective *captions*. The F_1 measure results are presented in columns 2 and 3 of Table 3.5 and also illustrated in Figure 3.6. In the table, scores of elements at which

PDFX equalled or outperformed SectLabel are shown in bold. For a clear overview of the achieved performance, both macro and micro F_1 averages are presented. Macro F_1 weighs each element type equally, being an average of element-specific F_1 scores. Micro F_1 weighs each article equally, being an average of paper-specific F_1 scores. In the figure, elements have been re-sorted to better highlight the strengths and limitations of PDFX when compared with the more targeted, trained model solution.

The results for the remaining three datasets are given in columns 4–6 of Table 3.5. The evaluation process for these collections was carried out automatically with a 0.95 similarity threshold, as measured by the Ratcliff/Obershelp method. The highest performance across the datasets is shown in bold. The evaluated elements in this case differed somewhat from the ones of the comparative evaluation. Body text accuracy could not be determined reliably, because the PDFX XML did not contain individual paragraph tags like the gold-standard XMLs, whereas considering entire sections would have been inconclusive due to the many possible intervening elements. Figures and page numbers could also not be checked, as they were unavailable in the gold-standard XMLs. In addition to the elements in common with SectLabel however, the performance for in-text citations and figure or table references is reported.

Lastly, because of the previously mentioned caveat of an automatic evaluation, Figure 3.7 shows bar graphs of the results obtained over the three datasets, with both the 0.95 and 0.8 thresholds, to highlight additional possible true positives retrieved by PDFX.

3.3.3 Discussion

Manual Comparative Evaluation

The conducted comparative evaluation against SectLabel yielded quite promising results and also highlighted some interesting facts. Despite the evaluation having been carried out on a dataset for which SectLabel was trained, PDFX managed to keep up with the performance of its learned model counterpart and even outperform it on 3 out of 14 elements (second-level heading, third-level heading and figure caption). At title and top-level heading identification, both systems performed the same. PDFX was only marginally behind the SectLabel system for 7 elements, but clearly behind for another 2 – figures and tables. These elements mark the areas in which the detailed

Table 3.5: Structure analysis performance results over four datasets, given as F_1 scores. The Luong et al. dataset was used for comparing PDFX with SectLabel. PDFX was additionally run over the PMC, PMC sample and Elsevier datasets. The best results of PDFX for each evaluation type are in bold.

Evaluation Type	**Manual**		**Automatic**		
Dataset	**Luong et al.**		**PMC**	**PMC Sample**	**Elsevier**
Element \ System	**SectLabel**	**PDFX v1.9**			
title	100	**100**	64.71	87.53	**94.98**
author	97.94	94.87	67.53	88.84	**92.73**
abstract	100	96	48.56	49.56	**71.41**
h1	93.51	**93.51**	84.17	82.55	**91.68**
h2	91.39	**92.98**	30.82	31.58	**80.04**
h3	81.69	**96.50**	1.74	2.91	**75.99**
reference (bib_item)	99.5	98.71	81.01	**80.80**	77.56
body	96.97	91.38	–	–	–
in-text citation	–	–	69.25	66.95	**72.33**
figure caption	76.91	**83.12**	47.42	52.95	**67.57**
table caption	80.69	80.10	47.42[a]	52.95	**67.57**
figure/table reference	–	–	51.26	**76.61**	64.93
table	79.59	69.44	**16.02**	9.61	4.94
figure	79.93	52.38	–	–	–
page	97.84	97.44	–	–	–
email	97.64	91.53	66.10	84.75	**94.27**
Macro F_1	90.28	88.43	52.38	59.55	**74.04**
Micro F_1	N/A	88.08	55.48	66.28	**75.74**

[a]Figure and table captions were not differentiated in the automatic evaluation.

visual analysis provided by an OCR system and prior learning of the layout specifics of articles prove valuable. Knowing style and layout information of tables beforehand is quite helpful, as these usually vary greatly in this respect and are hard to pin-point robustly using rules. The low accuracy for figures was due to them being often made up of plain-text with some possible vector graphics as background. PDFX v1.9 did not handle non-bitmap figures, thus their content was interpreted as text. This in turn had a negative effect on body text recognition, because more false positive body blocks were identified. The only bitmap images wrongly interpreted as figures were organisation logos, because there was no restriction imposed on the position of a figure on page, nor a requirement that every bitmap have an associated caption.

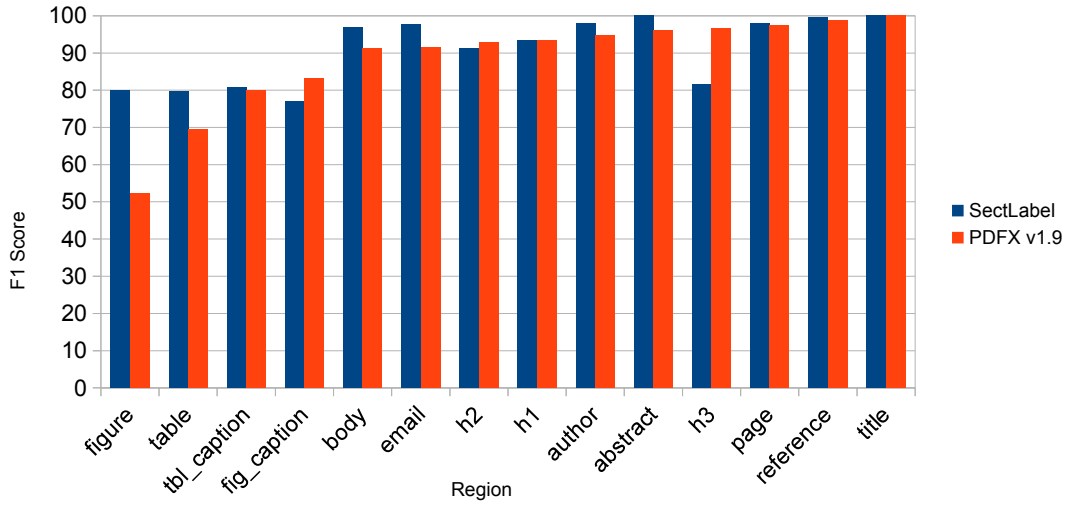
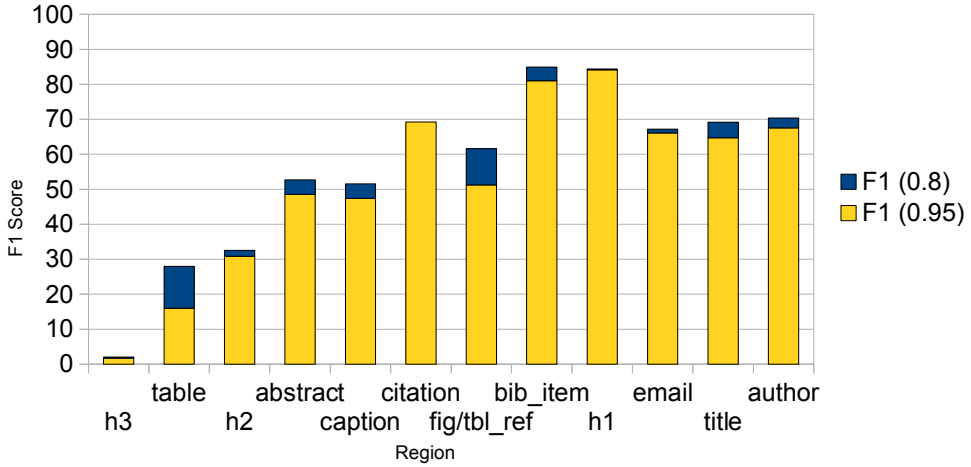


Figure 3.6: Bar graph view of the results in Table 3.5 for the the Luong et al. dataset (columns 2 and 3). Comparison between SectLabel and PDFX.

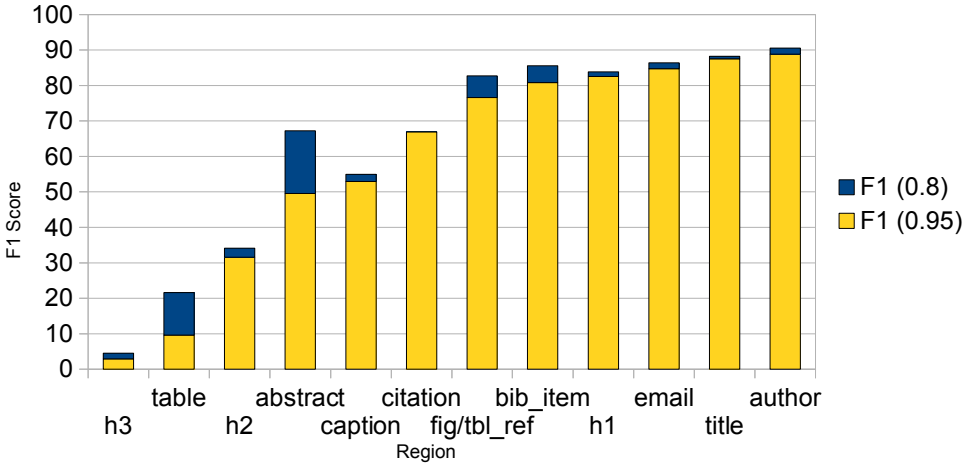
Automatic Evaluation

For the performance evaluation over the other datasets, the automatic fuzzy matching without manual inspection of the results, while satisfactory at times, was generally unforgiving. Upon manual inspection of fault examples, some recurring discrepancies between the PMC PDFs and the ground-truth XMLs were noted, that contributed negatively to performance metrics. Several examples are given below.

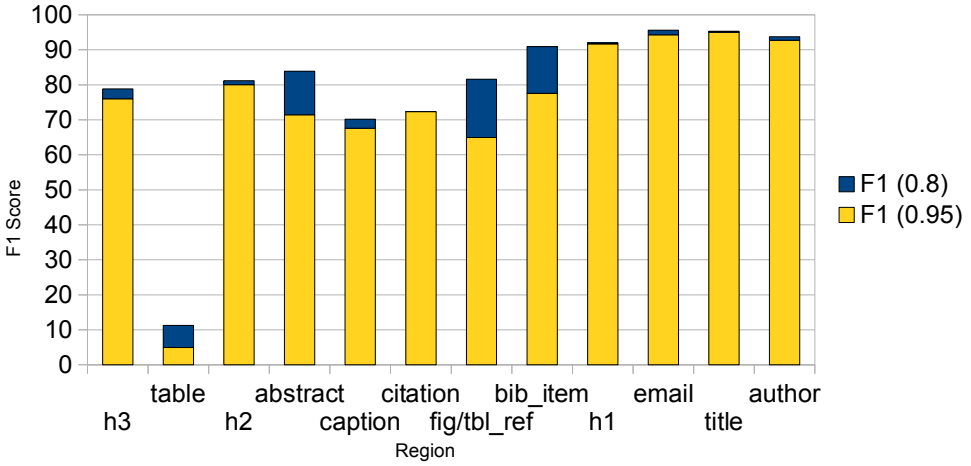
- Differences in character encodings or in the way the curator chose to translate special Unicode characters, e.g.
  - β (beta) / ß (sharp 'S') inconsistency (e.g. PMCID: 3315558)
- Differences in formatting bibliographic items or author names, e.g.
  - extra reference information in the XML (e.g. PMCID: 3081056)
  - author initials merged in the PDF (e.g. PMCID: 3036981)
- Subsections or text blocks missing either from the PDF or XML, e.g.
  - extra 'Journal Club' block in the PDF (e.g. PMCID: 3088469)
  - 'Author Details' back matter block of the PDF, included in the front matter of the XML (e.g. PMCID 3112439)
  - missing section headings in the PDF (e.g. PMCID 2876849)



(a) PMC dataset results.



(b) PMC sample dataset results.



(c) Elsevier dataset results.

Figure 3.7: Differences in PDFX performance results when using a 0.8 similarity threshold for judging a correct match, instead of a 0.95 one. Results shown are F_1 scores for the PMC (3.7a), PMC sample (3.7b) and Elsevier (3.7c) datasets.

- Misspelled words in one but not the other file.

Because of these factors, the results of the automatic evaluation can be considered an effective underestimate of PDFX's true performance. Reassessing the results with a 0.8 similarity threshold in addition to the 0.95 one offered insight on the elements that might still have been correctly identified. The average performance increase was of 4.3 macro F_1 points, with strong emphasis on four elements. Those with more textual content, abstracts and tables, hence more chances for PDF/gold-standard XML discrepancies, saw an 11.5 and a 10 F_1 point increase on average, respectively. In-text references to figures and tables saw a similar 11 F_1 point increase. In contrast to abstracts and tables, these elements had minimal text content, making even small discrepancies affect the similarity ratio considerably. The other noticeable difference was for bibliographic items (7.4 F_1 point increase). For these elements, depending on publishers' formatting rules, abbreviation use and the ordering of their constituent parts were often inconsistent between the XML and the camera-ready PDF.

Additionally, on the PMC sample dataset, the present evaluation also shows an average increase in performance over the results presented in Constantin et al. (2013) (obtained with PDFX v1.5). This gain is mainly due to two factors:

1. A better handling of the front matter in the newer version of PDFX;
2. A more detailed examination of the variations in gold-standard NLM XML formatting and their consideration in the evaluation script; for example, the annotation of figure and table references was often inconsistent, done either on labels and numbers (e.g. `<xref>Figure 1</xref>`), or just on the numbers (e.g. `Figure <xref>1</xref>`).

For both PMC datasets, an inspection of faults in author identification revealed that ~20% of them shared the same font name or size with the immediate-neighbouring affiliation field, which PDFX does not yet handle. Because of this, the two elements were sometimes merged together prior to outputting to XML and jointly considered the author region, resulting in a faulty match. Identifying affiliations is currently marked as future work in PDFX's development, and will likely improve author identification considerably.

Across all datasets, table accuracy was poor. This was mostly due to the current PDFX rule set for reconstructing tables not being more involved. Solving the table recognition problem is known to be an intricate task that requires special attention (Zanibbi et al.,

2003). When checking results automatically, the performance also dropped because the text in table cells was retrieved in different orders between the two XML versions, failing the 0.95 and even 0.8 similarity checks. Still, in many cases, PDFX was able to approximate the rectangular regions where the tabular content was situated. This information can help more sophisticated methods of table recognition, as it simplifies the problem to finding the most likely tabular arrangement of a set of words given accurate positioning data.

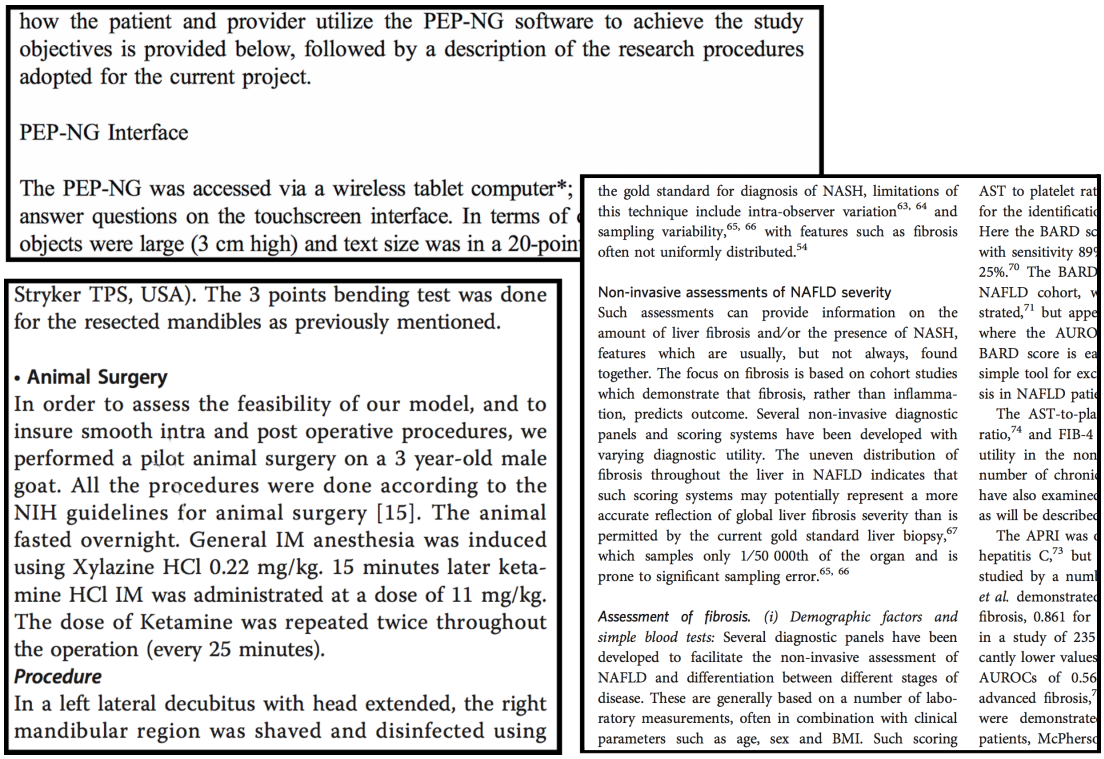
Lower level section heading identification was exceptionally poor for the PMC datasets. Inspection of failures to retrieve these elements revealed that ~30% of them were due to one of two possible reasons, both stemming from the system's reliance on uniform stylistics across the whole document:

- Sections such as 'Author contributions' were marked as top-level headings in the ground-truth XMLs, but occurred in a smaller font than their siblings in the PDF, thus being treated as lower-level headings.
- Elements such as 'Conflicts of Interest' or 'Open Access' were written in the same font as headings, but were not mentioned as such in the XML.

Additionally, some headings were found particularly difficult to separate from neighbouring regions, because of minimal stylistic differences. The recognition of these elements was considered non-trivial even for a human at times, upon manual inspection of problematic cases. Figure 3.8 illustrates some examples.

Citation identification was quite stable between the two similarity thresholds, achieving a 70 F_1 score on average. This suggests that PDF/gold-standard XML discrepancies were not an issue in this case. Examining examples of failures to identify citations revealed a general case that PDFX did not yet handle: citations occurring as superscripts in the PDF. Superscription usually meant that the citations did not need placing between square or round brackets as is customary and as the rule set assumed them to be. The consideration of this case in future work is likely to increase the overall performance for citations, elements for which the text mining community has shown increased interest in recent years (Ciancarini et al., 2013; Di Iorio et al., 2013).

The highest results for the automated evaluation were predominantly obtained for the Elsevier dataset. This comes to confirm that the styles used across the collection are likely less varied than for the other datasets, but also that the XML curation level and formatting rules to which Elsevier publications abide is stricter.


how the patient and provider utilize the PEP-NG software to achieve the study objectives is provided below, followed by a description of the research procedures adopted for the current project.

PEP-NG Interface

The PEP-NG was accessed via a wireless tablet computer*; answer questions on the touchscreen interface. In terms of objects were large (3 cm high) and text size was in a 20-poin

Stryker TPS, USA). The 3 points bending test was done for the resected mandibles as previously mentioned.

• **Animal Surgery**

In order to assess the feasibility of our model, and to insure smooth intra and post operative procedures, we performed a pilot animal surgery on a 3 year-old male goat. All the procedures were done according to the NIH guidelines for animal surgery [15]. The animal fasted overnight. General IM anesthesia was induced using Xylazine HCl 0.22 mg/kg. 15 minutes later ketamine HCl IM was administrated at a dose of 11 mg/kg. The dose of Ketamine was repeated twice throughout the operation (every 25 minutes).

Procedure

In a left lateral decubitus with head extended, the right mandibular region was shaved and disinfected using

the gold standard for diagnosis of NASH, limitations of this technique include intra-observer variation[63, 64] and sampling variability,[65, 66] with features such as fibrosis often not uniformly distributed.[54]

Non-invasive assessments of NAFLD severity

Such assessments can provide information on the amount of liver fibrosis and/or the presence of NASH, features which are usually, but not always, found together. The focus on fibrosis is based on cohort studies which demonstrate that fibrosis, rather than inflammation, predicts outcome. Several non-invasive diagnostic panels and scoring systems have been developed with varying diagnostic utility. The uneven distribution of fibrosis throughout the liver in NAFLD indicates that such scoring systems may potentially represent a more accurate reflection of global liver fibrosis severity than is permitted by the current gold standard liver biopsy,[67] which samples only 1/50 000th of the organ and is prone to significant sampling error.[65, 66]

Assessment of fibrosis. (i) Demographic factors and simple blood tests: Several diagnostic panels have been developed to facilitate the non-invasive assessment of NAFLD and differentiation between different stages of disease. These are generally based on a number of laboratory measurements, often in combination with clinical parameters such as age, sex and BMI. Such scoring

AST to platelet rat
for the identificati
Here the BARD sc
with sensitivity 89
25%.[70] The BARD
NAFLD cohort, w
strated,[71] but appe
where the AURO
BARD score is ea
simple tool for exc
sis in NAFLD pati

The AST-to-pla
ratio,[74] and FIB-4
utility in the non
number of chroni
have also examine
as will be describe

The APRI was
hepatitis C,[73] but
studied by a num
et al. demonstrate
fibrosis, 0.861 for
in a study of 235
cantly lower values
AUROCs of 0.56
advanced fibrosis,
were demonstrate
patients, McPhers

Figure 3.8: Examples of minimally emphasised second- and third-level section headings. Spacing and font differences are sparingly used.

3.4 Summary

This chapter has presented a versatile solution for the recovery of the fine-grained rhetorical structure of PDF articles. The solution is rule-based, utilising only information inherent in the PDF format and generic typographic conventions of scientific literature to carry out the task. A description of the PDFX system that implements this approach has been given, along with an analysis of the logical components of articles, and the relationships between them that can be exploited to reconstruct the entire document structure. The system carries out a single pass through an element identification sequence, having a current running time of 20-30 seconds for a typical 10-page article. The attribution of tags and confidence levels to individual elements fostered the modelling of an article's logical structure in a straightforward manner. Tag assignment decisions were made at each step in the identification sequence, out of a number of possible alternatives.

In a manual comparative study against the state-of-the-art, PDFX achieved competing results, without requiring prior knowledge of the articles' formatting specifics. The

evaluation was also useful in highlighting document elements that are more difficult to identify using generic rules. However, because of the way in which PDFX uses all available contextual information throughout its identification process, even minimal tailoring towards a specific layout can improve performance considerably. For example, PDFX does not currently account for the case in which publishers prepend their own journal-specific first pages to PDF articles, as was the case for 3 PDFs of the Luong et al. dataset. An extra check for the most probable front page would increase the identification accuracy for all front matter elements.

In addition, an automated performance evaluation on a much wider range of publications from the PMC OAS and Elsevier was also conducted. Several elements of interest such as titles, top-level headings, references and in-text citations were extracted with good accuracy (70–95 F_1 points), but overall, there is certainly still room for improvement. Nonetheless, given the shear size and complexity of these real-world datasets, as well as the possible inconsistencies between the PDFs and gold-standard XMLs, the results offer insight into the true potential of the proposed methodology. The many discovered interdependencies between elements imply that any future improvements in recognising one element will also positively affect several others that follow in the identification sequence. For example, a more accurate identification of captions will lead to a better recognition of tables, which may in turn contribute to better recognition of formulae, if the noise coming from unidentified table data is removed. The devised sequence of steps is largely imposed by the logical difficulty in accurately identifying each element type automatically. This makes any experimentation with the order of identification steps difficult, as it would require non-trivial alterations to the existing rule sets. Without such alterations, possible changes would be limited to groups of elements that seldom influence each other, such as the front and back matters, unlikely to produce visible differences in performance.

Another issue remaining for PDFX to handle is the identification of floating bodies such as block quotes or information boxes that are more often seen in magazine-style articles. As they can occur anywhere on page, and body text normally wraps around them, they are often wrongly subsumed into neighbouring regions. However, because these elements are usually highlighted for readability, one possible solution would be to distinguish them by their background colour and consider them obstacles when constructing text blocks.

Another beneficial aspect of PDFX's rule sets is that they allow most element types to

remain unidentified at the end of a run. This means that types of documents other than scientific articles could also be processed, without too many alterations to the current processing logic. Books, for example, would primarily only need a more elaborate consideration of the front matter (forewords, publisher information, quotes, etc.) and a consideration of chapters as separate documents in themselves, that might have their own bibliographies and citations. Newspapers, on the other hand, would have to discard the requirement of consistency in body text and section heading stylistics, and attempt to retrieve a title and author for every identified text region (i.e. article).

The success of PDFX in practice is also attested by its growing number of users. The system has had over 170k online submissions from more than 8.5k unique IP addresses in the second half of the year 2013. In addition, several research collaborations have been established to integrate PDFX functionality within institutional information systems and text mining projects of different scales. Usage statistics of the system in this setting are well in excess of half a million articles.

The system is suitable for use as the preprocessing stage of many specialised text analysis workflows, as it conveniently converts full-text PDF articles into structure-rich, JATS-compliant XML equivalents. In the XML, the elements are annotated with classes of the DoCO bibliographic ontology for interoperability with other services, collectively adding a valuable metadata layer to the original publication. The release of such article metadata in machine-readable form is perceived as one of the six golden rules that semantic publishers should adhere to (Shotton, 2009), because it fosters a high level of interaction with the articles' contents.

The next chapter covers the topic of keyphrase extraction from scientific publications, showcasing a solution that makes use of PDFX's functionality to obtain information about the rhetorical structure of the content being analysed. Superfluous regions such as headers and footers are ignored, whilst more weight is attributed to terms found in regions of particular interest, such as the abstract of an article, its section headings, or its bibliography.

Chapter 4

KEYPHRASE EXTRACTION

This chapter first examines the ways in which the keyphrase features discussed in Chapter 2 have been used in automatic term extraction software. An introduction on what the extraction task implies is given, followed by an account of the many documented efforts to perform this task automatically. Emphasis is placed on the fact that a general uncertainty as to what a keyphrase represents, seems to have thus far hampered widespread community adoption of a clear extraction solution. This in turn motivates an examination of what the current state-of-the-art in the field is, and of the effectiveness of using certain features over others.

The description of a novel keyphrase extractor called KPEX is then given, that draws upon the insight gained from the previous chapters. The purpose for its extracted keyphrases it to best describe the content of the analysed document. A simplistic set of features that stood out as the most profitable for automatic extraction were used in the implementation, particularly the ability to work with the logical structure of the input document, such as that recoverable with PDFX. The primary focus was the utilisation of features solely intrinsic to the input document, in order to thoroughly reassess their capability of discriminating important terms. This relieved the system of the necessity to consult collection statistics or external terminological databases, greatly widening its applicability. An evaluation using an established benchmark yielded top-performing results in comparison to the state-of-the-art (Section 4.3.3), whilst an empirical user study has found the solution very promising for the task of ontology enrichment (Section 4.3.4).

4.1 Background and Related Work

This section will aim to offer insight into the principal methods currently in use for assigning keyphrases to documents, either by extraction or by indexing with controlled terms. Their alignment to one another in terms of the features used, application domain and performance evaluation procedures is also discussed.

Given the established importance of the keyphrase extraction task, an impressive array of potential solutions for their identification and ranking has been documented. 172 of them have been reviewed in writing this dissertation. Whilst abundant, the solutions have greatly varying scopes and objectives, so there is only minimal relative agreement amongst them on performance evaluation datasets and metrics. This fact, plus the unavailability of many implementations for further experimentation, greatly diminish the possibility of their alignment for direct comparison purposes.

4.1.1 Pioneering Research

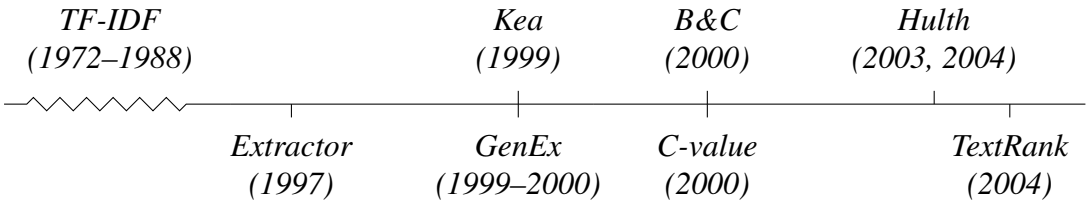


Figure 4.1: Timeline of pioneering research conducted on term weighting and keyphrase extraction until the early 2000's.

[TF-IDF]

Salton and Buckley (1988) provide a very useful review of the insights gained in automatic term weighting methods up until the 1990's. They write from the perspective of information retrieval system design, but the choices of how to weigh terms to foster better retrievability have much in common with feature considerations for descriptive keyphrase extraction as well.

Early suggestions for ranked retrieval of electronic documents were to use *content identifiers* to represent both documents (information sources) and user queries (information requests) and measure relevance as a statistical overlap between the sets of

identifiers. The identifiers could be either words extracted from the documents and queries, or descriptive terms manually chosen by professional indexers to represent them. The perceived advantage of using human indexers was that they would be familiar with both the subject areas under consideration and with the contents of the document collections. Consequently, they would be more knowledgeable as to how queries were likely to be formulated when searching for certain types of documents. In either case, using extracted or assigned terms, both documents and queries could be represented by *term vectors* of the forms:

$$D = (t_0, t_1, ..., t_m)$$

$$Q = (q_0, q_1, ..., q_n)$$

where t_k denotes a term assigned to a document, q_k, a term assigned to a query and m and n represent the number of terms in each vector. In this representation, the two term vectors may differ in size. A more formal representation of the above definitions is obtained by having each term vector include all p possible terms used by the retrieval system, and attributing to each term a value between 0 and 1 to represent its weight w. This weight would signify the term's importance, either in the document or the query:

$$D = (t_0, w_0; t_1, w_1; ...; t_p, w_p)$$

$$Q = (q_0, w_0^{'}; q_1, w_1^{'}; ...; q_p, w_p^{'})$$

Query-to-document similarity could then be computed using the conventional vector product formula:

$$similarity(D, Q) = \sum_{k=1}^{p} w_k w_k^{'}$$

which, by using a vector length normalisation factor yields the well-known cosine similarity formula:

$$cosine\_similarity(Q, D) = \frac{\sum_{k=1}^{p} w_k w_k^{'}}{\sqrt{\sum_{k=1}^{p}(w_k)^2 \cdot \sum_{k=1}^{p}(w_k^{'})^2}}$$

Two key points needed to be considered in designing a text retrieval system that used these definitions for documents, queries and similarity, both of which are also relevant to the task of keyphrase extraction: what the appropriate terms to use were and what mechanism should be used in generating effective weighting factors for those terms.

In many cases, using single-word terms in content representations produced satisfactory retrieval output. However, it became apparent that there was still room for improvement in terms of accurately describing and discriminating documents, and in the way queries were formulated to retrieve them. Various methods thus started being documented for generating more comprehensive representations of content:

- The generation of sets of related terms based on statistical co-occurrences of words across a document collection (Lesk, 1969; van Rijsbergen, 1977; Yu et al., 1983);
- The substitution of words for syntactic constructs such as noun phrases in the term vectors (Klingbiel, 1973; Spärck Jones and Tait, 1984);
- The use of controlled vocabularies, dictionaries, lexicons or thesauri to identify headings for classes of related terms. These headings would then be used as content identifiers instead of the terms themselves (Jones, 1971; Salton, 1971; Fox, 1980);
- The construction of complex knowledge bases (that would come to be known as ontologies) and the use of their conceptual representations of textual content as document identifiers (Spärck Jones, 1983; Croft, 1986).

At the time Salton and Buckley wrote their survey paper, there was little evidence to suggest that more complex entities extracted from texts or vocabularies would yield better retrieval performance. Consequently, efforts were considered better spent in trying to generate effective weighting factors for the individual terms, based on their perceived importance as content descriptors. In considering the measures of Precision and Recall described earlier in Section 2.2, three main features have stood out as efficient term weight generators:

1. Term frequency (TF) – a descriptive factor that signified importance within a document;
2. Inverse document frequency (IDF) (Spärck Jones, 1972) – a discriminating factor that would ensure relative scarcity across the collection so that the size of the returned document set was small enough to be useful;
3. A weight normalisation factor to prevent larger documents from being favoured in retrieval results.

Salton and colleagues had previously described how the product of TF and IDF would

give a reasonable measure of term importance – *TF-IDF* (Salton and Yang, 1973; Salton, 1975; Salton et al., 1975). Additional studies, such as Croft and Harper (1979); Wu and Salton (1981) had also concluded that under well-defined conditions, term relevance could be reduced to an inverse document frequency of the form $log((N-n)/n)$ where N represents the total number of documents in a collection, and n is the number of documents containing a particular term. Having been thus directly related to other theoretically valid probabilistic models of information retrieval, the TF-IDF measure was soon to become the most renowned and most used relevance measure for keyphrase extraction and document retrieval.

[Extractor]

An early report by Turney (1997) provided an empirical evaluation of four keyphrase extraction algorithms: Microsoft Word 97's AutoSummarize feature; a tool outputting most frequent noun phrases, as tagged by Eric Brill's tagger (Brill, 1992); Verity's Search 97's Summarize feature[1]; and *Extractor* – the author's own solution and one of the first supervised learning algorithms for the extraction task. Extractor took 12 parameters into consideration when generating the list of keyphrases for an input text. These included the desired stemming length, position of the first occurrence of a term, and the number of tokens in a term. Term scores were then calculated by multiplying their frequency of occurrence with empirically set boosting factors. These factors varied depending on the terms' first occurrences. One of the testing collections for Extractor comprised 75 articles taken from 5 journals on the topics of cognition, hospitality and chemistry. When a 3:1 training–testing split was used for Extractor, the learned model solution was found to outperform the other three systems with regards to the F_1 measure obtained against lists of author-provided keyphrases.

[GenEx]

As a follow-up from Extractor, Turney (2000) also evaluated a hybrid solution called *GenEx*. The name 'GenEx' came from *'Genitor plus Extractor'* and implied using the *Genitor* genetic algorithm (Whitley et al., 1989) to maximise Extractor's performance on some training data by fine-tuning the 12 parameters previously mentioned. GenEx was compared to the application of the general-purpose C4.5 decision tree induction

[1] Verity Inc. was acquired by HP Autonomy – `http://autonomy.com/`

algorithm (Quinlan, 1993) on the extraction task. A total of 110 different features were considered for the C4.5 implementation before deciding on the final 12 with which tests were conducted – a noteworthy effort to get the best out of the algorithm. Several experiments ultimately showed that the specialised domain knowledge that GenEx had gained in its hybrid learning approach allowed it to outperform C4.5 across all the six datasets used in the evaluation.

[Kea]

Around the same time as Turney, Witten and colleagues (Frank et al., 1999; Witten et al., 1999) developed *Kea*, another solution for keyphrase extraction, using a Naïve Bayes training model (Domingos and Pazzani, 1997). The approach had the advantage of a much shorter training time in comparison to GenEx. It also used only 2 keyphrase features by default, TF-IDF and first occurrence, plus a post-processing step to remove sub-phrases from the output list. Kea was found to perform better if trained on a corpus of documents belonging to the same field. With this observation, the authors considered extending the model to exploit collection-specific knowledge about the likelihood of a particular phrase being a keyphrase. The number of times a candidate phrase occurred as a keyphrase in the training dataset was integrated as an additional feature, for when corpora of the same domain were used for both training and testing. Kea and another implementation of C4.5 were then compared to the complex system GenEx on a subset of Turney's document collections. The comparison showed no statistically significant difference between the three algorithms, supporting the claim that specialised domain knowledge was valuable in an extraction task. Additionally, in Witten et al. (1999), the authors noted that an evaluation against author keyphrases alone might not suffice to assess the usefulness of an extracted set of keyphrases for a particular task. They proposed that evaluating the opinions of multiple judges on the quality of entire keyphrase *sets* would be more insightful.

[B&C]

Closely following the GenEx and Kea experiments was the work of Barker and Cornacchia (2000), who experimented with using noun phrases to limit the set of keyphrase candidates. A noun phrase was chosen based on its length, frequency and the frequency of its head noun. The authors identified noun phrases by a simpler method than the

widely-used Brill part-of-speech tagger mentioned earlier. They opted for looking-up words in two English dictionaries that offered a fairly complete coverage of closed-class words (articles, prepositions, conjunctions, etc.) and possible parts-of-speech for words. Two interesting observations were made. First, this new system, named B&C, was evaluated alongside Extractor at the level of individual keyphrases and also complete keyphrase sets, as Witten suggested. It was found to perform favourably despite not requiring prior training. Second, human evaluators did not seem to consider the quality of sets of keyphrases as a simple function of the quality of individual keyphrases. This fact suggested that neither the sole evaluation of individual keyphrases, nor of sets of keyphrases sufficed for a comprehensive performance measure. Other more intuitive ways of assessing keyphrase quality were needed.

[C/NC-value]

Yet another approach was proposed by Frantzi et al. (2000). The *C/NC-value* method that the authors introduced was domain-independent and required no training, but targeted only the extraction of multi-word terms. The proposed method consisted of two parts. The first part, *C-value* extracted keyphrase candidates based on a part-of-speech filter and then ranked them using a statistical measure of *termhood*. C-value considered a term's length in number tokens and its frequency of occurrence, normalising the scores obtained for nested keyphrases. Given a term a, its C-value was computed as follows:

$$C\text{-}value(a) = \begin{cases} log_2|a| \cdot f(a) & \text{if } a \text{ is not nested} \\ log_2|a| \cdot f(a) - \frac{1}{P(T_a)} \sum_{b \in T_a} f(b) & \text{otherwise} \end{cases}$$

where $|a|$ is the length of a in number of tokens,
$f(i)$ is the frequency of term i in an article corpus,
T_a is the set of candidate terms that contain a,
$P(T_a)$ is the number of these longer candidate terms

This procedure discounted the occurrences of nested shorter terms within longer ones from their overall score. This acted as a beneficial normalisation factor for shorter terms that were usually favoured in frequency-based statistics. For the same reason, the length of a keyphrase in number of tokens was directly linked to its frequency of occurrence. This functioned as a boosting factor for longer terms that occurred much less often than single-word terms.

The second part of Frantzi et al.'s procedure was named *NC-value* and was meant as a post-processing step to C-value, conducting a re-ranking of the output list with respect to context information, to improve performance. It took into account the context of each candidate keyphrase, assigning weights to specific verbs, adjectives and nouns that appeared in its vicinity. The weight factor of a context word w was higher for words that tended to neighbour keyphrases. This factor was computed as

$$weight(w) = \frac{t(w)}{n}$$

where $t(w)$ is the number of candidate keyphrases the word w appears with and n is the total number of candidates considered from the document. The NC-value was then formally defined as

$$NC\text{-}value(a) = 0.8 \cdot C\text{-}value(a) + 0.2 \cdot \sum_{b \in C_a} f_a(b) \cdot weight(b)$$

where C_a represents the set of distinct context words of term a and $f_a(b)$ is the frequency of b as a context word of a.

Even though it did not take *keywords* into account (i.e. single-word keyphrases), C/NC-value proved useful in various settings (Ananiadou et al., 2000; Milios et al., 2003; Zervanou and McNaught, 2004; Zhang et al., 2005; Zervanou, 2010). The fact that it had greater sensitivity to multi-word terms made it a good candidate for terminology extraction from scientific texts, where the language was generally more specialised. The National Centre for Text Mining (NaCTeM)[2] currently hosts an implementation of the C-value algorithm under the name *TerMine*.

[TextRank]

In 2004, an extraction model was proposed by Mihalcea and Tarau (2004) that was different in paradigm to the then learning-oriented state-of-the-art, yet just as promising. The authors introduced a graph-based ranking algorithm for texts called *TextRank*, taking after the famous solution developed for the World Wide Web, *PageRank* (Brin and Page, 1998). The TextRank model treated the document text as a graph, with terms represented as nodes and their co-occurrence within a certain word window as edges. The main idea employed was that of 'voting' or recommendation. Each node

[2]The National Centre for Text Mining (NaCTeM) – `http://www.nactem.ac.uk/`

was considered to cast a vote of importance on all its neighbouring nodes, i.e. on all terms found within a certain word window of it. The PageRank procedure employed the same approach for web pages linking to other pages. The original PageRank score (PR) of a vertex V_i in a directed graph $G = (V, E)$ was computed as follows:

$$PR(V_i) = (1 - d) + d * \sum_{V_j \in In(V_i)} \frac{1}{|Out(V_j)|} PR(V_j)$$

where $In(V_i)$ denotes the set of vertices that point to V_i,
$Out(V_i)$ denotes the set of vertices that V_i points to,
d is a damping factor with the role of integrating the probability of jumping from a given vertex to another random vertex. It was added by Bring and Page to simulate the possibly random behaviour of a web user when accessing web pages and set empirically to 0.85.

The TextRank approach was implemented in four main steps:

1. Identify the text units of interest and add them as vertices in the graph;
2. Identify relations that linked text units and draw the appropriate edges;
3. Iterate the above ranking algorithm until convergence (i.e. until the difference in vertex scores between two consecutive iterations falls below a given threshold);
4. Consider the vertices' final score to be their importance within the document.

For keyphrase extraction, directionality of the graph was not used in TextRank. However, an adaptation of the PageRank algorithm for natural language texts was required, to allow the attribution of different weights to the graph edges, to signify the link strength between two terms, as derived from the document. This strength was given by their co-location frequency within the document, i.e. the number of times the terms were found within an N-word window of each other. The formula integrating edge weights w_{ij} was given as:

$$PR_{weighted}(V_i) = (1 - d) + d * \sum_{V_j \in In(V_i)} \frac{w_{ij}}{\sum_{V_k \in Out(V_j)} w_{jk}} PR(V_j)$$

The TextRank approach for keyphrase extraction was evaluated on a corpus of 500 abstracts with indexer-assigned terms and was found to be quite versatile. Although PageRank formed the basis for the implementation, any other graph-based ranking

algorithm could be substituted in the pipeline, such as HITS (Kleinberg et al., 1999) or Positional Function (Herings et al., 2001). The authors also made an important remark regarding the terms to be considered nodes in the graph, and the relations used to draw edges between them. They stated that it was the application at hand, i.e. the actual intended use for the extracted terms, that should determine these choices.

[Follow-ups]

Several other noteworthy lines of research in keyphrase extraction followed from or were interspersed with the pioneering works just mentioned.

The work done by Hulth (2003, 2004) added more linguistic knowledge in the extraction procedure. Supervised machine learning was used again, but integrating part-of-speech information into the process this time. Experiments were performed with identifying keyphrase candidates as (a) uni-, bi- and trigrams; (b) noun phrases; (c) constructs matching additional, empirically defined part-of-speech patterns. Using noun phrase POS patterns had the advantage of candidates not being restricted to some arbitrary length, as was previously done. Experimentation with the statistical features proposed by Frank et al. (1999) and Turney (2000) for ranking candidates showed a significant performance increase when the additional linguistic knowledge was used. The evaluation was conducted over a collection of document abstracts rather than full texts, but the contribution of the linguistic element was to be acknowledged for full-texts as well by many other efforts over the following years (Nguyen and Kan, 2007; Ercan and Cicekli, 2007; Yang et al., 2009).

In another study, Turney (2003) noted that term candidate selection decisions were not independent for humans and suggested that, during the extraction process, prior keyphrase selections should influence the remaining selection decisions. He proposed to model the coherence of a set of keyphrases as a whole using Pointwise Mutual Information between a keyphrase candidate and all previously selected keyphrases.

Medelyan (2009) presented and evaluated an extension to the Kea algorithm, called Maui, designed for topic indexing. Kea's baseline procedure was augmented to include the mapping of keyphrase candidates to external knowledge sources, such as controlled vocabularies and Wikipedia. This had the effect of Maui's output favouring terms likely to be categories or subject descriptors under which the analysed documents could be catalogued. A variety of features were blended into Maui's learning model:

- Length in number of tokens;
- TF, IDF and TF-IDF;
- First occurrence, last occurrence and spread (the difference between last and first);
- Domain keyphraseness – the frequency within a manually-constructed topic set;
- Wikipedia keyphraseness – the probability of occurrence as an anchor text in Wikipedia articles;
- Inverse Wikipedia frequency – the number of incoming links to the most likely Wikipedia article for the term;
- Node degree in a controlled vocabulary – number of specified semantic links;
- Wikipedia semantic relatedness, with respect to the other keyphrase candidates;
- Generality of terms mapped to Wikipedia articles – a normalised distance between the article's Wikipedia category and the root of the category tree.

4.1.2 Surveys of Features and Approaches

[Jacquemin and Bourigault (2003)]

Jacquemin and Bourigault provided a survey of term extraction and automatic indexing techniques, but more from a natural language processing perspective. The survey regarded the representation, acquisition and recognition of terms, rather than their ranking with respect to some relevance factor. The work went through the basic linguistic characteristics of terms and discussed different documented approaches for formalising definitions of terms, their properties, detecting terms in text and the relations between them. A mixture of syntactic and statistical methods were analysed, with the three main identified lines for further research being:

- The construction of large-scale resources of terminology management;
- The combination of textual and structural information for the recognition of rich term contexts;
- Semantic tagging and the acquisition of semantic relationships from corpora.

[Krauthammer and Nenadić (2004)]

Krauthammer and Nenadić presented a survey of efforts concerned with terminology extraction for the biomedical domain, but also covered the issue of term disambiguation. Terminology extraction, largely synonymous to *term recognition*, is best mapped to the keyphrase candidate generation stage of a keyphrase extractor, as its main concern is the extraction of likely domain-specific concepts, rather than their ranking. The authors discussed state-of-the-art approaches in *term identification* – a 3-part process which, apart from term recognition, also implied *term classification* into high-level taxonomical categories and *term mapping* to uniquely identified knowledge base entries. The surveyed methods for term recognition were largely dependent on dictionaries or external knowledge bases. The main reason for this affinity was that term feature identification in the general sense, i.e. for general-purpose keyphrase extraction, was more difficult than the identification of features specific to individual classes of terms. With regards to term recognition, a mixture of features was found as being employed by state-of-the-art systems:

- orthographic – capital letters, digits, Greek letters, etc;
- morphological – various affixes or POS tags;
- syntactic – through shallow sentence parsing;
- statistical – through ranking mechanisms to promote term candidates into terms.

The variation and inconsistencies in surface expressions of terms, as well as their ambiguity were reported as the adamant hindering factors of achieving better performance in overall term identification.

[Kim and Kan (2009)]

A next relevant survey came five years later, from Kim and Kan (2009), after a good amount of additional experimentation had been done by the community towards alternative methods for keyphrase extraction. The authors targeted the extraction of keyphrases from scientific articles, revisiting the tasks of candidate selection and feature engineering in an attempt to identify some of the field's current limitations. The task of keyphrase extraction was seen as still having much room for improvement, primarily due to three core factors:

- An overall inability to deal with term variation adequately. This affected systems' performance due to limited coverage of gold-standard keyphrases: only few semantically distinct concepts were output.
- The lack of a detailed investigation of the nature and variation of keyphrases extracted by humans, to foster a better understanding of the task.
- The difficulty involved in attempting a direct, comprehensive comparison of existing approaches, due to their minimal intersection in terms of scope and benchmarking procedures.

Kim and Kan analysed results against author- and reader-assigned keyphrases for 250 papers of 4 different categories from the ACM digital library[3], making several useful observations. With respect to keyphrase variation, they touched upon how prepositional phrases such as *'quality of service'* could be translated into *'service quality'* with no loss in semantics. They also observed how simple conjunctions such as *'search and rescue'* could be considered valid keyphrases by humans but would not be recognised by typical part-of-speech patterns for keyphrase candidates. Abbreviations (e.g. *'inverse document frequency (IDF)'*) and possessive forms (e.g. *'Bayes' theorem'*) were also mentioned as variations to consider. The authors ultimately listed 25 features of keyphrases that they found useful in extracting keyphrases[4]. Among the features were also structural ones that singled out candidates occurring in the abstract of the document, in the introduction, conclusion or section headings.

[Hofmann et al. (2009, 2010)]

Similarly to Kim and Kan, Hofmann and colleagues also hypothesised that document structure provides useful cues for keyphrase extraction because of established typesetting conventions that direct readers to important parts within the document. In Hofmann et al. (2010), the authors systematically compared features for keyphrase extraction on a large corpus of scientific journal articles. They conducted their analysis on over 14k articles published by Elsevier in the domain of Food Informatics and Computer Science between 1995 and 2005. A useful aspect of the collection was that it had rich, manually-curated XML markup of each article made available by the publisher. This allowed the authors to inspect different kinds of keyphrase features at the

[3]The ACM digital library – `http://dl.acm.org/`
[4]All 25 features were recalled in Chapter 2 of this work.

document structure level (e.g. title, abstract, sections) and collection structure level (e.g. journal name, issue or publication date). The main finding was that the distinction between the different logical sections of a document, as well as between parts of a collection (e.g. articles belonging to different journals) contributed useful information in extracting keyphrases.

[SemEval-2010 Task #5 – Kim et al. (2010)]

One of the most insightful comparative studies of keyphrase extraction approaches was carried out by Kim et al. (2010). Rather than re-attempt to draw conclusions based on previously documented research, the authors gathered an up-to-date account of the state-of-the-art by proposing a keyphrase extraction competition as a SemEval[5] evaluation exercise for the ACL SigLex 2010 Event[6]. A benchmarking procedure was put in place, with a scientific article collection, gold-standard outputs and performance metrics clearly defined. Afterwards, contributors were invited to submit their systems' outputs over the collection, for comparison. Competition entries were then evaluated by matching their extracted keyphrases against the ones assigned both by the original authors as well as independent readers. The call was well-received, with a total of 19 systems participating in the end – an unprecedented level of convergence of keyphrase extraction efforts on a common evaluation framework. The outcome of the exercise provided valuable insight into the relative performances of different combinations of keyphrase features, and has functioned as a benchmark for future efforts ever since.

The SemEval dataset and scoring procedure is also used in this dissertation as one of the two methods of evaluating the performance of KPEX – the keyphrase extractor described in Section 4.2. Since the task was to specifically extract keyphrases from scientific articles, the evaluation provided a good appreciation of the system's capabilities. The performance achieved by KPEX using the SemEval procedure is presented at length in Section 4.3.1, after a more detailed account of the 2010 exercise, its participating systems and their performances. Other related work documented since the competition, that has used the same dataset and evaluation procedure is also mentioned.

[5]SemEval-2 (2010) website – `http://semeval2.fbk.eu/`

[6]SigLex – The Special Interest Group on the Lexicon of the ACL – `http://www.clres.com/siglex.html`

4.1.3 Usage of Document Logical Structure

All the surveys described in the previous section showed a level of agreement on the benefits of utilising document structure information in the extraction task. In addition, there are other individual studies that also agree, coming from diverse subject areas. This subsection outlines them.

[Shah et al. (2003)]

Shah and colleagues asked the very pragmatic question *'Where are the keywords?'* with respect to their localisation in sections of scientific articles. The incentive was that in many cases it had proved useful to quantify and qualify the information in a full-text article before attempting a more involved information extraction process over it. In a study of biomedical literature, the authors concluded that the keyphrase content from different sections of a standard article (Abstract, Introduction, Methods, Results and Discussion (AIMRD)) was very heterogeneous. They suggested that, in the text mining of full-text articles, different strategies should be employed for different logical sections, depending on the specific goal in mind for the extracted keyphrases.

[Nguyen and Kan (2007)]

Nguyen and Kan (2007) used a maximum entropy classifier to infer 14 generic section headings of scientific publications, similar to Shah et al.'s AIMRD model. The authors integrated a term's section occurrence vector as a novel feature in their Naïve Bayes learning solution, on top of the baseline features of Kea, POS, suffix, and abbreviation information. The obtained improvements over the performance of Kea were found to be statistically significant.

[Esposito et al. (2008); Ferilli et al. (2009)]

Using machine learning, Esposito and colleagues presented an ample framework for digital document processing, that went from layout analysis to metadata extraction. Their system, *DOMINUS*, centred around a machine learning server with the ability to choose among different learning strategies, the one that would work best for a particular document type. This multi-strategy technique was initiated with three phases,

going through initial geometric layout analysis, document classification by type (e.g. newspaper, scientific article, email, etc.), followed by logical component inference (e.g. signature, author, footnote, etc.). After the document's logical structure was in a machine processable format, the document was indexed with respect to its logical components to foster more effective content-based retrieval. In Ferilli et al. (2009), DOMINUS was extended to include two lexical resources, WordNet and WordNet Domains (Miller et al., 1990; Magnini and Cavaglia, 2000), and a density function defined over them, that transformed a document into a weighted map of *synsets* (synonymous term sets) which described it conceptually.

[The SemEval-2010 Task #5 contestants]

Out of the 19 systems that participated in the SemEval-2010 challenge, 12 made use of the input document's logical structure in one form or another in their extraction pipeline, including 6 of the 8 best performing systems according to the official results[7]. Some of the features contestants experimented with were:

- Input cleaning by logical structure - through the removal of superfluous elements such as headers and footers, affiliations, emails, etc;
- Term occurrence within logical regions of interest (e.g. title, abstract, introduction, conclusion, bibliography);
- Term frequency within logical regions of interest;
- Term typeface in the original publication (e.g. bold, italic, etc.).

The methods and features used by the top-performing participants are listed in Table 4.1. The aim of the table is to emphasise the variety of features that were considered. This work will not proceed to detail all the features mentioned therein, but the reader is welcome to refer to the descriptions of each participating system for more information. The SemEval competition setup and results are detailed in Section 4.3.1.

[7]For the ICL system, there was no publication in the SemEval-2 workshop to clearly specify the features used in the competition, but the system's authors have recommended Wang and Li (2011) as a description of the method used in SemEval-2010, which includes the use of article title information (Li, 2013 – private communication).

Table 4.1: Features used by the top-performing systems of the SemEval-2010 keyphrase extraction challenge and their scopes (*Phrase, Document, Collection, External*). Features related to the input document's logical structure are marked in gray. Delineated features in *italics* were considered, but left out from the setups that yielded the official results. The system by You et al. (2012), that also used the SemEval procedure for evaluation is represented as well.

System / (Method)	Feature	Scope			
		Phr.	**Doc.**	**Col.**	**Ext.**
HUMB [8] (Bagged Decision Trees)	Length	X			
	First occurrence		X		
	Phraseness (GDC)		X		
	In Titl / Abs / H / Intro / Con / Bib		X		
	Informativeness (TF-IDF)		X	X	
	Keywordness (global TF)			X	
	Occurrence in GRISP [9]				X
	Wikipedia keyphraseness				X
	HAL[10] stats on co-usage				X
	NP filtering (POS)	X			
	Language model deviation		X		
	Term variants				X
	Global (HAL) keywordness				X
	Wikipedia term relatedness				X
You et al. (2012) (Rule-based)	First occurrence		X		
	In Titl / Abs / first paragraph		X		
	TF	X			
	Length	X			
	Contains CoreWord	X			
	IDF difference (for overlap removal)			X	
WINGNUS [11] (Naïve Bayes)	Input cleaning by logical structure		X		
	Body paragraphs abridged to 1st sent.		X		
	Length	X			
	TF	X			
	TF of substrings	X			

Continued on next page

[8] The HUMB keyphrase extraction system – Lopez and Romary (2010)
[9] The GRISP terminological database – Lopez et al. (2010)
[10] The HAL Open Archives system – Baruch (2007)
[11] The WINGNUS keyphrase extraction system – Nguyen and Luong (2010)

Table 4.1 – continued from previous page

System / (Method)	Feature	Scope			
		Phr.	Doc.	Col.	Ext.
WINGNUS *(continued)*	First occurrence		X		
	TF-IDF		X	X	
	Last occurrence		X		
	Typeface		X		
	In Titl / Abs / H / Intro / Res / Con		X		
	TF in Abs / H / Intro / Res / Con		X		
	InTitle of any DBLP doc				X
KP-Miner [12] (Rule-based)	No punctuation	X			
	No stopwords	X			
	Min. no. of occurrences		X		
	First occurrence		X		
	TF-IDF with boosting factor		X	X	
SZTERGAK [13] (Naïve Bayes)	Length	X			
	POS patterns	X			
	Suffix	X			
	First occurrence		X		
	Acronymity		X		
	Generalised PMI		X		
	TF-IDF		X	X	
	SF-ISF (section-wise TF-IDF)		X	X	
	Wikipedia keyphraseness [14]				X
	Wikipedia synonymy				X
SEERLAB [15] (Random Forest)	Length	X			
	Acronymity		X		
	TF		X		
	TF in Titl / Abs / H / Intro / Res / Con		X		
	IDF (by DBLP)				X
	TF-IDF		X		X
	In Title of at least 3 DBLP docs				X

Continued on next page

[12] The KP-Miner keyphrase extraction system – El-Beltagy and Rafea (2010)

[13] The SZTERGAK keyphrase extraction system – Berend and Farkas (2010)

[14] Another keyphraseness feature reported in Berend and Farkas (2010), inspecting the appearance of a term as an author-assigned keyphrase, was ultimately left out in deriving the official results.

[15] The SEERLAB keyphrase extraction system – Treeratpituk et al. (2010)

Table 4.1 – continued from previous page

System / (Method)	Feature	Scope			
		Phr.	Doc.	Col.	Ext.
KX_FBK [16] (Rule-based)	Length	X			
	POS patterns	X			
	TF		X		
	First occurrence		X		
	Shorter concept subsumption		X		
	Longer concept boosting		X		
	Acronym expansion		X		
	IDF			X	
	Corpus frequency			X	
DERIUNLP [17] (Rule-based)	POS patterns	X			
	Length	X			
	Rank boosting by logical section		X		
	Acronym expansion		X		
	TF in collection as skill type			X	
	TF-IDF			X	
	Google hits cutoff				X

4.1.4 The State-of-the-art

From the point of view of feature comparison and visibility within the field, the SemEval-2010 challenge stands out as reference material, and the competition's top-performing systems can be seen as the accountable, benchmarked state-of-the-art. Since the SemEval evaluation procedure was made available, many other systems have been published that target the extraction of keyphrases from text. Unfortunately, very few of them (only six, according to this dissertation's literature review) have reused the procedure's corpus and metrics for evaluation. Out of the six, only one, You et al. (2012), performs favourably in the system ranking, reaching 2nd place. The features of this solution were also included in Table 4.1. The implementation concentrated on two aspects: pre-emptively reducing the candidate set of keyphrases to those that contain *core words* (the most frequent, different-stem words from the document), and subphrase elimination from the output list, using IDF for selecting the proper granularity.

[16] The KP_FBK keyphrase extraction system – Pianta and Tonelli (2010)
[17] The DERIUNLP keyphrase extraction system – Bordea and Buitelaar (2010)

The forefront of development in the area of keyphrase extraction seems to rely progressively more on external knowledge sources to get insight on concept semantics, or to retrieve some declared measure of significance for particular terms. Wikipedia is amongst the top choices for deriving such knowledge (e.g. Lopez and Romary (2010); Berend and Farkas (2010); Liu et al. (2012); Joorabchi and Mahdi (2013); Lei et al. (2013)), probably due to its coverage of many disciplines and the ease of access provided by open-source software such as Wikipedia Miner (Milne and Witten, 2013).

Best performances are achieved when one or several of these external sources are used in conjunction with phrasal, linguistic, structural and document collection metrics. Only well-crafted combinations of these seem to manage to approximate a human annotator's decision making process when selecting keyphrases for documents. It is interesting to note, however, that although a human annotator might be an expert in the field of the article being analysed, he/she has only limited capabilities of deriving the information that keyphrase extraction systems rely on, such as TF-IDF or global keyphraseness. This brings up the question of how much prior knowledge humans actually use when extracting keyphrases, and whether automated tools use the simpler document-scoped features to their full potential. In what follows, several features bound only to the document scope are carefully combined into a rule-based extraction system and proven to be very effective.

4.2 Proposed Solution: KPEX

Despite a range of existing lines of research, the conducted literature review has highlighted only limited agreement on the efficiency of certain keyphrase features and only modest integration of approaches into real-world information systems. This apparent lack of a viable solution prompted a reassessment of the true impact of some fundamental features that stand out across the literature. The focus has been on document-scoped features derivable without the use of specialised domain knowledge. These covered frequency of occurrence, information content (i.e. descriptiveness or specificity), and the implicit importance attributed through the article's rhetorical structure. A new keyphrase extractor called KPEX has been implemented to combine these features robustly and test their ability to single out salient terms.

The way in which structural information is used in KPEX is deemed by the author to

be the key to its apparent success, as it yields many advantages over the bulk of related work:

- The PDFX system described in Chapter 3 is first employed to recover the document structure, giving KPEX the ability to work with scientific articles in their prevailing format, the PDF, thus vastly expanding its application domain.
- Document structure identification fosters the removal of a substantial amount of superfluous text prior to processing, such as headers, footers and tabular data, making computed statistics more accurate.
- Relying on structural information in the post-processing stage to re-weigh the initial list of keyphrases, boosts KPEX's performance considerably. This is because information from ten different logical regions of the article is used to better approximate the importance that humans are likely to give to certain terms.

KPEX does also support post-ranking with respect to external knowledge sources (as will be mentioned), but this feature is only briefly described in the following sections, as the central objects of study are features intrinsic to the input document.

The remainder of this chapter describes and evaluates KPEX. The next section goes through the design principles of the algorithm, and details each of the four processing stages it employs in extracting keyphrases from articles. Section 4.3 presents the experiments conducted to evaluate KPEX's effectiveness when matched against state-of-the-art systems, as well as when evaluated by human experts on its utility within a real-world scientific workflow.

4.2.1 Design Principles

The extractor's design is modular, with the desired keyphrase features being parameterisable to a large extent to allow their easy management from the command-line. This aids in the analysis of the features' impact on the identification of high-quality keyphrases. The features utilised by KPEX are given in Table 4.2. The tasks undertaken in the processing pipeline can be split into four functional groups, each of which will be explained in more detail in subsequent sections.

1. **Input preparation** – reading, normalisation and sanitisation of the input to generate the reference text for analysis. Depending on the input file format, this

stage may include PDF analysis and XML parsing.

2. **Text analysis and transformation** - several processing steps are conducted on the reference text and on the more structured data derived from it to generate statistics. These include part-of-speech tagging, the generation of a keyphrase candidate list, candidate filtering and the merging of different variations of the same term.

3. **Term weighting** – tasks of modelling and computation. The application of a variation of the term weighing algorithm C-value (Frantzi et al., 2000) over the compiled keyphrase candidate list.

4. **Post-processing and output** – an optional alteration of the initial term ranking to better fit a specific purpose, by considering logical region weights or re-ranking with respect to an external knowledge source. Term overlap removal is also carried out at this stage.

Table 4.2: Features used by KPEX and their scope. Features related to the input document's logical structure are marked in gray. The logical regions KPEX can distinguish are *Title, Abstract, H_1, H_2, H_3, Introduction, Conclusion, Body, Caption, Bibliography*.

Feature	**Scope**			
	Phr	**Doc**	**Col**	**Ext**
No punctuation	X			
No stopwords	X			
POS patterns	X			
Abbreviation identification		X		
Term variant conflation		X		
Modified C-value	X	X		
Optional Features				
Input cleaning by logical structure		X		
In Titl / Abs / H / Intro / Con / Body / Cap / Bib		X		
Rank aggregation with an external list				X

4.2.2 Input Preparation

As outlined in earlier chapters, the majority of scientific articles are available solely as PDF documents not designed for automated content extraction by machines. Since plain text is ultimately required for extraction tools, the common solution thus far

has been to preprocess the PDFs with Poppler's `pdftotext` tool (mentioned in Table 3.2 of Section 3.1.2), as it is freely available and does not require training (Kim et al., 2010; Abulaish and Anwar, 2012). This approach sacrifices logical structure information, however, and different article components are often intertwined. KPEX was designed to work in conjunction with PDFX to overcome this issue. If the input is a PDF file, PDFX is first run and the resulting XML is parsed prior to commencing the keyphrase extraction process. When considering that processing is done on camera-ready publications, XML parsing offers the convenient ability to ignore intruding elements responsible for many false-positive keyphrase hits. Common problematic elements are headers, footers, tables of data and author information. The parsing conducted for keyphrase extraction purposes extracts only the following 10 logical regions: `title`, `abstract`, `h1`, `h2`, `h3`, `introduction`, `conclusion`, `body`, `caption` and `bibliography`. After this step, the extraction pipeline proceeds in the same manner as for any plain text input, with the added difference that the keyphrase list obtained from XML-parsed input may be re-ranked with respect to set weights for the different logical regions. This procedure is detailed in Section 4.2.5 on post-processing.

The input text is first read in as a Unicode string, considering character encoding issues. One common problem with text analysis software is the assumption made regarding the encoding of the input text, as it often leads either to certain texts being considered invalid input or to illegible results. Input files may be encoded using many schemes, such as `ASCII`, `Latin-1`, `UTF-8` or `UTF-16`. Without specifically asking the user to provide character encoding information, tools are left guessing and often incorrectly assume one encoding – the default of their programming environment. In KPEX, if the encoding of the input file has not been declaratively specified, the system will make the assumption that the input text is either `UTF-8`- or `Latin1`-encoded. The order of the assumed encodings matters, because it is quite possible to incorrectly interpret an `UTF-8`-encoded string as a sequence of `Latin-1` characters but not vice versa. Consequently, the first attempt assumes an `UTF-8` encoding, `Latin-1` being the fallback in case of a decoding error.

All Unicode symbols and whitespace read in are also normalised both by canonical equivalence and compatibility. As has been described in the chapter on PDFX (Section 3.3.3), the way in which special characters are represented may differ across formats and publishers or even within the same article. This leads to many inconsistencies

that prevent accurate statistics from being drawn. Decomposing and then recomposing Unicode strings can help overcome this problem, as it yields an uniform representation of these different variants. For example, by canonical equivalence, the character 'Å' (U+00C5 – 'LATIN CAPITAL LETTER A WITH RING ABOVE') will be acknowledged as equivalent to 'Å' (U+212B – 'ANGSTROM SIGN'), whilst compatibility will consider 'ff' (U+0066 U+0066 – two consecutive Latin 'f' characters), to be the same as *'ff'* (U+FB00 – the common ligature character).

As a last step, in case document structure information is not used or unavailable, full-stops are inserted before multiple consecutive end-of-line characters to avoid candidates being considered across logical regions. Neighbouring elements such as a section heading and the following paragraph are not normally separated by punctuation and may thus yield false positives if not distinctly separated. Additionally, possessive markers are removed as these were found to be often used inconsistently throughout articles, e.g. '*Bayes' theorem*' vs. '*Bayes theorem*' vs. '*Bayes's theorem*'.

4.2.3 Text Analysis and Transformation

The Keyphrase Grammar

At this stage, the sanitised reference text is passed through the default-trained TreeTagger part-of-speech tagger (Schmid, 1994), and a regular expression keyphrase grammar is matched against the tagged output to identify keyphrase candidates. The grammar was constructed from the classical POS tag sequences used also in previous works (Turney, 1997; Frantzi et al., 2000; Hulth, 2004), but was extended based on observations made regarding the structure of author-provided keyphrases. For reference, Table 4.3 lists the POS tags used in this chapter, and their respective meanings.

Traditionally, the typical pattern for content-bearing terms, in its simplified form, was defined as:

$$(N|J) * (N|VG)$$

When the head of the construct is a noun, the above pattern would constitute a noun phrase. Noun phrases have proven to be very efficient in pin-pointing content-bearing terms. In an earlier analysis of controlled technical terms from five different domains, Justeson and Katz (1995) remarked that 92.5–99% of all terms were noun phrases. Moreover, 97% of these consisted of just nouns and adjectives only, with nearly all

Table 4.3: Part-of-speech tags used in this chapter, and their respective meanings.

Tag	Description	Example
JJ	adjective	*few*
JJR	adjective, comparative	*fewer*
JJS	adjective, superlative	*fewest*
NN	noun	*keyphrase*
NNS	plural noun	*keyphrases*
NP	proper noun	*KPEX*
NPS	plural proper noun	*Windows*
VB	base form verb	*extract*
VG	gerund verb	*extracting*
VN	part participle verb	*extracted*
RB	adverb	*efficiently*
RBR	adverb, comparative	*better*
RBS	adverb, superlative	*best*
CC	coordinating conjunction	*and, or, &*
CD	cardinal number	*2, three*
IN	preposition or subordinating conjunction	*in, of, like*
DT	determiner	*the*

Tag Set	Description
J	JJ, JJR, JJS
N	NN, NNS, NP, NPS
V	VG, VN
R	RB, RBR, RBS

being formed of nouns, adjectives and the preposition '*of*'.

With global publication rates having risen considerably in recent years, however, the general academic vocabulary is also likely to have grown in size and complexity. This justified a re-examination of the POS structure of keyphrases currently in use. For KPEX, the traditional keyphrase grammar has been extended following an evaluation of the part-of-speech tags commonly assigned nowadays to author-provided keyphrases. The proceedings of the 2010 Federated Logic Conference (FLoC)[18] were used in this evaluation. The demographics of the FLoC dataset are given in Table 4.4. The collection consists of 681 papers and 1611 unique keyphrases, manually chosen by their original authors.

POS taggers are known not to be 100% accurate, and, as was highlighted by this analysis, keyphrases of scientific literature proved more difficult to tag confidently than

[18]The Federated Logic Conference – `http://www.floc-conference.org/`

Table 4.4: Statistics for the author-provided keyphrases of the FLoC-2010 accepted papers.

Statistic		Value
Articles		681
Articles with keyphrases		630
Total keyphrases		2290
Distinct keyphrases		1611
Keyphrases in original article		1139 (70.70%)
Keyphrases anywhere in collection		1398 (86.77%)
Keyphrases assigned to each article	1 kp	0
	2 kps	92 (14.60%)
	3 kps	195 (30.95%)
	4 kps	194 (30.80%)
	5 kps	149 (23.65%)
Average keyphrases per article		3.36
Lengths of keyphrases	1 token	404 (25.19%)
	2 tokens	821 (51.18%)
	3 tokens	277 (17.27%)
	4 tokens	74 (4.61%)
	5+ tokens	28 (1.75%)

other terms. Chapter 2 on the properties of keyphrases has highlighted the fact that true keyphrases are unambiguous to their intended audience. With respect to keyphrases of scholarly articles however, this audience is the scientific community, which has developed complex and very specialised vocabularies within different fields of study. When scientists choose keyphrases, they are likely to make their choices based on some judgement of the terms' discriminating qualities, derived through experience and knowledge of the field. This may be problematic for pre-trained, general-purpose taggers, because this discriminating aspect translates into a degree of uncertainty in tagging the terms, since they rarely occur in general language. The corpus that was used to train the default version of TreeTagger for English is the Penn Treebank (Marcus et al., 1993) – a collection intended to cover only standard language use. Retraining the tagger was infeasible, because it would require substantial human effort to manually annotate corpora of documents with POS tags. Without retraining, the consistency of automatic tagging was therefore expected to be lower for keyphrases than for other terms, but, as will be shown, it was possible to alleviate this issue to a large extent.

A tagging confidence measure over keyphrases was derived for each distinct POS sequence assigned by TreeTagger to author-provided keyphrases, in order to assess any

difficulty the tool might have in unambiguously tagging them. Each keyphrase contributed a confidence measure for the most common POS sequence with which it was tagged. This was computed as a percentage between the number of times the keyphrase was tagged with that most common sequence, and the keyphrase's total number of occurrences as a stand-alone term (i.e. when it was not part of a longer term). For example, the author keyphrase '*predicate abstraction*' was tagged with '`JJ NN`' 55 times, occurring 61 times in the corpus as a stand-alone term. This contributed a 90.1% tagging confidence towards the tag sequence `JJ NN`'s overall measure.

Since FLoC authors were not restricted with respect to the keyphrases they could choose at submission time, only 70.7% of the 1611 distinct keyphrases were found to occur in their original articles, while 86.77% occurred in at least one article of the collection. For each author keyphrase, sentences from across the entire collection that contained it were extracted to retain the necessary contextual information for POS tagging. The sentences were then tagged using TreeTagger and the frequencies of all distinct tag sequences for each keyphrase were computed, e.g.

```
predicate/JJ  abstraction/NN, 55
predicate/VB  abstraction/NN,  3
Predicate/NP  Abstraction/NP,  2
predicate/NN  abstraction/NN,  1
```

The confidence measures of the most common POS sequences of each keyphrase were then aggregated and averaged to yield the overall measure. Statistics for the sequences that appeared more than twice in the corpus are given in Table 4.5 (first column). The overall measures for each sequence in this case are not very encouraging. The majority of tagging inconsistencies occur between different noun forms (e.g. regular vs. proper noun) or between nouns and adjectives (as in the '*predicate abstraction*' example above). These, however, should not be problematic for a keyphrase grammar, when considering the following:

1. Within keyphrases, there should be no restriction regarding the possible forms of the same base part-of-speech could have – where one form is allowed (e.g. singular noun), the others should also be (e.g. plural or proper noun);

2. Adjectives and nouns are both central to a keyphrase grammar and they can also be allowed to occur interchangeably in most cases, without the risk of obscuring meaning or yielding false positives. In the previous example, the 2+1 times in

which the word '*predicate*' from '*predicate abstraction*' was tagged as a noun and not an adjective, do not affect the probability of extracting the term as a keyphrase candidate, if the grammar is $(N|J) * N$. The only positions in which adjectives cannot replace nouns in an English keyphrase grammar, are the last tag of the sequence and any noun introduced by the preposition '*of*'.

The above two causes of tagging inconsistencies were thus considered as allowed, in order to highlight any remaining, less frequent cases. The statistics for sequences merged in this way are also given in Table 4.5. Different forms of the same base part-of-speech were first aggregated into tag sets, and the resulting sequences were then

Table 4.5: Statistics for the most common part-of-speech tags (as output by TreeTagger) for author-provided keyphrases of FLoC-2010. Columns show tag sequences, coverage (Cov.) and confidence (Conf.) for default tag sequences, tag sets and conflated tag sets, respectively. Tags are defined as in Table 4.3. Tag sequences with less than 3 occurrences in the corpus are not represented.

Default Tags			**Tag Sets**			**Tag Sets + J/N Conflation**		
Sequence	**Cov.**	**Conf.**	**Sequence**	**Cov.**	**Conf.**	**Sequence**	**Cov.**	**Conf.**
JJ VG	0.3	90.0	N	30.2	100	N	30.2	100
NN NNS	6.4	88.8	N IN N	0.5	100	J/N J/N J/N N	0.9	100
JJ NN NNS	1.1	86.4	J N N N	0.3	100	N IN N	0.5	100
JJ NNS	8.2	84.1	N CC N	0.2	100	N IN J/N N	0.3	100
NN NN	16.0	83.4	N J N	0.5	90.4	J/N CC N	0.2	100
JJ NN	17.0	83.3	J N	25.1	86.0	J/N CC J/N N	0.2	100
NN	22.9	79.3	N N	33.6	84.0	J/N J/N CC N	0.2	100
NN JJ NN	0.2	77.7	J N N	6.6	79.2	J/N N IN N	0.2	100
NNS	5.4	77.7	R J N	0.3	79.0	J/N N	50.8	98.2
JJ JJ NNS	0.3	77.6	N V N	0.3	77.0	J/N J/N N	12.2	96.1
VG NNS	0.4	76.6	V N	2.7	75.4	J/N V N	0.3	77.0
JJ NN NN	4.8	74.2	N N N	6.7	73.0	V J/N N	0.7	76.7
NN NN NN	1.3	73.3	J J N	1.4	70.1	V N	2.7	75.4
NN NN NNS	0.9	72.9	V N N	0.6	70.1	R J/N N	0.4	73.2
VN NN	1.1	70.7	J VG	0.5	69.8	J/N VG	0.5	69.8
JJ JJ NN	1.0	69.5	V J N	0.2	43.3			
NP NN NN	1.1	60.4						
NP NN	4.0	58.3						
NP NP NP	4.4	44.2						
NP NNS	1.8	43.9						
NP	18.5	42.1						
NP NP	14.4	35.6						
JJ NP	1.6	31.4						
Keyphrase coverage achieved with >95% tagging confidence								
0.0%			31.3%			96.2%		

further conflated by adjective (J) and noun (N) in all positions except the last one or prior to the preposition '*of*' (IN). The table shows how the majority of all occurrences of author-provided keyphrases throughout the corpus (96.2%), are caught with >95% confidence by TreeTagger, from the point of view of the keyphrase grammar. Moreover, it highlights how this good coverage is achievable with a simplistic grammar.

To test the hypothesis that keyphrases are harder to tag confidently than non-keyphrases, all other terms in the corpus tagged with the sequences in Table 4.5 were also extracted and their tagging confidence was computed. The aim was to inspect how this confidence would compare to that achieved over keyphrases. Figure 4.2a shows the results obtained over the standard tag set. Non-keyphrases are visibly easier to tag confidently by TreeTagger, having an average confidence margin of 25.2% over author-provided keyphrases for the standard tag set. However, as mentioned, when constructing a keyphrase grammar, the different forms of the same part-of-speech are allowed to occur interchangeably, as are nouns and adjectives themselves, at certain positions in the POS sequence. After this conflation (Figure 4.2b), the margin in tagging confidence drops considerably to just 7.9%. This suggests that the issue of specialised language use and of POS taggers not being 100% accurate is alleviated to a large extent through how tags are allowed to co-occur in a keyphrase grammar.

With this knowledge, KPEX's grammar extends the patterns used by previous works to also cover the most common POS sequences found for author-provided keyphrases in this experiment. With a few additions such as the possibility of intervening numbers or determiners, the grammar, in its simplified form, can be written as:

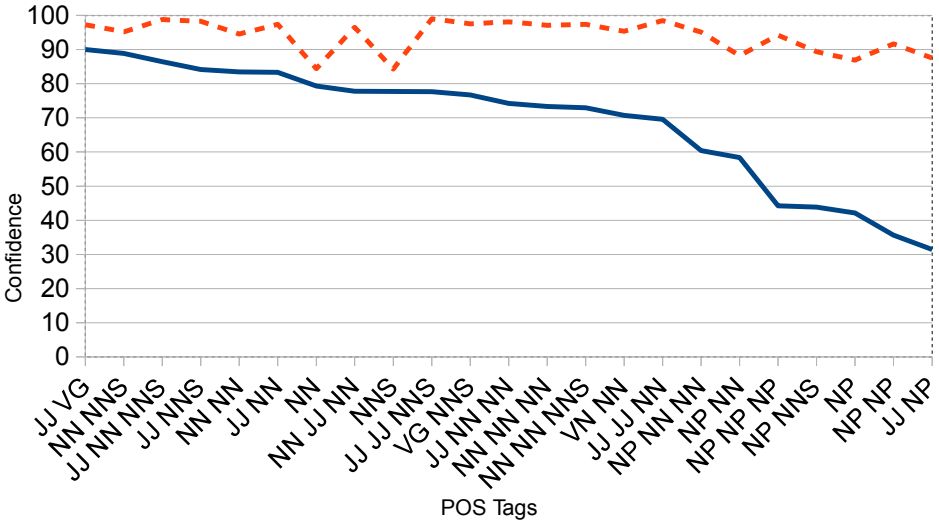
$$(VN|R|CD)?((J|N|VG|VN|R) + (IN|DT|CC|CD)?) * (N|VG)+$$

Candidate Generation

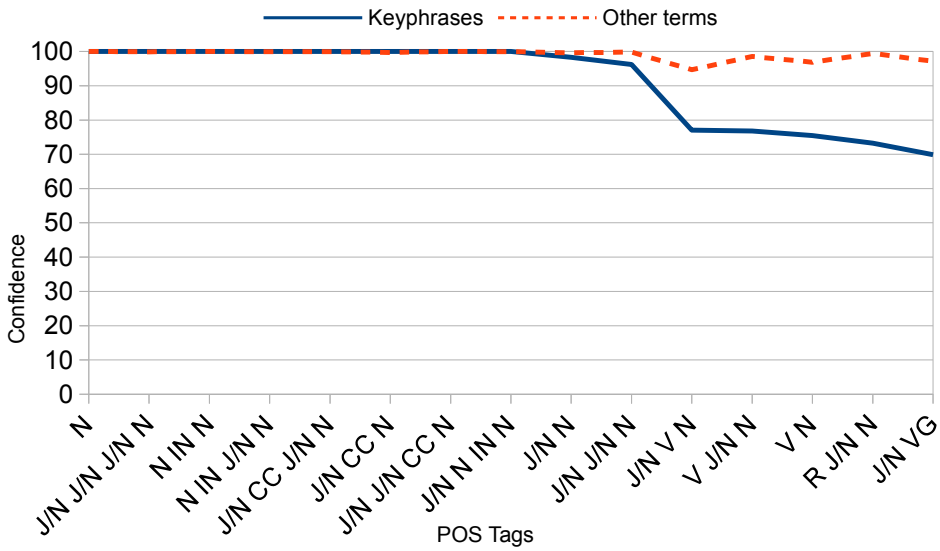
Using TreeTagger's output and the keyphrase grammar, keyphrase candidates are extracted next. All overlapping matches of the grammar are considered at this point. The discounting of nested candidates is conducted further down the pipeline, once the frequencies of all terms have been computed.

Acronyms and abbreviations are also detected here, as they are a prime source of term variants declaratively specified in text, and only meaningful, repeated concepts are usually abbreviated for better readability (Nenadić et al., 2002; Sánchez and Isern,

2011). In KPEX, abbreviation definitions for phrases of up to five tokens are detected using regular expressions. The abbreviation pattern starts from the conventional rule of a phrase followed by a parenthesised sequence of letters that matches the first letter of each word in the phrase. Modifications to this rule were then made to also account



(a) Tagging confidence over standard tags.



(b) Tagging confidence over tag sets + adjective/noun conflation.

Figure 4.2: The tagging confidence achieved by TreeTagger on keyphrases (continuous lines) in comparison to non-keyphrases (dotted lines), when considering standard tags (4.2a) versus tag sets with conflated tags, as allowed by the keyphrase grammar (4.2b). The measure is given as a percentage.

for plurality, introductory words (such as '*or*' and '*also known as*'), keyphrases that contain abbreviations, and for the abbreviation occurring before the phrase itself. The following example illustrates the cases that KPEX can handle:

```
public transport networks (PTNs)
public transport networks (or PTNs)
public transport networks (PT networks)
public transport networks (public TNs)
PT networks (public transport networks)
```

When outputting candidates with abbreviations, KPEX will output both versions of the terms, separated by a special marker. For a 1-word abbreviation, for example '`PTNs`', the expanded version '`public transport networks`' will be displayed first, while for a keyphrase containing abbreviations, such as '`PT networks`', this shorter version will take precedence, for better readability.

Candidate filtering

The generated list of keyphrase candidates is now filtered for false positive, invalid or otherwise unwanted terms. Filtering proceeds with respect to 3 factors: stoplists, term length and token composition.

By default, KPEX filters keyphrase candidates through 2 stoplists: one for beginnings and ends of terms and one for full-term matches. Making the distinction between margin and full-match stoplists is useful since it allows a more concise specification of false-positive cases. By using margin stoplists, problematic prefixes such 'new' or '*number of*', and suffixes such as '*list*' or any single character are identified and removed from the keyphrase candidate, after which the candidate is rechecked if it still matches the grammar. Regarding the full-match stoplist, terms such as '*current study*' or '*Figure 1*' are common occurrences in scientific articles that do not bear information relevant to the topics being discussed. A full-match stoplist is a convenient way of dealing with such cases, especially if regular expressions are used.

In addition, a third, optional stoplist that is domain-specific may be chosen at run time to help minimise superfluous output for users knowledgeable of the area, such as '*programming language*' for computer science or '*amino acid sequence*' for biomedicine. KPEX's initial domain stoplist was for computer science and contained 56 entries.

It was compiled through observation and by inspecting terms' inverse document frequency across document collections. Through the integration of KPEX within the ScienceWISE platform (to be discussed in Section 4.3.4), the stoplists for physics and biomedicine are now being constructed through expert user feedback on the extracted terms.

Special Unicode characters such as those of non-Latin alphabets, mathematical symbols, the copyright symbol (©) or the em dash (—) are often tagged as nouns and may yield false positives. Consequently, candidates formed primarily of tokens shorter than 3 characters, those beginning or ending with Unicode punctuation marks, and those above 5 tokens in length are filtered out by default.

Term Variation Merging

As the last step of the text transformation stage, filtered keyphrase candidates found to be equivalent semantically are now merged together for frequency calculation purposes. Depending on specified command-line arguments, KPEX may output stemmed or lemmatised output to help aggregate different term variations into one. This can be beneficial for the purposes of achieving more diversity in the output list and increasing the chances of matching human-assigned keyphrases. As Turney (1997)) remarked, the keyphrases chosen by a human for a document never seem to map to the same stemmed phrase. This suggests that aggregation by stemming could be conducted by default when automatically extracting keyphrases. When processing standard language text such as media articles or for the purposes of taxonomical categorisation, stemming is beneficial as it produces only one representative term form for a semantic class of concepts. In considering this feature for KPEX, stemming using the well-known Porter stemmer (Porter, 1980) was found to aggregate terms too aggressively to be used by default. Scientific articles are generally rich in terminology, and certain terms with the same stem might have different semantic meanings. As a depiction of this behaviour, consider the following list of terms:

```
index          indexed
indexable      indexer
indexation     indexing
```

The list is reduced to one single stem, '*index*', by the Porter stemmer, although it contains several semantically distinct concepts. KPEX's design decision was to not make it

too restrictive in terms of the terminology it could extract, but at the same time manage to adequately aggregate semantically equivalent term variations. A useful compromise that was employed instead, was aggregation by lemmatisation. For English, lemmatisation implies the grouping of the different inflected forms of a word, created primarily by means of a prefix, suffix, infix, or an internal modification such as a vowel change (Brinton, 2000). Lemmatisation is a more complex process than stemming, requiring knowledge of the grammar of a language. In the above example, TreeTagger found the term '*indexed*' to be a past participle verb, hence its lemma was the verb '*index*'. All other terms were found to be either adjectives or nouns, their lemmas remaining unchanged. Lemmatisation therefore maintains more semantic differentiation among terms than stemming and is consequently carried out in KPEX by default for merging purposes. Terms that map to the same lemma are merged together, with the most frequent term variant being chosen as the one to output, unless lemmatised output was specifically requested at run time.

KPEX has no additional consideration of lemmas' POS information when merging. This means that part of the polysemy problem still remains, e.g. the noun '*index*' is not treated as distinct from the present-tense verb '*index*', and the terms will be aggregated. However, the probability of encountering this issue drops off significantly as the number of tokens in a keyphrase increases, and was mediated through KPEX's ranking procedure that favours multi-token terms.

The other candidate merging steps considered are listed below and align to previous efforts of term variation management (e.g. Torii et al. (2007); Kim and Kan (2009); Spasić et al. (2013)).

- Merging of different character casings of a term into the most frequent one;
- Merging of variants with or without the preposition '*of*' in them, such as '*cluster of galaxies*' and '*galaxy cluster*';
- Merging of inconsistently hyphenated or spaced terms, e.g. '*semimetals*' – '*semi-metals*' – '*semi metals*';
- Merging of abbreviations and their expanded versions;
- Merging of names in '*First-name Last-name*' form with '*Initial. Last-name*' or just '*Last-name*'.

4.2.4 Term Weighting

KPEX employs a modified version of the C-value algorithm (Frantzi et al., 2000) for weighting candidate keyphrases. The method (introduced in Section 4.1.1) was chosen because it was found empirically to yield promising results, despite its simplicity: it uses only term frequency and term length as features. For convenience, the C-value formula is recounted below:

$$C\text{-}value(a) = \begin{cases} log_2(|a|) \cdot f(a) & \text{if } a \text{ is not nested} \\ log_2(|a|) \cdot f(a) - \frac{1}{|T_a|} \sum_{b \in T_a} f(b) & \text{otherwise} \end{cases}$$

where a is a term,
$|a|$ is the number of tokens in a,
$f(i)$ is the frequency of term i in a collection,
T_a is the set of candidate terms that contain a.

The three modifications made to the formula in the implementation of KPEX are described below.

1. The value of the normalising logarithm (the frequency multiplier)

The number of tokens in a keyphrase is usually an indicator of its specificity. For example, the term *'cluster of galaxies'* is more specific than *'cluster'* alone. However, as Kim and Kan (2009) remarked, not all tokens contribute to the information content of the keyphrase (like the preposition *'of'* in this example). Therefore, words functioning as prepositions, conjunctions and determiners were left out from the computation of a term's frequency multiplier, to foster a better weighting scheme. Numbers, although they can be considered to add specificity to a term, were also left out because they mostly denote quantities or amounts which do not change the underlying semantics of the base concept.

2. The ability to extract keywords as well as keyphrases

Keywords (single-word keyphrases) are often treated as a separate case when it comes to keyphrase extraction (Turney, 2000; Kim and Kan, 2009) because of the inherent bias towards shorter term use when writing natural language, in the interest of

brevity and legibility. This brings up the need to normalise the considered frequency of occurrence of keywords in articles so that they do not dominate the output list of keyphrases. There is also a problematic case when keywords are used as substitutes for other keyphrases within the same article. As Barker and Cornacchia (2000) noted, long noun-phrases are usually not repeated too frequently in a document, being often replaced by shorter equivalents. For example, a physics article examining a *'cluster of galaxies'* might mention this longer concept once, in the beginning of the document, with all subsequent references to this concept made only as *'the cluster'*. A semantic disambiguation mechanism is usually required to recognise that all references of *'cluster'* actually relate to the longer construct *'cluster of galaxies'*. Barker and Cornacchia (2000) attempted to bypass this problem by choosing keyphrase candidates among noun phrases whose head nouns had the N-highest frequencies in the article.

In the implementation of KPEX, C-value's mechanism of discounting the frequency of keyphrase candidates as nested terms was found to perform satisfactory in neutralising shorter keyphrase bias. The only issue was to modify the algorithm so that the multiplication factor for a keyword's frequency $log_2(|a|)$ would be non-zero, but still normalise the score of the keyword adequately, in comparison to the scores of keyphrases. The level of required normalisation was derived by inspecting the distribution of manually-assigned keywords over documents and across curated knowledge bases, such as thesauri. Several sources were inspected for this distribution. Their keyword percentages are given in Table 4.6.

The average rate of occurrence of keywords across these sources is 24.75% of all existing terms. This percentage was therefore used to empirically evaluate a suitable modifier μ in the modified formula

$$log_2(|a| + \mu)$$

The goal was to have the percentage of output keywords near the theoretically optimal 24.75%, whilst still maintaining good performance in practice. The SemEval-2010 144-article Training dataset was used as the exemplar real-world collection to experiment with, as gold-standard keyphrase lists were readily available. Experiments were conducted with several sub-unitary values for μ. Figure 4.3 shows the resulting variance in output keywords, and the respective F_1 performance measure.

Table 4.6: Rates of occurrence of keywords (single-word keyphrases) across document collections and manually curated knowledge bases of various domains.

Data Source	*Keyword %*	Source	Size	Domain
Document Collections				
Hulth (2004)	13.70	Indexers	1000 abstracts	CS
Wan and Xiao (2008) dataset	17.32	Readers	308 news articles	Multiple
SemEval-2010 (train)	17.70	Readers	144 full-texts	CS
CERN-290 [b]	~20.00	Indexers	290 full-texts	Physics
HAL 2012 snapshot [a]	23.24	Authors	~434k full-texts	Multiple
FLoC 2010	25.19	Authors	681 full-texts	CS
Nguyen and Kan (2007)	26.59	Readers	154 full-texts	CS
Nguyen and Kan (2007)	28.99	Authors	205 full-texts	CS
SemEval-2010 (train)	29.70	Authors	144 full-texts	CS
NLM-500 [b]	35.78	Indexers	500 full-texts	Medicine
FAO-780 [b]	42.36	Indexers	780 full-texts	Agriculture
Thesauri / Controlled Vocabularies / Ontologies				
MESH 2009 snapshot [b]	~14.00	Thesaurus	~141.000 terms	Medicine
ScienceWISE [c] 2013 snaphot	17.01	Ontology	~24.000 terms	Physics
Agrovoc 2009 snapshot [b]	~36.00	Thesaurus	~40.000 terms	Agriculture
Average Percentages				
CS collections	23.64			
All collections	25.50			
Knowledge bases	22.00			
Overall	**24.75**			

[a] The HAL article repository (Baruch, 2007)
[b] Dataset used in Medelyan (2009)
[c] The ScienceWISE physics ontology (Aberer et al., 2011)

The best value for μ is 0.1, achieving both the closest keyword percentage to the optimum 24.75%, and also the highest F_1 measure. The best normalisation factor for keywords is therefore $log_2(1.1) \approx 0.137$.

3. The considered term frequency

The last modification made to the C-value formula was to only consider the frequency of a term within the article being analysed, and not across an entire article collection. It was found empirically that, with adequate normalisation and aggregation of term variants, the full-text of a scientific article conveyed sufficient information about the implied importance of the topics it discussed. KPEX's main purpose is to extract the main focus points of a document and possibly aid in the discovery of new, emergent

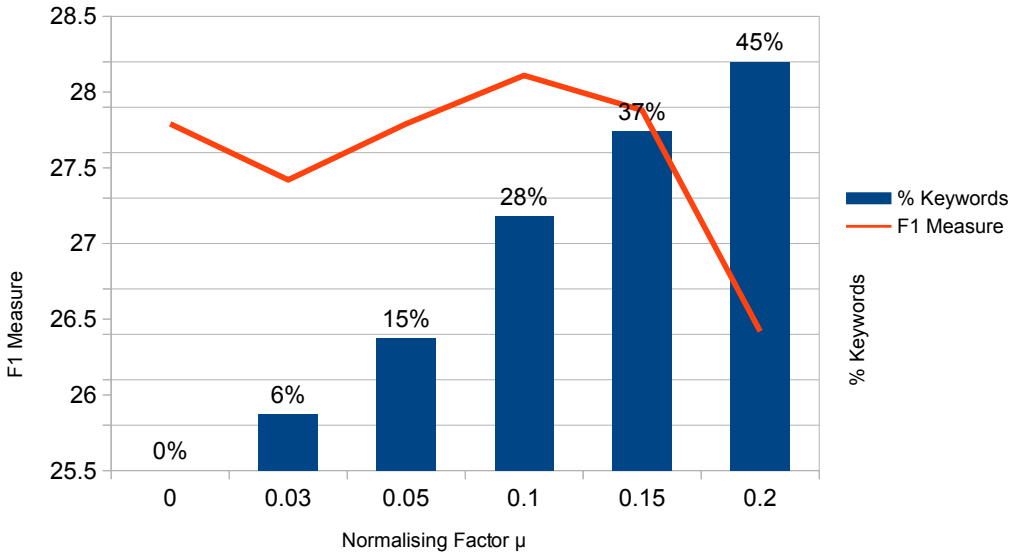


Figure 4.3: Variance of the percentage of keywords in KPEX output lists (top-15 terms) for different values of the normalising factor μ and the corresponding system performance. The dataset used is the SemEval-2010 144-article Training collection.

concepts. Using collection frequency in the process was likely to skew the results in favour of concepts already established within the corpus, inhibiting the extraction of novel terms as soon as they appeared in publications.

Considering all three proposed modifications to the original weighting function, the formal definition of the modified C-value becomes:

$$Mod.\,C\text{-}value(a) = \begin{cases} log_2(max(|C_a|, 1.1)) \cdot f(a) & \text{if } a \text{ is not nested} \\ log_2(max(|C_a|, 1.1)) \cdot f(a) - \frac{1}{|T_a|} \sum_{b \in T_a} f(b) & \text{otherwise} \end{cases}$$

where C_a is the set of content-bearing tokens in a (i.e. excluding prepositions, conjunctions, determiners and numbers),
$f(i)$ is the frequency of term i in the article,
T_a is the set of longer candidate terms that contain a.

4.2.5 Post-processing and Output

By now, a ranked list of keyphrases has been generated using primarily statistical information from the input document. This stage makes some final adjustments to the ranking, to ensure that the output keyphrases better fit the user's intended purpose for them. Three factors are considered in this respect:

1. Keyphrases' provenance in terms of logical regions of the article, and their possible re-weighting with respect to this information;
2. The output terms' alignment with an external list of concepts of interest, and their possible re-ordering with respect to this list;
3. The possible removal of certain shorter keyphrases nested within longer ones, to increase the overall topic coverage.

1. Re-weighting by logical parent

As was discussed in Chapter 2, scholarly publications benefit from rich rhetorical structures that help readers easily identify elements of interest and navigate their content efficiently. This is highly-valuable information for a keyphrase extraction system. Its usage in practice is only limited by the scope and availability of structured, machine-readable article corpora. In KPEX, this problem is circumvented by pre-processing PDF articles with PDFX, the PDF-to-XML converter presented in Section 3.2. This expands KPEX's reach beyond the outputs of publishers or third parties that readily provide this additional markup.

As an interoperability feature, the PDFX XML is parsed into a plain-text file with special markers to indicate the start of logical regions, and it is this file that will be processed by KPEX. This approach is more versatile than tailoring the extractor for PDFX XML. Various other formats such as JATS XML, TEI P5 XML (Wittern et al., 2009) or LaTeX can also be parsed to the same plain-text representation, allowing the extractor to be equally proficient over all of them.

The weights given to logical regions function as score multipliers for keyphrases that occur within those regions. Each score is multiplied by the maximum weight out of all logical parents that the respective keyphrase has. A maximum value was chosen over a cumulative one since the latter was found empirically to favour terms occurring

in many sections, often too general to be useful. Integer weights from 0 to 5 are specifiable from the command-line for each of the 10 logical elements that KPEX was designed to process (given in Table 4.7). All other regions, such as headers, footers, acknowledgements or author affiliations were considered superfluous for the purposes of keyphrases extraction and consequently left out. This in turn also helped increase the precision of the extraction task by reducing the amount of noise in the input.

Table 4.7: Example weight combination for the 10 logical regions that KPEX can differentiate. Keyphrases occurring in title of the article are given double weight, while those occurring only within captions are left out completely.

T	Abs	H1	H2	H3	Intro	Concl	Body	Cap	Bib
2	1	1	1	1	1	1	1	0	1

2. Re-ranking with an external list of concepts

At this stage, the user has the possibility of tailoring the keyphrase ranking to his or her needs, by providing a list of concepts of interest. A numeric weight may be provided for each concept in this list, in order to prioritise the re-ranking procedure. If such weights are not provided, all concepts in the external list will be considered to have the same weight. These external weights, as well as the modified C-value scores for the extracted keyphrases will then be normalised and transformed into integer ranks. Any KPEX keyphrase not occurring in the external list will be considered as having the lowest rank within it.

The two ranks of each KPEX keyphrase (by modified C-value and the external one) are aggregated. The external list is given 50% weight by default, meaning that the resulting rank will be the average of the two. The following two use cases exemplify the post-ranking functionality:

- When a list of concepts ranked by a user's expertise is provided, KPEX's output will favour the external higher-ranked concepts more. This is useful in outputting, for example, a personalised relevance factor for each document, or for the task of matching articles with potential reviewers.
- When an unranked vocabulary of concepts is provided, any KPEX keyphrases not in this list will be demoted. This is useful for example in document indexing,

to suggest known indexing terms that can be used to catalogue the article, and that are also well-represented in the text.

3. Nested terms and coverage in the output list

At this point in the processing workflow, a keyphrase's score represents its importance in relation to the entire input text and optionally also in relation to an external list. The approach was found to produce satisfying results, but further gains were still possible by analysing the relationships among the output keyphrases themselves. As Turney (2003) noted, the list of keyphrases as a whole should also be analysed for coherence. This can help reduce term overlap and increase coverage of the topics discussed in the article. C-value's discounting of nested terms functions well to reduce their considered frequency with respect to the frequencies of the longer terms. However, there may still be cases when the list of top-N terms will contain both nested and longer terms. If the desired number of output keyphrases is low, for example 5, this repetition is costly in terms of topic coverage, as there is little room left for other useful keyphrases.

As a final step before outputting the results, KPEX thus passes through the keyphrases list one more time, with the aim of removing nested terms that do not contribute to a better description of the article's contents. The scores of overlapping terms determine which will be removed. A nested keyphrase will be considered for removal only when a longer keyphrase containing it is placed higher in the ranking. The opposite case, when the nested term is ranked higher, is considered to happen only if that nested term has shown sufficient independence to be meaningful when presented separately. The example in Table 4.8 illustrates this behaviour. It shows KPEX's output for the same article when 10, respectively only 5 keyphrases are requested.

The example in Table 4.8 has the term '*protein*' extracted as a separate keyphrase from the longer ones that contained it, because it occurred in diverse contexts throughout the article. The fact that other longer keyphrases that contained '*protein*' appeared in the top-10, and were ranked lower than '*protein*' itself, functioned as a justification for the extraction of '*protein*' as a separate, independent concept. When only 5 keyphrases were output however, the higher-ranked '*Protein Data Bank (PDB)*' keyphrase already covered the information content of the shorter term. '*protein*' was therefore discarded in favour of another, non-overlapping term ('*substitution matrix*'), with a better contribution to content coverage.

Table 4.8: Difference in KPEX output when requesting 10 and 5 keyphrases respectively. The nested keyphrase '*protein*' was treated differently when fewer keyphrases are requested, given the relative ranks of its similar, longer terms.

#	10 keyphrases	5 keyphrases
1	ligand-binding site	ligand-binding site
2	extreme value distribution (EVD)	extreme value distribution (EVD)
3	*Protein Data Bank (PDB)*	*Protein Data Bank (PDB)*
4	statistical model	statistical model
5	*protein*	substitution matrix
6	substitution matrix	
7	*protein structures*	
8	ligand-binding site similarity	
9	*protein-ligand interaction network*	
10	drug resistance	

Sample output

For a better appreciation of the the overall quality of extraction achieved by KPEX, Table 4.9 presents the top-15 extracted keyphrases from two scientific articles of the computer science and genomics domains, respectively. A verbose, annotated log of the processing of one of the articles is given in Appendix B, along with an extended view of the top-60 extracted terms for each of the two (Table B.1).

Table 4.9: The top-15 keyphrases extracted by KPEX from two scientific articles in the fields of computer science (Krauthammer and Nenadić, 2004) and genomics (Antoniou et al., 2003). KPEX was run over the PDFX output for the PDFs of the articles, with a region weight combination of 1111111110, to leave out Bibliography sections.

Krauthammer and Nenadić (2004)	Antoniou et al. (2003)
term recognition	TATA binding protein (TBP)
term identification	enhanced green fluorescent protein (EGFP)
automatic term recognition (ATR)	CpG island
natural language processing (NLP)	Locus control regions (LCRs)
protein names	methylation-free CpG islands
term mapping	open chromatin
term classification	tissue culture cells
gene names	housekeeping genes
gene and protein names	locus control
biomedical literature	dominant chromatin
information extraction (IE)	dominant chromatin opening
term identification process	dominant chromatin opening function
expanded forms (EFs)	transfected tissue culture cells
term occurrences	divergently transcribed promoters
Gene Ontology (GO)	position effect variegation (PEV)

4.3 Evaluation of KPEX

Previous sections of this chapter have presented the keyphrase features and weighting mechanisms in use, as well as the choices made in KPEX's design given this knowledge. This section goes through the two procedures employed for evaluating KPEX's performance. The first procedure involves benchmarking against existing state-of-the-art keyphrase extraction systems, by reusing the SemEval-2010 evaluation suite introduced in Section 4.1.2. The second procedure draws upon the insight gained from Section 2.2 on an arguably more insightful way of assessing keyphrase quality: assessing the success of real-world applications that make use of them. The ScienceWISE platform[19] provided two well-suited use cases in this respect: article indexing for personal collection management and domain-ontology enrichment through crowd-sourcing of scientific concepts.

4.3.1 Benchmark: The SemEval-2010 Challenge

Datasets

The SemEval data made available for the 2010 extraction task comprised 144 articles for trial runs and model training, with associated gold-standard keyphrases, and 100 articles for the actual evaluation. The whole collection remains available on the SemEval-2010 website (`http://semeval2.fbk.eu/`) for reuse in further research. All articles are conference and workshop papers collected from the ACM Digital Library, converted to plain-text using the `pdftotext` utility. For the evaluation of KPEX, the PDF versions of the articles were also retrieved and pre-processed with PDFX to recover their logical structures.

Each article was assigned three sets of keyphrases: those provided by the original authors, those selected by readers and an additional combination of the two. The collection of reader-assigned keyphrases was conducted by hiring 50 Computer Science students to read and extract keyphrases from the texts. Table 4.10 presents the demographics of the SemEval datasets. It shows the distribution over the four ACM categories that were covered[20]: C2.4 (Distributed Systems), H3.3 (Information Search

[19]The ScienceWISE platform – `http://www.sciencewise.info/`

[20]The categories were taken from to the 1998 ACM Computing Classification System – `http://www.acm.org/about/class/1998/`

and Retrieval), I2.11 (Distributed Artificial Intelligence – Multi-agent Systems) and J4 (Social and Behavioural Sciences – Economics). The table also shows the total number of author, reader and combined keyphrases assigned to the collections.

Table 4.10: Demographics of the SemEval datasets.

Dataset	Size					Keyphrases		
	Total	ACM Category				Author	Reader	Combined
		C	H	I	J			
Train	144	34	39	35	36	559	1824	2223
Test	100	25	25	25	25	387	1217	1482

Evaluation procedure

Participating systems were asked to submit outputs containing the top 15 most relevant keyphrases of each article in the Test collection. These were then compared by the organisers to the gold-standard keyphrase lists extracted by humans. Precision, Recall and F_1 measures were computed at three keyphrases thresholds – Top 5, Top 10 and Top 15 – using micro-averaging to derive corpus-wide statistics. The micro-averaging gave each extracted keyphrase equal weight, the F_1 measure being calculated over global Precision and Recall values, as follows:

$$P_{global} = \frac{\sum_{i=1}^{N} |TP_i|}{Th * N}$$

$$R_{global} = \frac{\sum_{i=1}^{N} |TP_i|}{\sum_{i=1}^{N} |G_i|}$$

$$Micro\_F_1 = \frac{2 * P_{global} * R_{global}}{P_{global} + R_{global}}$$

where Th denotes one of the three thresholds (5/10/15),
N is the number of documents in the collection,
TP_i is the set of true positive keyphrases extracted for article i,
G_i is the set of gold-standard keyphrases for article i, i.e. $TP_i + FN_i$ (false negatives).

Both human-assigned and automatically extracted keyphrases were stemmed using the English Porter stemmer (Porter, 1980) to reduce discrepancies due to the different

forms in which a term might have appeared in text. Prior knowledge of the author-assigned keyphrases was forbidden. Occurrences of 'Keywords' sections in the plain-texts of the articles were manually removed by the organisers. In evaluating KPEX, this information was also removed from the parsed PDFX XMLs of the original PDFs, to comply with the competition rules.

Official 2010 Results

Table 4.11 shows the official results obtained by the participating systems of the SemEval challenge over the Test dataset, in 2010. The results reported by You et al. (2012) on the same collection are also represented, as they compare favourably with those of the original participants. The performance of the Maui system (Medelyan, 2009) is highlighted, as it is an extension of the popular term extraction software Kea (Frank et al., 1999; Witten et al., 1999), the tool commonly used in keyphrase extraction literature for comparative evaluations.

4.3.2 Revision of the SemEval-2010 Procedure

The SemEval initiative has been very beneficial to the text mining community for providing a direct comparison of different keyphrase extraction approaches. However, several particularities of the original evaluation procedure imposed an upper-bound on the maximum achievable performance of systems. An analysis of the gold-standard keyphrases of the Test dataset conducted by the competition organisers revealed that only 81% of the terms assigned by authors and 85% of the ones assigned by readers actually occurred in the article texts. For the evaluation conducted in this dissertation, theoretical maximums were also computed for micro-averaged Precision, Recall and F_1 measure over the combined gold-standard keyphrase set. In the first instance, the theoretical maximum Recall was found to be 91%. However, the competition imposed a restriction that further diminished this value: systems were requested to output precisely 15 keyphrases per article, but the distribution of keyphrases over the collection was not uniform. As shown in Figure 4.4, the number of keyphrases manually assigned to each article ranged from 8 to 37 for the training data (15.4/article on average) and from 8 to 28 for the test data (14.7/article on average). As systems could only match at most 15 keyphrases per article, the maximum achievable Recall was 86% when using the plain-texts of articles provided by the organisers.

Table 4.11: Official SemEval-2010 results for the participating systems, as reported in Kim et al. (2010), along with the results obtained by You et al. (2012) and baseline systems: TF-IDF, Naïve Bayes (NB) and Maximum Entropy (ME). The performance of the Maui system is highlighted for reference.

#	System	Top 5			Top 10			Top 15		
		P	**R**	F_1	**P**	**R**	F_1	**P**	**R**	F_1
1.	HUMB	39.0	13.3	19.8	32.0	21.8	26.0	27.2	27.8	27.5
2.	You et al. (2012)	–	–	–	–	–	–	26.2	26.8	26.0
3.	WINGNUS	40.2	13.7	20.5	30.5	20.8	24.7	24.9	25.5	25.2
4.	KP-Miner	36.0	12.3	18.3	28.6	19.5	23.2	24.9	25.5	25.2
5.	SZTERGAK	34.2	11.7	17.4	28.5	19.4	23.1	24.8	25.4	25.1
6.	ICL	34.4	11.7	17.5	29.2	19.9	23.7	24.6	25.2	24.9
7.	SEERLAB	39.0	13.3	19.8	29.7	20.3	24.1	24.1	24.6	24.3
8.	KX_FBK	34.2	11.7	17.4	27.0	18.4	21.9	23.6	24.2	23.9
9.	DERIUNLP	27.4	9.4	13.9	23.0	15.7	18.7	22.0	22.5	22.3
10.	Maui	35.0	11.9	17.8	25.2	17.2	20.4	20.3	20.8	20.6
11.	DFKI	29.2	10.0	14.9	23.3	15.9	18.9	20.3	20.7	20.5
12.	BUAP	13.6	4.6	6.9	17.6	12.0	14.3	19.0	19.4	19.2
13.	SJTULTLAB	30.2	10.3	15.4	22.7	15.5	18.4	18.4	18.8	18.6
14.	UNICE	27.4	9.4	13.9	22.4	15.3	18.2	18.3	18.8	18.5
15.	UNPMC	18.0	6.1	9.2	19.0	13.0	15.4	18.1	18.6	18.3
16.	JU_CSE	28.4	9.7	14.5	21.5	14.7	17.4	17.8	18.2	18.0
17.	LIKEY	29.2	10.0	14.9	21.1	14.4	17.1	16.3	16.7	16.5
18.	UvT	24.8	8.5	12.6	18.6	12.7	15.1	14.6	14.9	14.8
19.	POLYU	15.6	5.3	7.9	14.6	10.0	11.8	13.9	14.2	14.0
20.	UKP	9.4	3.2	4.8	5.9	4.0	4.8	5.3	5.4	5.3
	TF-IDF	22.0	7.5	11.2	17.7	12.1	14.4	14.9	15.3	15.1
	NB	21.4	7.3	10.9	17.3	11.8	14.0	14.5	14.9	14.7
	ME	21.4	7.3	10.9	17.3	11.8	14.0	14.5	14.9	14.7

Maximum Precision was also diminished because of the 15-term restriction and the way in which it was computed – micro-averaging out of exact values for each threshold (i.e. 5/10/15 * the number of articles, which was 100). The highest possible number of matching keyphrases per article was at most the number of existing gold-standard keyphrases for that article, but not exceeding 15. As Figure 4.4 shows, more than half of the articles in the Test dataset had fewer than 15 gold-standard keyphrases assigned to them. Therefore, any keyphrase additionally extracted up to the imposed 15-term mark would have been considered erroneous. The maximum theoretical Precision was found to be 86.73% at the Top-15 threshold, yielding a maximum F_1 measure of 86.36%.

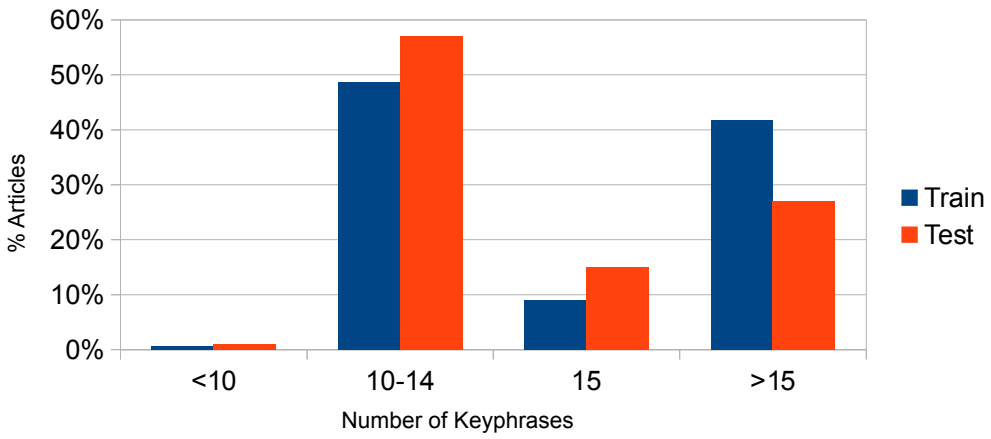


Figure 4.4: The distribution of combined gold-standard keyphrases (from authors and readers) across articles from the SemEval-2010 Train and Test collections.

In addition to the above limitations, the original SemEval evaluation procedure treated Precision and Recall differently at each keyphrase threshold. Recall at Top 5 and Top 10 was discounted, being always considered as a fractionary part of the global set of gold-standard keyphrases. Moreover, this global set did not have the 15-term per article limit. This meant that a regular F_1 score could have only been derived when computed at Top 15, and only if the number of gold-standard keyphrases for each article would have been 15.

In light of these considerations, a revised set of measures was defined for evaluating KPEX, that brought the maximum achievable F_1 score back to 100%. This was accomplished by (A) permitting varying numbers of extracted keyphrases per article; (B) treating Recall at each threshold relative to the gold-standard keyphrase subset seen up to that point; (C) considering effective thresholds for each article, with respect to the number of available gold-standard keyphrases. The formulas for the revised measures are the following:

$$P_{global} = \frac{\sum_{i=1}^{N} |TP_i^{Th(i)}|}{\sum_{i=1}^{N} (|TP_i^{Th(i)}| + |FP_i^{Th(i)}|)}$$

$$R_{global} = \frac{\sum_{i=1}^{N} |TP_i^{Th(i)}|}{\sum_{i=1}^{N} |G_i^{Th(i)}|}$$

$$Th(i) = min(Th, G_i)$$

where $Th(i)$ denotes the effective threshold at which to compute Precision and Recall for article i. It is the minimum between the desired hard threshold (5/10/15) and the actual number of gold-standard keyphrases available for that article. This factor prevents the treatment of keyphrases extracted beyond the number of entries in the gold-standard file as false positives. An example is the 15th extracted keyphrase in the case when the gold-standard contains only 14.

The revised measures were proposed to the SemEval-2010 organisers and original participants, as considerations towards a better appreciation of the true capabilities of systems. The proposal was well-received, and, with the consent of all involved parties, the original 2010 submissions for 17 out of the 19 participating systems were retrieved and re-evaluated using this updated procedure[21]. Authors for the remaining 2 systems (BUAP and UKP) could not be reached. The evaluation suite is available at `http://pdfx.cs.man.ac.uk/serve/SemEval_revised.zip` and also attached to this PDF document[22].

Revised Results

Table 4.12 shows the revised results obtained over the Test dataset, for the systems that have replied to the proposal in acceptance of the re-evaluation. Additionally, a run of the SZTERGAK system that has been further developed by its authors since the competition is also displayed, for a more complete picture of the current state-of-the-art in keyphrase extraction.

Several KPEX configurations were run over the same Test dataset, both in the plain-text form provided by the organisers, and as content extracted from the original PDFs using PDFX. The majority of runs over PDFX input achieved improved performance just by limiting the output to terms occurring in certain logical regions, without the need to set different region weights. The best run, however, did also make use of weights to emphasise regions of particular interest, as was explained in Section 4.2.5 on post-ranking. The combination of weights was taken from the best run achieved over the Training dataset, which was determined empirically. Table 4.13 presents the revised results obtained by KPEX over the Test dataset. Since the competition winner was judged by its performance over the Top 15 keyphrases, only results with this threshold

[21] Whilst the SemEval organisers and participants consented to the re-evaluation, the revised results presented in this thesis do not form part of the official SemEval Keyphrase Extraction challenge.

[22] For instructions on how to access the attached file, please see Section 5.2.

Table 4.12: Revised SemEval-2010 results obtained over the Test collection, for 17 of the original submitted runs of systems, baseline solutions (Maximum Entropy (ME), Naïve Bayes (NB) and TF-IDF), as well as an updated implementation of the SZTERGAK system. The performance of the Maui system is highlighted for reference.

#	System	Top 5			Top 10			Top 15		
		P	**R**	F_1	**P**	**R**	F_1	**P**	**R**	F_1
1.	HUMB	38.0	38.0	38.0	31.1	31.2	31.2	25.1	28.0	26.5
2.	SEERLAB	41.8	41.8	41.8	30.9	31.0	31.0	24.3	27.1	25.6
3.	WINGNUS	39.0	39.0	39.0	29.8	29.9	29.9	23.7	26.5	25.1
4.	ICL	33.6	33.6	33.6	28.5	28.6	28.6	23.3	26.0	24.6
5.	KP-Miner	36.6	36.6	36.6	29.3	29.4	29.4	23.1	25.8	24.4
6.	SZTERGAK (2014)	39.8	39.8	39.8	29.5	29.6	29.6	23.0	25.7	24.3
7.	KX_FBK	34.2	34.2	34.2	27.0	27.1	27.1	22.0	24.6	23.2
8.	SZTERGAK	31.8	31.8	31.8	25.4	25.5	25.5	20.8	23.3	22.0
9.	Maui	35.6	35.6	35.6	25.6	25.7	25.7	20.1	22.4	21.2
10.	DFKI	29.0	29.0	29.0	23.0	23.1	23.0	19.3	21.6	20.4
11.	SJTULTLAB	30.0	30.0	30.0	22.6	22.7	22.6	17.9	20.0	18.9
12.	DERIUNLP	25.4	25.4	25.4	20.6	20.7	20.6	17.6	19.7	18.6
13.	UNICE	27.8	27.8	27.8	22.3	23.4	22.3	17.4	19.5	18.4
14.	Likey	28.8	28.8	28.8	21.3	21.4	21.3	16.1	18.0	17.0
15.	UNPMC	17.8	17.8	17.8	18.5	18.6	18.5	16.0	17.9	16.9
16.	JU_CSE	26.8	26.8	26.8	19.6	19.7	19.6	14.6	16.3	15.4
17.	UvT	24.2	24.2	24.2	18.2	18.3	18.2	13.5	15.1	14.3
18.	POLYU	23.0	23.0	23.0	16.8	16.9	16.8	13.1	14.7	13.9
	ME	23.4	23.4	23.4	18.8	18.9	18.8	14.9	16.7	15.8
	NB[a]	23.4	23.4	23.4	18.8	18.9	18.8	14.9	16.7	15.8
	TF-IDF	22.0	22.0	22.0	17.8	17.9	17.8	14.1	15.7	14.9

[a]The performances of the ME and NB implementations were identical

are shown in the table for clarity, and to highlight the effect of different logical region combinations. A table containing KPEX results for all three keyphrase thresholds is given in the Appendix (Table C.1).

4.3.3 Discussion of the SemEval-2010 Results

The results obtained over the SemEval collection place KPEX first in the ranking, with a revised F_1 measure of 27.44% for the Top 15 extracted keyphrases. Interesting aspects highlighted in the experiment relate primarily to the use of logical structure information in the extraction process. The results obtained by KPEX when using this information highlight the many benefits of being able to skip, retain or emphasise

Table 4.13: KPEX results for the SemEval-2010 Test dataset, using the revised evaluation procedure. Results shown are for the Top 15 keyphrases, on runs with different weight combinations for the 10 different logical regions. The first column presents how much of the original full-text was covered with the selected regions (as a percentage over the number of words). The columns for the region weights represent, in order: T (*Title*), A (*Abstract*), $H_{1/2/3}$ (*Heading 1/2/3*), I (*Introduction*), Cn (*Conclusion*), Bd (Body, aside from I and Con), Cp (*Caption*) and Bb (*Bibliography*). The final column shows the improvement achieved over the full-text run (last row).

Text	**Region Weights**										**Top 15**			F_1
Cov.	**T**	**A**	**H_1**	**H_2**	**H_3**	**I**	**Cn**	**Bd**	**Cp**	**Bb**	**P**	**R**	F_1	**Diff**
21.1%	5	3	1	1		2				1	26.0	29.1	27.4	+4.1
(SemEval 1st place)											25.1	28.0	26.5	
9.5%		1	1							1	24.8	27.7	26.2	+2.9
9.2%	1	1								1	24.7	27.6	26.0	+2.7
2.5%		1	1								24.3	27.1	25.6	+2.3
2.1%	1	1									24.1	27.0	25.5	+2.2
(SemEval 2nd place)											23.7	26.5	25.1	
11.1%						1					23.7	26.2	24.9	+1.6
87.1%						1	1	1			23.1	25.9	24.4	+1.1
93.3%	1	1	1	1	1	1	1	1	1	1	23.0	25.7	24.3	+1.0
100%	*full-text*										22.1	24.7	23.3	

keyphrases of different logical regions. The key observations are the following:

- Leaving out superfluous text by abridging the input to the 10 logical regions of interest increases KPEX's performance over the full-text by 1 F_1 point.
- The Introduction section on its own achieves better results than any other single region, also outperforming the use of all 10 regions. This finding is in agreement with Shah et al. (2003), who made a similar observation for the field of Genetics.
- When using two regions in combination, the Title + Abstract, the most widely-available meta-information sources, further improve performance. This fact suggests that the two regions are indeed representative of articles' contents, but can also mean that readers seldom consider terms that are not mentioned in these regions as relevant to an article.
- The best 2-region combination is Abstract + H_1 (top-level heading), suggesting that collectively, these section headings hold more relevant information than the article title. A key observation is that even though this combination of regions accounts for only 2.5% of the full-texts, the performance achieved by KPEX in

this case is 25.6 F_1 points – a tie for second place, according to the data in Table 4.12.

- Since article titles are generally quite descriptive, and the Bibliography section of an article contains many titles related to the topics being discussed, adding this region to 2-region combinations brings further benefits. Title + Abstract + Bibliography, meta-information still commonly available in article stores, achieves very good performance, reaching the 26.0 F_1 point marker. Similarly, the combination Abstract + H_1 + Bibliography is a further incremental improvement, falling only 0.3 F_1 points short of the original SemEval leader (HUMB).

- Applying the best weight combination achieved over the Training set to the Test collection yields the best overall performance, with an F_1 score of 27.44%. This combination – 5311020001 – seems to adequately reflect the general rhetorical significance given by humans to the different regions of an article: the title is given most importance, as the element with the highest visibility, both stylistically within the PDF, and across article stores or search engines that index the document. The Abstract and Introduction follow, as a summarisation of the article's contribution, respectively a description of current knowledge and the key concerns addressed within the work. Section headings and the Bibliography complement the information gained from the other regions, by providing an overview of the organisation of the article over sub-topics, as well as its declared neighbourhood of related literature. A remark here is that the Conclusion section does not seem to contribute to improving performance. This is likely due to the fact that, in practice, the section is often simply a repetition of the claims made in the Abstract, with possible mentions of future work that inherently lead away from the core topics of the article in question.

Region Sensitivity Analysis

Figure 4.5 presents an analysis of the sensitivity of KPEX's extraction performance to the different regions of articles, as a tornado diagram. The best-performing combination of weights obtained over the Training set (5311020001) was treated as the baseline, with an F_1 score of 27.44. Then, each region weight was individually modelled as an uncertain value, by lowering and raising it by a margin of 2 units, to see how much impact it would have on the outcome.

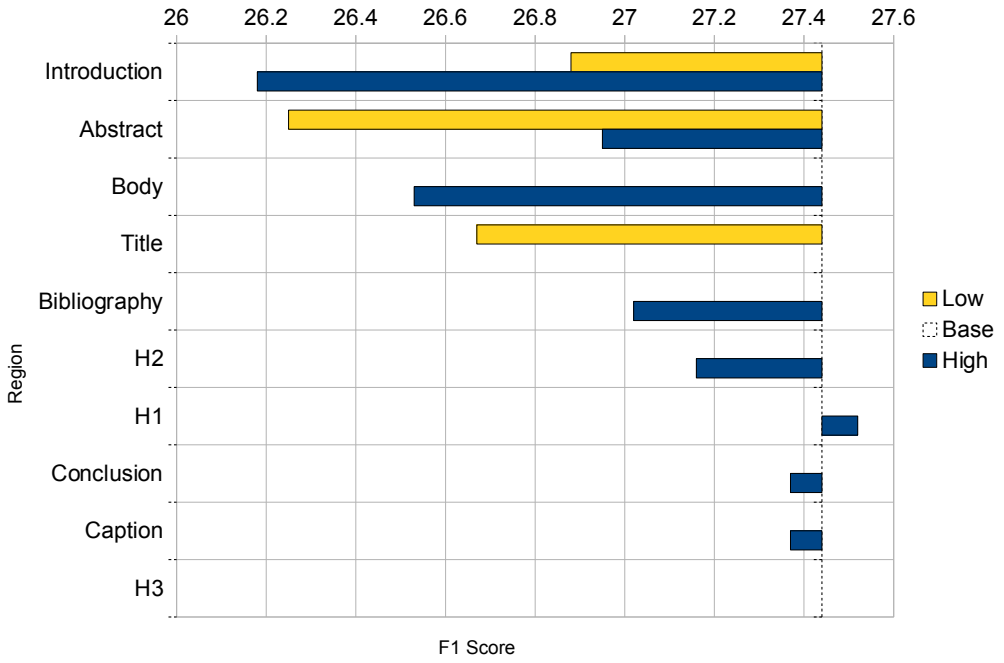


Figure 4.5: Tornado diagram showing the sensitivity of the extraction performance (in F_1 score) to the alteration of region weights. The 'Low' bars show the obtained score when each region's weight is decreased by 2 units. The 'High' bars show the same, for when the weight is increased by 2 units. Weight alterations of the regions at the top of the diagram yield the largest differences in extraction performance, suggesting an increased sensitivity of KPEX's performance to these elements.

Increasing the weight for the Introduction section yields the largest decrease in performance (1.3 F_1 points). This is due to the region's size, as the section customarily introduces many concepts discussed throughout the article. Boosting their scores thus places many candidates towards the top of the list and does not leave room for relevant, albeit less frequent keyphrases among the Top 15. Lowering the Introduction's weight also has a negative impact on score, but a diminished one (0.5 F_1 points), as terms found in this region are likely to also occur elsewhere in the text.

The Abstract follows the Introduction in influence, but here the score is diminished more if the Abstract's weight is decreased, rather than increased. This suggests that the Abstract holds many relevant keyphrases that do not occur often enough in the text for KPEX to place them in top positions, without a boosting factor. The baseline weight for the Abstract (3) seems to compensate for this well.

The core body text, having been left out from the baseline weight combination, only

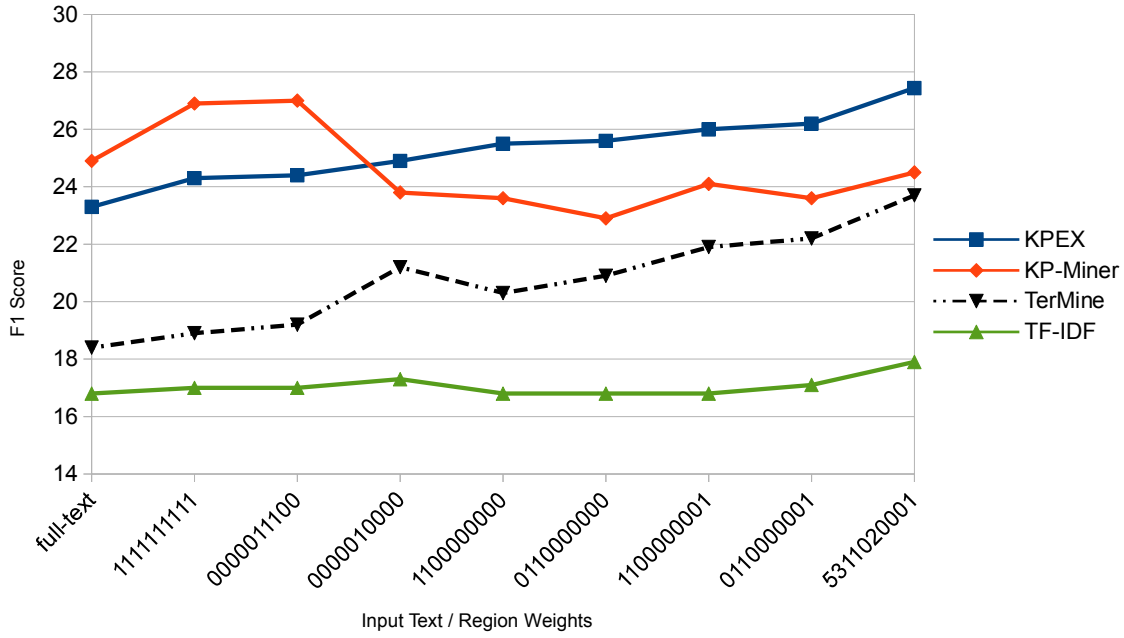


Figure 4.6: The effect of utilising structure information and region weights on different systems: KP-Miner (El-Beltagy and Rafea, 2010), the TF-IDF baseline, TerMine (Frantzi et al., 2000) and KPEX. The results shown were obtained over the SemEval 100-document Test dataset.

affects the score negatively when considered, as the amount of added noise is too large. In contrast, the Title is the opposite, contributing the most when left with the highest weight, as the terms it contains often have a diminished frequency of occurrence, but are relevant.

A final remark regards the small gain in performance that can be achieved over the Test dataset if the weight of top-level headings is increased. As this weight combination was not derived from results over the Training dataset, it was left out of the results reported in Table 4.13, to foster a fair comparison with the original SemEval-2010 participants.

Impact of Logical Structure on Other Systems

The promising results achieved by KPEX when using structure information have prompted an investigation on whether other extraction systems would benefit from this feature in a similar way. Figure 4.6 presents the differences in performance achieved by three other systems over the SemeEval Test dataset, when structure information is applied as post-ranking feature.

The first chosen system, KP-Miner (El-Beltagy and Rafea, 2010), was a contestant in the original SemEval challenge. KP-Miner was kindly made available for the purposes of this research by its original authors. It was chosen for having stood out in the SemEval competition results, as with only a small set of document-scoped features that did not include article structure information, the system performed very well, ranking fourth using the revised procedure. The features the system did use were different than the ones used by KPEX (please see Table 4.1, page 87). It was therefore interesting to inspect how adding structure information as a post-ranking feature influenced KP-Miner's output. The second system was a TF-IDF solution, chosen for reference as it is the best-known term extraction method. The original TF-IDF version used in the SemEval competition was unavailable, so the solution had to be reimplemented. The reimplementation had a higher F_1 score over the full-texts than the TF-IDF run presented in Table 4.12 (16.8). The third chosen system was TerMine (Frantzi et al., 2000). Whilst not a SemEval-2010 contestant, TerMine is the original implementation of the C-value algorithm that KPEX is based on. Its inclusion in the following diagram has two roles. First, it fosters an appreciation of the improvements brought by KPEX's design to the core algorithm. Second, it shows how structure information and input abridging can alleviate possible shortcomings of the extraction workflow, such as input sanitisation and candidate filtering.

The systems were run over two different inputs: the full-texts provided by the SemEval organisers, and the abridged text from the 10 logical regions parsed from PDFX output, to which weights were attributed. For all three additional systems, structure information was applied similarly to KPEX, as a post-ranking feature. Term scores output by each system for the abridged input, were multiplied by the maximum weight value for the logical regions in which the respective terms occurred.

Regarding KP-Miner's results, the weight combination 1111111111 just strips the input of superfluous regions such as author affiliations and copyright statements, yet it was enough to yield a sizeable 2.0 F_1 point increase over analysing the full-texts. This score (26.9) is better than the one the system obtains when using KPEX's best weight combination (24.5). The behaviour is likely due to the use of the position of a term's first occurrence as a feature. KP-Miner already boosts the scores of the candidates found to occur early on in the text. Boosting them again with region weights would essentially have the front matter dominating the output. Additionally, KP-Miner

imposes a 400-word cutoff into the input text for selecting keyphrase candidates (El-Beltagy and Rafea, 2009). Any terms that appear for the first time after the cutoff are ignored. Author affiliations can take up 30-50 words towards this limit. A copyright statement, when present, usually occupies the bottom part of the first column of ACM's two-column layout and, along with the publication venue and date, can take up 70-80 words as well. PDFX input was therefore beneficial in helping KP-Miner to consider more valid keyphrase candidates. Another observation is that front-matter information alone seems to not be enough for KP-Miner to derive good keyphrase suggestions, as the system also requires terms to occur at least 3 times in the text in order to be considered. KP-Miner therefore needed the body text present in order to derive adequate statistics and perform well. The system's best result of 27 F_1 points was achieved, rather surprisingly, over just the core text of the articles (Introduction + Conclusion + Body), without requiring front-matter information. This score would place KP-Miner ahead of the original SemEval-2010 leader (HUMB), in the ranking of Table 4.12.

Differences in the results of the TF-IDF implementation were less visible, with only a 1.1 F_1 point increase being achieved for the final combination of weights (5311020001). Normalising the original TF-IDF scores prior to weighting did not make a noticeable difference either, because too many terms from each region were proposed. This made a boost in a region's overall scores less effective, as false positives (in terms of the gold-standard keyphrases) would also get promoted along with true positives. A noteworthy aspect, nonetheless, is that TF-IDF's achieved performance was the same over the full text as it was over the title and abstract, regions that account for just over 2% of the entire narrative. This means that the information obtained from running TF-IDF over the full-text had the same value as when its output was limited just to terms of the front-matter. Importantly however, the overlap between the true positives obtained from the full-text and those obtained from Title + Abstract is only 57% on average. This suggests that an aggregation of the results of the two runs has the potential to yield a significant increase in performance.

The TerMine system benefits most from structure information. A 5.2 F_1 point increase over the full-text run is achieved with KPEX's best weight combination – more than is seen for KPEX itself. In this setting, input abridging seems to have functioned well to eliminate many false positive term occurrences from consideration. The score gap by which KPEX outperforms TerMine drops from 4.9 to 3.7 F_1 points, between analysing the full-texts and using the best weight combination.

Overall, knowledge of an article's organisation over logical units thus proves to be advantageous across term extraction approaches, although not uniformly. Depending on the specific features used by a system, different regions weights contribute different benefits.

Additional Remarks

A shortcoming of KPEX's extraction procedure seemed to be that its method of conflating similar terms was too limited. For several articles, KPEX extracted multiple terms that were semantically very close in the context of the article, e.g.

```
protocol module
protocol
protocol framework
```

However, the gold-standard list only contained one of these variants (`protocol framework`), because it was enough to capture the required information. A better variant conflation method would be a first avenue to KPEX's future improvement.

Another shortcoming was that the input provided by PDFX to the extractor was not 100% accurate for all articles. The Introduction section was not identified correctly for two articles, the Conclusion section for one article, while several tables of data and formulae were considered body text, adding noise to the input. For the purposes of this experiment, the output of PDFX was not manually corrected, in order to showcase a more realistic extraction scenario of employing this two-step (PDFX+KPEX) process.

Overall, the proposed approach has proven to be successful, achieving better results than the SemEval-2010 contestants, being 0.9 F_1 ahead of the challenge's leader. Benchmarking sets of keyphrases against gold-standard lists was convenient for a comparative evaluation against the state-of-the-art. However, as Turney (2000); Barker and Cornacchia (2000); Barriere and Jarmasz (2004) and others have remarked, more than a single list of terms can be deemed as a high-quality description of an article's contents. Section 2.2 discussed this issue and highlighted that the precise purpose in mind for the extracted keyphrases (i.e. the specific application that will use them) is also an adequate, if not better, assessor of their quality. A second evaluation experiment will thus inspect KPEX's performance in a real-world setting, by collecting expert opinions on the quality of the extracted terms. The ScienceWISE platform (Aberer et al., 2011;

Boyarsky et al., 2012) was very effective in providing such feedback, as through it, experts annotate and help organise authoritative corpora of publications as part of their daily scientific work. The next section will detail this additional experimental setup and its outcomes.

4.3.4 Real-world Performance: The ScienceWISE Experiment

A major goal of the ScienceWISE project has been to utilise crowdsourced efforts of scientists to sustain an interactive research environment through routine tasks of literature review and management of personal article collections. This environment is linked to field-specific ontologies and has direct connections to periodically-updated corpora of research papers. The project started in 2009 with a focus on the field of physics. The document corpus of reference is arXiv.org – the standard physics preprint server since the early 1990's. Since its creation, the physics branch of ScienceWISE has amassed over 1000 users, 400k articles and 24k semantically-interlinked concepts in its ontology. The ScienceWISE team has kindly agreed to integrate KPEX within the system's article bookmarking facility to see how it could contribute to this already established scientific workflow. The performance indicators measured were (A) the extent to which users considered KPEX-extracted keyphrases as relevant to the article being bookmarked, and (B) how many new valid scientific concepts it could help identify, that were not already in the ontology.

The bookmarking of articles in ScienceWISE is customarily conducted with the set of controlled ontological terms that are found in the article texts. Suggested bookmarking terms are ranked by a modified TF-IDF implementation computed over the entire collection, to help users select meaningful concepts quickly (Prokofyev et al., 2012). The implementation puts all concepts found in the title of the document at the top of the output list, followed by all the concepts in the abstract, and the rest of the standard TF-IDF ranking. An illustration of an ontological entry in the system and its bookmaking interface is given in Figure 4.7. The ontology entry (4.7a) shows a concept catalogued and linked to external definitions, as well as to other related concepts. The bookmarking interface (4.7b) shows the same concept selected for bookmarking a scientific article. The left column, '*Found concepts*', contains the modified TF-IDF ranking of concepts from the ontology, whilst the middle column, '*Chosen concepts*', contains a user's choices of bookmarking terms. The interface offers users four options

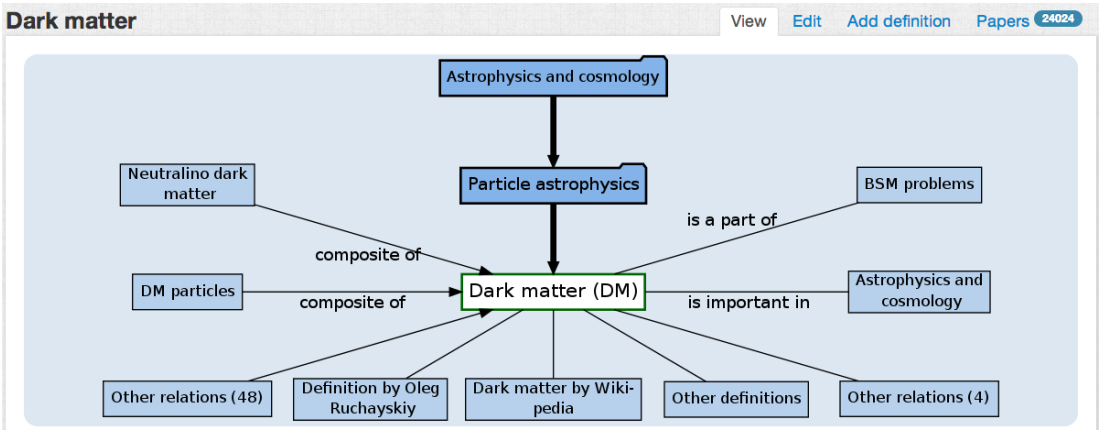
regarding the automatically extracted concepts:

1. Choose a concept as a bookmarking term – implying that it is considered relevant to the article content (e.g. '*dark matter*' for the article in Figure 4.7b);
2. Mark a term as a False Positive – implying that the term is erroneous or irrelevant to the field (e.g. '*Figure 3*', '*fact*' or '*International Conference*');
3. Neither choose a term, nor mark it as a False Positive – implying a valid concept, but not central to the article content (e.g. '*Bayesian*' for the article in Figure 4.7b).
4. Manually enter novel bookmarking terms – this starts the process of adding a new concept to the ontology: adding definitions, selecting categories, declaring links to other concepts, etc. This process is the main method through which ScienceWISE crowdsources its ontologies.

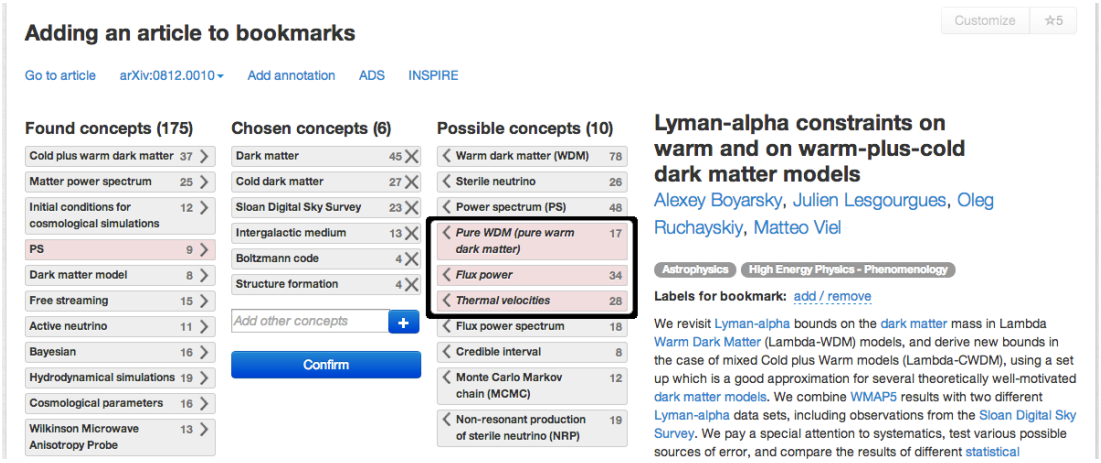
Integration with the Physics Branch

For the experiment, the bookmarking interface of ScienceWISE was extended to include a third column, '*Possible concepts*' (also shown in Figure 4.7b), containing the top-10 keyphrases extracted by KPEX. Users that volunteered to take part in this experiment were informed that a new method for automatic concept extraction was being tested, and were asked to help judge the quality of the extracted terms using the routine bookmarking procedure. In the first instance, the experiment provided direct expert feedback on keyphrase quality. Additionally, the selection of novel KPEX keyphrases for bookmarking facilitated also measuring the extent to which KPEX contributed to enriching the physics ontology, as it effectively equated to option (4) in the previous list of user choices.

Over the course of the experiment, KPEX was run with two different configurations – one with structure information and one without, in order to inspect how the usage of structure information would affect the two use cases. The weight configuration used was a slight alteration of the best-performing KPEX run over the SemEval-2010 dataset (5311020001). This was because preprocessing articles with PDFX was replaced by ScienceWISE's method of parsing the LaTeX sources of the imported arXiv.org articles. The method only differentiated the Title and Abstract of the article, treating everything else up to the bibliography as body text. This required that KPEX's weight



(a) Ontology entry for the concept '*Dark Matter*'.



(b) Bookmarking interface example.

Figure 4.7: The ScienceWISE ontology definition for the concept '*Dark Matter*' (4.7a), and the article bookmarking interface (4.7b), in which the left column contains the ScienceWISE ranking of concepts from the ontology; the middle column contains a user's choices for bookmarking; the right column, 'Possible Concepts', is populated with KPEX-extracted keyphrases. The three outlined keyphrases were not already existent in the ScienceWISE ontology.

combination effectively be changed to 5311111110, as the Title and Abstract could be emphasised, headings, captions, the introduction and conclusion were subsumed into the body text, and the bibliography was left out.

To maintain the integrity of the KPEX list for evaluation, keyphrases that KPEX extracted that were also in the ontology, were removed from the ontological list ('Found Concepts') when displayed to users. While limiting this list to the top-10 concepts as well would have fostered a fairer evaluation, it was considered too restrictive for the platform's users by the ScienceWISE team, as the experiment was run in a live setting.

Table 4.14 gives an overview of the resources involved in the ScienceWISE experiment, as well the results, in terms of users' judgements on the quality of the keyphrases extracted by KPEX.

4.3.5 Discussion of the ScienceWISE Results

User opinion seems to be in accordance with the SemEval experiment, that emphasising regions of interest produces better overall results. When weights were used, one in three keyphrases from KPEX's top-10 output was deemed relevant enough to be selected as a bookmarking term. At the same time, the number of novel terms deemed irrelevant diminished by more than half in comparison to the full-text run, to 11.5%.

More than half of the keyphrases extracted by KPEX were novel, i.e. not part of the physics ontology. Out of these, an average of only 8.4% were chosen for bookmarking articles and added to the ontology. Still, given the maturity of the physics knowledge

Table 4.14: Statistical overview of the ScienceWISE experiment. Keyphrases were judged as 'Relevant' (chosen as bookmarking terms), 'Adequate' (not chosen, nor marked as False) and 'Irrelevant' (marked as False).

1st Round (No Weights)			
Annotators	11		
Bookmarked Articles	128		
Avg. Chosen Concepts / Article	13.26		
KPEX Keyphrases	**All**	**Ontological**	**Novel**
	1194	519	675
Relevant (Chosen)	324 (27.1%)	261 (50.3%)	63 (9.3%)
Adequate	709 (59.4%)	258 (49.7%)	451 (66.8%)
Irrelevant	161 (13.5%)	–	161 (23.9%)

2nd Round (With Region Weights)			
Annotators	11		
Bookmarked Articles	138		
Avg. Chosen Concepts / Article	16.15		
KPEX Keyphrases	**All**	**Ontological**	**Novel**
	1224	541	683
Relevant (Chosen)	418 (34.2%)	367 (67.8%)	51 (7.5%)
Adequate	727 (59.4%)	174 (32.2%)	554 (81.0%)
Irrelevant	78 (6.4%)	–	78 (11.5%)

base, and post-experiment user feedback, this figure was considered promising, as will be explained shortly. The percentage of ontology additions obtained when structure weights were used (7.5%) was lower than the one obtained when they are not used (9.3%). Overall, however, more KPEX terms were chosen for bookmarking with this setting, as the intersection with the ontology grew to more than 44%. For the use case of article bookmarking, structure weights were thus preferred, but as a support for ontology enrichment, it is interesting how KPEX functioned better when weights were not used. This was found to be due to the fact that the no-weight run extracted more concepts that were not emphasised in the articles' front matter, thus less likely to have been previously noticed and added to the knowledge base.

For a better appreciation of KPEX's contribution to the enrichment of the ontology, Figure 4.8 presents, for the days in which concepts were added or modified in the physics branch of ScienceWISE, how many of these came from KPEX's output. Despite accounting for only a small portion of the novel terms that KPEX extracted, the ones deemed relevant accounted for approximately 44% of all concepts being added or altered over the course of the experiment. This suggests that the extractor permitted

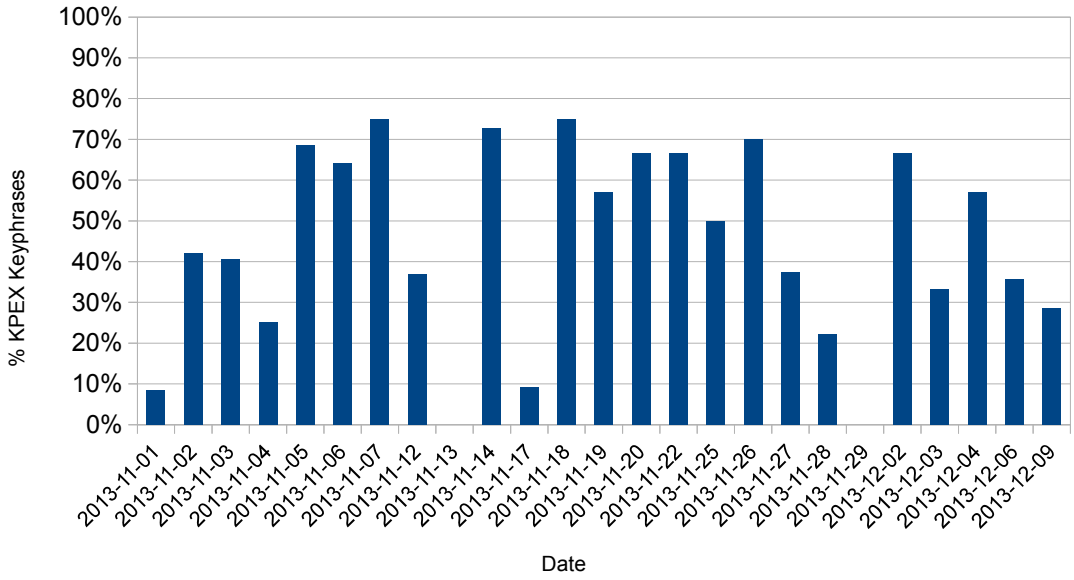


Figure 4.8: Percentages of KPEX-extracted keyphrases involved in additions or modifications of concepts to the ScienceWISE physics ontology during the course of the experiment, out of all modifications made. The percentages shown are derived from the entire physics user community's contributions during this time, not just from that of the annotators involved in the experiment.

users volunteering in the experiment to be highly proficient in enriching the already mature ontology. KPEX either extracted new relevant concepts that were not in the ontology previously, or alternative forms for existing concepts, such as full or partial abbreviations.

User Experience

Following the experiment, volunteers were asked about their user experience with the KPEX system. Overall, the extractor was perceived as a welcome addition to the services provided by ScienceWISE. It facilitated the addition of novel relevant concepts to the domain ontology, and in many cases was able to identify and propose existing ontology concepts as bookmarking terms, without prior knowledge of their significance within the physics field.

While it helped develop the ontology, KPEX was not considered a substitute for it in the bookmarking use case. The quality of the ontological concepts was superior, because they had been crowdsourced by domain experts. For the purpose of bookmarking scientific articles in ScienceWISE, these concepts already established within the field were considered a necessity. Out of all terms that KPEX extracted, which pre-existed in the ontology, 59% were chosen for bookmarking on average, whilst out of the novel terms, only 8.4% were chosen. One reason for this was that the physics ontology was developed enough to cover the majority of concepts relevant to the field. Most of the keyphrases not covered by the ontology were likely either not represented well enough in the literature (e.g. terminology used only by a particular person or department), or not directly related to the field of physics (e.g. statistical measures or general terms). A suggested improvement was that KPEX also make use of article collection information or external knowledge bases – similar to several SemEval-2010 keyphrase extraction systems. As was explained earlier in this chapter, the extractor does have the ability to re-rank its output with respect to an external information source, but a focus point in KPEX's evaluation was to examine its proficiency when using features solely intrinsic to the input article, such as its logical structure. Collection- and external-scoped feature use was deliberately avoided.

4.4 Summary

This chapter has addressed the topic of keyphrase extraction from scientific publications, going through an account of previous efforts, the variety of keyphrase features currently in use, as well as the presentation of a novel approach. The approach is rule-based, utilising only a simplistic set of features derived from the input document to carry out the extraction task. It identifies, filters and merges keyphrase candidates using linguistic methods, such as POS tagging and abbreviation identification. It then weighs the candidates using a modified C-value algorithm, to yield a statistical ranking of their significance within the document. Finally, it can use information of the article's rhetorical structure to alter this ranking in favour of keyphrases that occur in certain parts of the document. The system implementing the proposed method, KPEX, was evaluated in two settings: a benchmark against the state-of-the-art, and an experiment in a real-world setting. Both scenarios yielded promising outcomes.

The benchmark placed KPEX first amongst 18 other systems, when using a combination of logical region weights which seems to adequately represent the parts of articles that best describe their content. Other combinations such as the abstract along with top-level headings, also achieved very good results, whilst only accounting for 2.5% of the articles' full-texts. This fact encourages the use of these regions in analysis pipelines with memory and processing time constraints. Experiments on two real-world use cases were additionally carried out within the ScienceWISE platform. As a result of positive overall results and feedback from domain experts, ScienceWISE has now fully endorsed and integrated KPEX as a means of ontology enrichment. An important remark is that when rhetorical structure information was used, only 6.4% of the terms output by KPEX were deemed erroneous or irrelevant to the field of study. This achievement attests to the system's overall success at extracting valid, domain-specific concepts from an article, irrespective of whether or not a human chose them for bookmarking. This is particularly useful because there are many text and data mining applications that do not require human relevance judgements for individual terms, such as document indexing, clustering, topic modelling and article recommenders. These would all benefit greatly from accurate automatic keyphrase extraction.

The conducted experiments have also highlighted suggestions for further improvement, that promise an even better approximation of the human perception of term relevance. These cover more involved methods of conflating semantically-similar terms, as well

as utilising collection statistics to help reduce false positive keyphrases in the output. Given the capabilities of the PDFX solution employed for structure recognition, an interesting exercise would be to attempt to use an article's own bibliography as the relevant article collection from which to derive these statistics.

A current performance bottleneck of the system is its string searching mechanism, as it is not optimised. For computing term frequencies, for example, and for determining the set of logical parents of each term, the Aho-Corasick algorithm (Aho and Corasick, 1975) could be employed to match all the terms simultaneously and get their positions in text. A similar approach may also be taken for the step of merging different term candidates that share the same lemma, as it is currently a time-consuming process.

A central aim of this dissertation's proposed approaches for PDF structure analysis and keyphrase extraction has been practicality. The issues addressed by these research topics have a current high demand for robust solutions, readily applicable to real-world data. The next chapter of this work will discuss how PDFX and KPEX are currently being used in practice, as well as additional interesting use cases enabled by the provided functionality. Notes on the availability of all relevant resources presented herein are also provided, in hopes of facilitating further research.

Chapter 5

APPLICATIONS AND AVAILABILITY

One of the motivating factors behind this research has been the richness of its application domain. Bridging the format gap between what publishers produce as output and what information services expect as input, is the key contribution of the PDFX system presented in Chapter 3. It conveniently converts camera-ready PDF publications into semi-structured XML representations that expose their logical structure. Likewise, the methodology implemented in the KPEX system, detailed in Chapter 4, represents a viable, robust solution for keyphrase extraction from English language texts[1]. It is particularly proficient over scientific articles for which logical structure information is available.

The presented workflow has been integrated, in full or in part, in real-world applications and other strands of research by various academic groups. This chapter describes several such use cases, also outlining the many other added-value services facilitated by the outcomes of this research. A recount of the availability of all resources used herein is additionally given, in terms of software, datasets, evaluation scripts and LaTeX sources of tabular data and formulae.

[1]The application of KPEX to other languages is dependent on the availability of an adequate part-of-speech tagger and on the analysed language following the same general syntax as English, so that KPEX's keyphrase grammar remains meaningful.

5.1 Applications

5.1.1 PDFX

The principal beneficiary of the structure recovery capability of PDFX is the field of text mining, where knowledge relevant to a specific domain or task is typically searched for across vast, unstructured document collections. The ability to distinguish between various logical divisions of an article such as the abstract, citations, tables or individual references can greatly decrease the search space and improve efficiency. The added-value brought grows in proportion to the size of the analysed corpora, and may include the highlighting of insightful trends or patterns in the data, unobtainable without knowledge of articles' rhetorical structures.

The ScienceWISE platform (Aberer et al., 2011; Boyarsky et al., 2012) used in the evaluation of KPEX, is under active development to provide new features to its users. It also employs PDFX to recover the core text of articles for which LaTeX or XML variants are not available, and expand its indexed collections. Current considerations are to also harvest reference and citation information as a precursor to an expert finding utility.

The Public Knowledge Project (MacGregor et al., 2014) has been using PDFX as part of a completely automated XML publication pipeline. PKP has put together a service-oriented toolchain of several scholarly parsing tools, including Pandoc[2], ParsCit (Councill et al., 2008), CiteProc [3] and PDFX, to provide a fully-automated solution for transforming article drafts into JATS-compliant XMLs. This allows low-budget open access publishers to progress beyond the PDF in their publication workflows, and generate the required meta-information for indexing, at no extra cost.

Crossref (Pentz, 2001) is a renowned DOI registration agency and not-for-profit network for publisher collaboration. Its amassed citation-linking network covers over 65 million journal articles, books chapters, theses and technical reports from thousands of scholarly and professional publishers. It has chosen to use PDFX to facilitate indexing, assignment of DOIs, and linking of bibliographic references for the numerous scholarly works of small publishers that come only in PDF form.

[2]The Pandoc markup converter – `http://johnmacfarlane.net/pandoc/`

[3]The CiteProc-JS implementation – `https://bitbucket.org/fbennett/citeproc-js/`

CiTalO (Di Iorio et al., 2013; Ciancarini et al., 2013) is a system for identifying the nature of citations (i.e. the reasons for which a work was cited in a particular context). It makes use of PDFX to extract citation information from publications and semantically annotates them with types from the CiTO ontology[4].

The Hiberlink project[5] (Sanderson et al., 2013) is a large-scale initiative aimed at gauging the extent to which web links in scholarly works presently fail to lead to the resources that were originally referenced. The project has employed PDFX over large corpora to identify web links in articles along with their logical parent regions (e.g. footnotes, references, etc.).

The 2012 NDBC/DBCLS BioHackathon event (Katayama et al., 2014) had as one outcome the integration of PDFX's functionality within several biomedical workflows. These included the extraction of front matter metadata from PDF article stores for better indexing, the identification of bibliographic references for better literature recommendation (Iwasaki et al., 2010), and the annotation of PDF-formatted manuscripts with bio-ontological entities.

The Partridge project[6] (Ravenscroft et al., 2013) combined PDFX and the SAPIENTA[7] system's functionality (Liakata et al., 2012) to provide scientific discourse annotation for PDF articles.

In another application, PDFX has been used to provide section-wise text and heading information to a system aimed at aligning scholarly documents with their associated slide presentations (Bahrani and Kan, 2013).

Other parties that are currently using or have expressed interest in PDFX are

- The Roche pharmaceutical company
- Harvard Medical School
- The University of Santa Cruz Center for Biomolecular Science and Engineering
- The UC San Diego department of Biomedical Informatics
- The CERN Document Server

[4]CiTO, the Citation Typing Ontology – `http://purl.org/spar/cito/`
[5]The Hiberlink project – `http://www.hiberlink.org/`
[6]The Partridge project – `http://papro.org.uk/`
[7]The SAPIENTA system – `http://www.sapientaproject.com`

- The Florida State University iDigInfo institute for Digital Information and Scientific Communication

In addition to the above projects and services, it is also worth noting that there are many use cases for a structure recovery system, when functioning as a personal tool. The following are some examples gathered from PDFX user feedback.

Linking of geometrical and logical structures. Having logical elements retain information about their geometrical layout in the original PDF facilitates overlaying the analysis result on top of the initial typeset version. Information about the correspondence between the geometrical and logical structures opens up a suite of possibilities for new applications. A visual example of this overlaying is given in Figure 5.1. With such a display, it is straightforward to facilitate correct cross-column text selection or the visualisation of citation and keyphrase contexts. This latter use case could support a faceted visualisation of the article's contents that would allow users to quickly access information of interest, whether this regards a specific keyphrase, paper or author.

Accessibility support. Reconstructing the flow of text into a single, steady stream of information is particularly useful for individuals with visual impairment. Relieving the content of formatting embellishments such as headers and footers can help screen readers maintain fluency of discourse. A study conducted in 2007 on 100 blind users of screen readers reported that their top cause of frustration was the page layout causing confusing screen reader feedback (Lazar et al., 2007). Additionally, detailed page content markup can expose previously intangible meanings of certain expressions. For instance, the meanings of phrases like "please see paragraph 1 of column 2 for ..." or "the third column of the table contains ..." lie beyond the current expressive power of screen readers.

Reading on a small screen. There are increasing technical demands from users for the ability to consume content on various electronic devices (Pettifer et al., 2011). Browsing through a multi-column PDF on the average smartphone is generally deemed inefficient because of the inability to wrap the content to screen width. Reading flow reconstruction can help solve this issue as well and PDFX can facilitate this.

Data replication. In addition to extracting the core text, PDFX also aims to extract figures and recognise tables. An author drawing upon the works of others might want to reuse such information in his or her own work as a premise, justification or for comparison purposes. In such cases, screen capturing, image editing or the manual

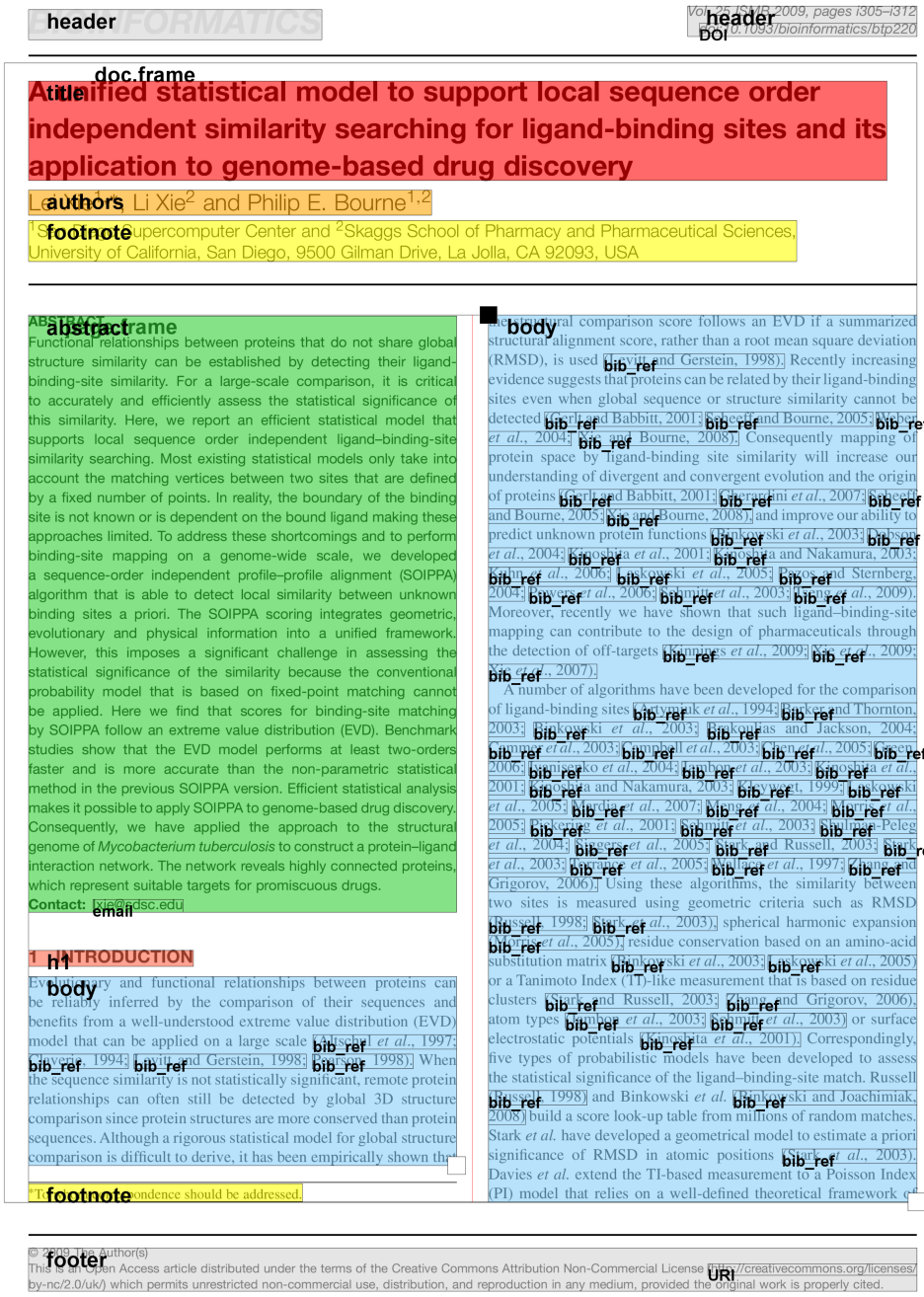


Figure 5.1: Example first page of a PDF article overlaid with PDFX's structural analysis result. A PDF viewer having access to this information could, for example, skip the intervening footnote and footer when body text is selected across the two columns. A similar view of the entire article is available at `http://pdfx.cs.man.ac.uk/example.overlaid.pdf`

reconstruction of tables and graphs could stop being a concern of the citing author. For example, the XML tabular data output by PDFX is transformable via XSLT to any other convenient format such as CSV or LaTeX.

Lastly, other features of PDFX meant to widen the system's application spectrum are the mapping of identified elements to DoCO[8] entities, the ability to output sentence-level tags, and DOI resolution for individual references. A more involved extension of PDFX will aim to also include reference string parsing by incorporating the ParsCit tool's functionality (Luong et al., 2011), and the semantic annotation of bibliographic citations through the use of CiTalO.

5.1.2 KPEX

The numerous uses of keyphrases are well-established. They range from text summarisation, indexing and query refinement to document clustering and similarity computation, making use of both the descriptive and discriminating qualities of keyphrases. There is also a matching number of approaches that have been proposed for keyphrase extraction, but the percentage of showcased real-world applications that employ them is less than 20%, according to this dissertation's literature review. This is likely a deterring factor in the algorithms' adoption by the community, as their real-world performance remains often questionable. More practical applications would be beneficial in alleviating this problem.

The initial goal of the KPEX system was to adequately extract a small set of phrases from the text of an article, that best characterised its content. This was meant to help readers quickly discern if certain articles were worth further inspection. Such a use case was captured in ScienceWISE's employment of KPEX for suggesting bookmarking terms for articles. The procedure assisted in the management of personal article collections, in which articles were stored to be read and referenced at a later time.

Through its functionality and configuration options, the system can also conduct terminology extraction, abbreviation identification and re-ranking with respect to external lists. The latter facility is an easy way to conduct, for example, ranked ontology term recognition, as the final KPEX ordering can be limited to just a set of relevant concepts.

Through the ScienceWISE platform, KPEX is also being used to help enrich ontologies

[8]DoCO, the Document Components Ontology – `http://www.purl.org/spar/doco/`

of scientific concepts. Given that KPEX was deemed to perform satisfactorily even on the mature physics ontology, it is now being employed in the population of a new life sciences ontology, with computer science and digital humanities scheduled to follow.

As an additional exemplification of KPEX's capabilities, Table 5.1 presents the top-10 chapter-wise keyphrases extracted from the PDFX output for this dissertation. While longer manuscripts are not the type of documents for which PDFX was designed, the structure of academic dissertations bares enough resemblance to the typical research article to foster an adequate conversion. In this case, PDFX had difficulty with the dissertation's front matter and chapter headings, but much of the logical structure was properly recovered. Being that the structure of individual chapters was different than that of the typical research article, the only elements emphasised when running KPEX were section headings. These were given a weight factor of 2.

The substantial richness of this research's application domain has also prompted discussions on article/author proximity and semantic publishing, having high-quality keyphrases as a basis. Noteworthy examples regarding scientific conference management and writing support are given below.

Reviewer assignment to conference submissions (Hettich and Pazzani, 2006; Conry et al., 2009; Charlin et al., 2012). This process could commence by deriving reviewer expertise (in the form of keyphrases), either from his or her own publications, or from other papers exemplar of the reviewer's competence. Simple reviewer assignment could then be done by matching the keyphrases of the submission with this derived expertise. The alternative would be to treat the task as an optimisation problem and attempt to maximise the overlap between the keyphrases of a submission and the joint expertise of e.g. 3 reviewers. This approach would favour the coverage of multiple technical aspects of a paper during the review – a common issue when dealing with multidisciplinary submissions.

Conference programme generation, given the similarities between accepted papers and scheduling constraints (Eglese and Rand, 1987; Thompson, 2002; Sampson, 2004). This would constitute another optimisation problem with the goal of maximising the keyphrase overlap between presentations scheduled in the same session.

Section-wise related literature recommendation. Such a service could be used to suggest related reading material in support of content comprehension, or relevant works to consider referencing, when a paper is being written.

Table 5.1: The top-10 chapter-wise keyphrases extracted by KPEX from the PDFX output for this dissertation.

Chapter 1: Introduction	Chapter 2: Keyphrases
Portable Document Format (PDF)	keyphrases
keyphrase extraction	keyphrase extraction
structure recovery	keyphrase features
keyphrase extraction algorithm	document
conventional article layout	extracted keyphrases
logical structure recovery	information needs
thesis outline	keyphrase quality
text mining	search engines
freely-available PDF conversion service	human
human readers	human choices for keyphrases

Chapter 3: Structure Analysis	Chapter 4: Keyphrase Extraction
PDFX	keyphrase extraction
optical character recognition (OCR)	keyphrases
body text	KPEX
top-level headings	extraction
logical structure	keyphrase candidates
PDF	term frequency (TF)
section headings	inverse document frequency (IDF)
structure recovery	keyphrase extraction system
XML	gold-standard keyphrases
PDF articles	logical regions

Chapter 5: Applications	Chapter 6: Conclusions
logical structure	keyphrase extraction
PDFX	structure analysis
geometrical and logical structures	keyphrase
revised SemEval-2010 evaluation	future directions
typical research article	PDF object model
PDF articles	term weighting scheme
screen readers	better term variant conflation
structure recovery	closing remarks
KPEX	KPEX system
keyphrases	large-scale format migration process

Lastly, as a related practical note on writing support, running KPEX over this dissertation also facilitated identifying hyphenation and casing inconsistencies in the text. The system identified as equivalent term variants e.g. '*gold standard*' and '*gold-standard*' or '*C-Value*' and '*C-value*'. Inconsistent use of typography or character encodings is unfortunately not uncommon in electronic documents. Term extraction solutions that

take this into consideration therefore prove useful in multiple settings.

5.2 Availability

LaTeX sources for the tables and formulae presented in this dissertation, as well as other referenced materials, were attached to the original PDF submission of the manuscript, as individual files. These attachments can be accessed through standard PDF viewers such as Adobe Reader or Evince.

5.2.1 PDFX

PDFX is available at `http://pdfx.cs.man.ac.uk/` as an interactive web page and free-to-use programmatic web service. Submitted PDF articles are processed in real-time, the user being given three options of interacting with the output:

- Access or retrieve the generated XML version.
- View a reconstruction of the article in HTML form, using the generated XML. The core content of the original article is presented as a single-column stream of text, free from elements such as headers, footers or side notes, with figures and tables placed to the side.
- Download an archive containing the entire output, including rendered images, for offline viewing.

Input and output files for each processing job are stored for 24 hours since the time of submission, under randomly-generated URLs.

The following is a list of resources relevant to the logical structure recovery topic, available either as attachments or online:

- Table 3.1 – The logical elements that PDFX can identify.
- Table 3.2 – Existing PDF structure recovery tools.
- Table 3.3 – PDFX's element identification sequence.
- Table 3.4 – Demographics of the datasets used in PDFX's evaluation.
- Table 3.5 – PDFX results over all four datasets.

- Other PDFX-related resources can be found at `http://pdfx.cs.man.ac.uk/usage`.

Should the electronic copy of this thesis not contain the attachments, please refer to the University of Manchester's eScholar service (`https://www.escholar.manchester.ac.uk/`) for the original PDF version.

5.2.2 KPEX

The KPEX keyphrase extractor is currently integrated into the ScienceWISE platform's article bookmarking interface. For the life sciences branch of the platform (`http://bio.sciencewise.info/`), the extractor's functionality is available to all users, as an aid in the initial population of the ontology. The top-10 KPEX-extracted terms will appear in the 'Possible Concepts' column of the bookmarking interface of any article. The physics branch will be next to make KPEX available to all users, followed by the computer science and digital humanities branches.

The LaTeX files and resources related to KPEX that were used in this dissertation are the following:

- Table 4.1 – Features used by SemEval-2010 participants.
- Table 4.2 – Features used by KPEX.
- Table 4.3 – The POS tags used by KPEX.
- Table 4.4 – Statistics for the keyphrases of the FLoC-2010 conference.
- Table 4.5 – Most common POS tags for FLoC-2010 keyphrases.
- Table 4.6 – Percentages of keyword versus keyphrase usage across knowledge bases.
- The original C-value formula.
- The modified C-value formula used by KPEX.
- Formulae for the original SemEval-2010 performance evaluation metrics.
- Formulae for the revised SemEval-2010 evaluation metrics.
- Table 4.10 – Demographics of the SemEval-2010 datasets.

- Table 4.11 – The original SemEval-2010 results.
- Table 4.12 – The revised SemEval-2010 results.
- The revised SemEval-2010 evaluation procedure (gold-standard and script). This data is attached as a .zip file. The Evince PDF viewer can readily access it. Acrobat Reader's security constraints forbid access to this type of attachments by default. The behaviour is customisable through Adobe's `BuiltInPermList` setting. Alternatively, the archive is available at `http://pdfx.cs.man.ac.uk/serve/SemEval_2010_Task_5_revised.zip`.
- Table 4.13 – KPEX results over the SemEval Test dataset using the revised procedure.
- Table 4.14 – KPEX results in the ScienceWISE experiment.

The two document processing solutions were made available at the locations mentioned for non-commercial use.

Chapter 6

CONCLUSIONS

The research described in this dissertation has addressed two challenging topics of particular importance in today's data-riddled research landscape: the logical structure recovery of scientific publications and the extraction of keyphrases from their narratives. Contrary to the amount of published research on these topics, there is still very high demand for robust and easily integratable solutions. A general lack of access to these complementary descriptions of articles has been a resilient bottleneck in many analysis workflows. This final chapter will aim to conclude this work with a summary of the methodology employed herein for addressing these two issues, in comparison to those of existing efforts. The summary will cover the contributions made, the significance of the obtained results, highlighted limitations, gained insights, and an outlook on future directions of research.

6.1 Structure Analysis of Scientific Articles

The structure of a scientific publication carries much implied rhetoric because of its disciplined, modular structure, in effect evolved from the requirement to be peer reviewed. The scholarly article has thus developed well-understood emphasis cues that help readers navigate its content and also steer their attention to elements of particular interest. Access to this semantic information is of great importance for text mining tools, as it conveys a substantial advantage in the provision of more intelligent, time

saving services to users. However, this feature has not yet seen the widespread adoption that its benefits seem to warrant. The main reason for this is a recurring technological setback: most scientific publications do not come in the computationally amenable format required by text processing solutions. This is true even for the field of physics, where authors are believed to use arXiv.org consistently for depositing their preprints in LaTeX format. A recent one year study has concluded that arXiv.org can be said to provide comprehensive coverage only for sub-fields representing at most 20% of all physics (Ingoldsby, 2009). The most widespread publication format remains the PDF – a format that is optimised for viewing and printing, rather than for programmatic access. It is therefore notorious for the hardships involved in attempting to extract coherent content from it automatically.

As was summarised in Table 3.2, very few PDF structure recovery solutions are available. Fewer still attempt to recover an article's *logical* structure in terms of title, sections, references, etc. None, however, have thus far proven versatile enough to readily extract fine-grained structures from multiple article layouts, such as are customarily found in personal collections or online repositories. A novel and straightforward solution called PDFX was afterwards presented in Section 3.2, in support of this need. The approach does not require any prior information of the article's layout. Instead, it relies on rules that exploit generic conventions of scientific literature, prevalent across publishers and disciplines. These conventions regard characteristics that some elements should have in relation to others, for example that section headings should always have a distinguishable emphasis from the body text (in font size or face), consistent throughout the article. The careful instrumentation of such rules rendered the method independent of the hard-coded parameters of journal templates, and of the requirement for a priori model training.

The level of detail of the recovered structure is matched only by a state-of-the-art machine learning solution which PDFX was found to rival in a comparative evaluation. Additionally, PDFX was also found to produce promising results over three other datasets, much larger in size and diversity than have been previously evaluated in related work. This entails that the rule sets used in the approach offer valuable insight into key dependencies between articles' logical units of discourse, that can be used for structure recovery. The PDFX implementation is currently unique in its capability of conducting fine-grained, layout-independent logical structure recognition with such accuracy. Having been made freely-available as a web service, it has garnered much

attention and support, as Section 5.1 has evidenced, and promises to retain an active interest from the community in the foreseeable future.

6.2 Keyphrase Extraction

In contrast to document structure analysis, an ample amount of research on the topic of keyphrase extraction has been documented. This brings up the question of existing implementations of these solutions and of their actual uses in practice, because, on a relative scale, these seem to be quite scarce. Noticeable factors that seem to have deterred the widespread adoption of approaches can be summarised as follows.

First, many of the published evaluations of algorithms are limited to only ad-hoc tests against the choices of humans, lacking convincing evidence, in terms of real-world performance, to encourage their adoption by the community. Chapter 2 has argued that the human judgement of keyphrase quality is an inherently subjective measurement. It should not be trusted to provide conclusive evidence, on its own, of the true proficiency of a method. This dissertation's literature review on keyphrase extraction has comprised 172 articles discussing different approaches, with varying degrees of success and applicability. 75% of them do not report having a precise purpose in mind for the extracted terms. Consequently, they do not showcase concrete information services that have a visible benefit from using the proposed solutions over others.

The assessment of keyphrase quality will always be dependent on the precise application in mind for a particular set of keyphrases. Evaluations over corpora with manually-assigned terms are practical for benchmarking purposes against other systems. However, it is the application context that is arguably the best setting for conducting a meaningful assessment of a method, because of the availability of clear performance indicators to measure. For these reasons, this work has evaluated its developed KPEX system in two separate settings: first with an established benchmark – the SemEval-2010 procedure detailed in Section 4.3.1, and second on two real-world use cases within the collaborative article management platform ScienceWISE (Section 4.3.4).

A second shortcoming of previous efforts is that many of the keyphrase features that have been proposed come from observations of results made in restricted, niche environments. Without the availability of live services that users can experiment with,

sound theoretical baselines are usually required to draw in supporters for new approaches. However, in the practical field of text mining, little attention has been given to underlying theoretical considerations on what makes a keyphrase and what people perceive keyphrases to be. Such knowledge is also an important factor to take into account, if next generation tools are to break the apparent performance plateau that has been reached[1]. This work has covered this theoretical aspect as well, by providing an examination of the concept of keyphrase in Chapter 2. This process has, in turn, highlighted what seem to be influential factors on the human perception of term relevance. The main discovered factor, rhetorical structure information, was integrated in the KPEX system, along with re-engineered preprocessing steps and an improvement of an existing term weighting scheme, C-value (please see Section 4.2.4).

Lastly, it may be that the complexity involved in certain implementations has also been a discouraging factor of the tools' adoption. A conclusion of the SemEval-2010 competition (Kim et al., 2010) was that the field of keyphrase extraction still has much room for improvement, despite many intricate features having been introduced, such as the derivation of *keyphraseness* metrics from third-party thesauri or ontologies of scientific concepts. In contrast, the overall simplistic set of features used by KPEX was enough to achieve top-ranking results in the SemEval-2010 benchmark. An important remark highlighted in the experiment was that such promising results were achieved without model training or collection statistics, and without the need to consult external knowledge sources that covered the analysed domain. In the first instance, this finding attests that KPEX, in conjunction with PDFX, provides a better and more applicable keyphrase extraction solution than the current state-of-the-art. In addition, the performance was obtained without considering terms from the core body text, captions and conclusions of articles. This provided an insightful look at how content deemed relevant is distributed across the different logical units of scientific articles.

The above factors stand out as decisive in the endorsement and utilisation of novel tools, hence also for further research and advancements in the field. This dissertation has aimed to cover all these aspects throughout Chapters 2 and 4.

While KPEX achieved very good results in the SemEval benchmark, a setting in which

[1]The original SemEval-2010 leader had an F_1 score of just 26.5, out of 19 participating systems (please see Table 4.12 for details).

its functionality was not sufficient has also been highlighted, in the ScienceWISE experiment. Nearly half of all KPEX-extracted keyphrases were valid ontological concepts, and 59% of these were chosen for bookmarking on average. Out of the novel keyphrases extracted however, only 8.4% were chosen. This constituted a scenario in which a curated knowledge base proved valuable, as an automatic keyphrase extractor could not function as a replacement for it. However, in a fast-paced research environment, there is also a need for services to be able to integrate new knowledge quickly and efficiently. In the ontology enrichment use case, KPEX proved its worth, by either helping users to identify new relevant concepts, or alternative forms for existing ontological entries. A synergy of the two paradigms thus seems to work best in this setting: solutions such as KPEX unrestrictedly extract salient terms from documents, and collective knowledge helps put these into context, for an overall better appreciation of the works' true contributions.

6.3 Future Directions

The several other real-world applications in which the described approaches have been integrated were described in Chapter 5. These, along with the many more enabled possibilities for added-value information services, warrant the further development of the products of this research. The following are listings that cover planned future work.

Structure Analysis

- Rendering of the most likely region of a vector graphics figure. PDFX does not currently handle non-bitmap images, but similarly to the approach used for bounding a table region, the area where the figure is likely to be can be approximated and rendered. Given the high accuracy achieved for figure captions, and with positioning information for any neighbouring regions, outlining these elements should be relatively straightforward.
- Improvement of the procedure to identify the most likely tabular arrangement of a set of words. This can be done by trying out different algorithms for table structure recognition, such as the ones presented in Zanibbi et al. (2003). Better table recognition could also be achieved if vector graphics information will be accessible from the PDF object model, as delineated headers, rows and columns will be more easily distinguished.

- Reiteration of the identification steps that are not time consuming. Table 3.3 has presented the relative times spent by PDFX for identifying each element type. The identification of elements such as the abstract, bibliography and tables is a relatively quick process. Therefore, it could be reiterated at the end of the system's element identification sequence, to take advantage of any new information gained in the interim.

- Ability to plug-in different physical structure extractors. PDFX's current procedure relies on a component from Utopia Documents to access the PDF object model, and recover information about pages, words, fonts and bitmap images. The whole system's functionality would be applicable also to legacy article images if, for example, an OCR solution would be substituted to provide the required information.

- Better handling of front matter elements and lower-level headings. Author affiliations often share stylistic characteristics with neighbouring author regions and end up being incorrectly merged with these. Their identification can be facilitated by the use of lists of countries and cities, academic departments, and institutions. Additionally, finer-grained stylistic differences between neighbouring words, in conjunction with positioning statistics, can foster the better identification of lower-level section headings.

- Identification of additional elements such as superscripts, bulleted lists and enumerations.

- The automatic detection of a document's language, to choose appropriate cue lists and be readily applicable to diverse languages.

Keyphrase Extraction

- Better conflation of semantically equivalent terms. As highlighted in the discussion of KPEX's results, a hindering factor of the system is a limited ability to identify different textual representations of the same concept. Better term variant conflation would lead to better results.

- The ability to plug in different term weighting schemes. KPEX currently uses a modified C-value algorithm for term ranking. The implementation could be extended to allow the specification of an external process to conduct this step, as a command-line argument. This feature would facilitate, for example, a TF-IDF

implementation running over the keyphrase candidates identified by KPEX, after all the preprocessing steps described in Section 4.2.

- Experimentation with aggregating the top keyphrases of each section of an article, in hopes of achieving better content coverage.
- A consideration of an article's keyphrases in relation to those of the cited literature. PDFX could be used to identify the references of the article being analysed, along with their DOIs (via the CrossRef service). If retrievable, these documents could also be mined for keyphrases and compared against those of the citing article. An intersection between the keyphrase sets might help identify coarse topics, such as the field of study or the general methodology used, whilst a difference between them could contribute in identifying the citing article's own contributions.

6.4 Closing Remarks

Recent years have seen the emergence of new initiatives aimed at reshaping the realm of digital publishing, in response to changing technological and sociological needs (Bourne et al., 2012; Dewan, 2012). Inherent in this, is the issue of a format shift away from the prevailing PDF, a shift that seems to now be more a matter of time than of possibilities. Whatever the new format of choice will be, huge catalogues of legacy material available solely in PDF form will need to be aligned with this next generation. Versatile conversion solutions such as the one of this dissertation will likely be relied upon to kickstart this large-scale format migration process.

Once a more structured representation of research data will be available, it will be left with analysis, transformation and reasoning tools to acquire and make use of new knowledge. Keyphrase extractors like the one also presented here, are a foundational step in this direction, providing a first level of insight into the contents and contributions of scientific publications.

More work is needed until the described approaches can be deemed to have reached their full potential, but even in their current state, they have opened up numerous new opportunities for knowledge discovery. Through their integration within the processing workflows of a number of institutions, the solutions already promise to have a sizeable contribution to scientific advancement.

Bibliography

Aberer, K., A. Boyarsky, P. Cudré-Mauroux, G. Demartini, and O. Ruchayskiy (2011). Sciencewise: a web-based interactive semantic platform for scientific collaboration. International Semantic Web Conference (ISWC) (Demonstration Track).

Abulaish, M. and T. Anwar (2012). A supervised learning approach for automatic keyphrase extraction. Int'l J. of Inn. Comp. Inf. and Ctrl.(IJICIC) 8(11).

Aho, A. V. and M. J. Corasick (1975). Efficient string matching: an aid to bibliographic search. In Communications of the ACM, Volume 18, pp. 333–340. ACM.

Ananiadou, S., S. Albert, and D. Schuhmann (2000). Evaluation of automatic term recognition of nuclear receptors from medline. Genome Informatics Series 11, 450–451.

Anderson, J. D. and J. Pérez-Carballo (2001). The nature of indexing: how humans and machines analyze messages and texts for retrieval. part i: Research, and the nature of human indexing. Information Processing & Management 37(2), 231–254.

Antoniou, M., L. Harland, T. Mustoe, S. Williams, J. Holdstock, E. Yague, T. Mulcahy, M. Griffiths, S. Edwards, P. A. Ioannou, et al. (2003). Transgenes encompassing dual-promoter cpg islands from the human tbp and hnrpa2b1 loci are resistant to heterochromatin-mediated silencing. Genomics 82(3), 269–279.

Attwood, T., D. Kell, P. McDermott, J. Marsh, S. Pettifer, and D. Thorne (2009). Calling international rescue: knowledge lost in literature and data landslide! Biochem. J 424, 317–333.

Attwood, T. K., D. B. Kell, P. McDermott, J. Marsh, S. Pettifer, and D. Thorne (2010). Utopia documents: linking scholarly literature with research data. Bioinformatics 26(18), i568–i574.

Bahrani, B. and M.-Y. Kan (2013). Multimodal alignment of scholarly documents and their presentations. In Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries, pp. 281–284. ACM.

Bairoch, A. (2009). The future of annotation/biocuration. Nature Precedings (713).

Barker, K. and N. Cornacchia (2000). Using noun phrase heads to extract document keyphrases. Advances in Artificial Intelligence 1822, 40–52.

Barriere, C. and M. Jarmasz (2004). Keyphrase extraction: enhancing lists. In Proceedings of the Computational Linguistic in the North- East, Volume 48079.

Baruch, P. (2007). Open access developments in france: the hal open archives system. Learned Publishing 20(4), 267–282.

Beel, J., S. Langer, M. Genzmehr, B. Gipp, C. Breitinger, and A. Nürnberger (2013). Research paper recommender system evaluation: a quantitative literature survey. In Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation, pp. 15–22. ACM.

Berend, G. and R. Farkas (2010). Sztergak: Feature engineering for keyphrase extraction. In Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 186–189. Association for Computational Linguistics.

Berg, Ø. R. (2011). High precision text extraction from pdf documents.

Berg, Ø. R., S. Oepen, and J. Read (2012). Towards high-quality text stream extraction from pdf: technical background to the acl 2012 contributed task. In Proceedings of the ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries, pp. 98–103. Assoc. for Computational Linguistics.

Bishop, A. P., L. J. Neumann, S. L. Star, C. Merkel, E. Ignacio, and R. J. Sandusky (2000). Digital libraries: Situating use in changing information infrastructure. Journal of the American Society for Information Science 51(4), 394–413.

Bordea, G. and P. Buitelaar (2010). Deriunlp: A context based approach to automatic keyphrase extraction. In Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 146–149. Association for Computational Linguistics.

Bourne, P. E., T. Clark, R. Dale, A. De Waard, I. Herman, E. Hovy, D. Shotton, et al. (2012). Improving future research communication and e-scholarship: a summary of findings— macquarie university researchonline. Informatik-Spektrum 35(1), 55–62.

Boyarsky, A., O. Ruchayskiy, D. Iakubovskyi, and J. Franse (2014). An unidentified line in x-ray spectra of the andromeda galaxy and perseus galaxy cluster. arXiv preprint arXiv:1402.4119.

Boyarsky, A., O. Ruchayskiy, Z. Yang, O. Zozulya, M. Charlaganov, and P. De Los Rios (2012). From scientific papers to the scientific ontology: dynamical clustering of heterogeneous graphs and ontology crowdsourcing. Joint Workshop on Large and Heterogeneous Data and Quantitative Formalization in the Semantic Web.

Brill, E. (1992). A simple rule-based part of speech tagger. In Proceedings of the workshop on Speech and Natural Language, pp. 112–116. Association for Computational Linguistics.

Brin, S. and L. Page (1998). The anatomy of a large-scale hypertextual web search engine. Computer networks and ISDN systems 30(1), 107–117.

Brinton, L. J. (2000). The structure of modern English: a linguistic introduction. John Benjamins.

Burnham, J. F. (2006). Scopus database: a review. Biomedical digital libraries 3(1), 1.

Charlin, L., R. S. Zemel, and C. Boutilier (2012). A framework for optimizing paper matching. arXiv preprint arXiv:1202.3706.

Church, K. W. and P. Hanks (1990). Word association norms, mutual information, and lexicography. Computational linguistics 16(1), 22–29.

Ciancarini, P., A. Di Iorio, A. G. Nuzzolese, S. Peroni, and F. Vitali (2013). Semantic annotation of scholarly documents and citations. In AI* IA 2013: Advances in Artificial Intelligence, pp. 336–347. Springer.

Conry, D., Y. Koren, and N. Ramakrishnan (2009). Recommender systems for the conference paper assignment problem. In Proceedings of the third ACM conference on Recommender systems, pp. 357–360. ACM.

Constantin, A., S. Pettifer, and A. Voronkov (2013). Pdfx: fully-automated pdf-to-xml conversion of scientific literature. In Proceedings of the 2013 ACM symposium on Document engineering, pp. 177–180. ACM.

Cooper, W. S. (1978). Indexing documents by gedanken experimentation. Journal of the American Society for Information Science 29(3), 107–119.

Corty, M. (2010). Niche markets to boost reed elsevier sales. Published online. [Link].

Councill, I. G., C. L. Giles, and M.-Y. Kan (2008). Parscit: An open-source crf reference string parsing package. In Proceedings of LREC, Volume 2008, pp. 661–667. European Language Resources Association (ELRA).

Croft, W. B. (1986). User-specified domain knowledge for document retrieval. In Proceedings of the 9th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 201–206. ACM.

Croft, W. B. and D. J. Harper (1979). Using probabilistic models of document retrieval without relevance information. Journal of documentation 35(4), 285–295.

Csomai, A. and R. Mihalcea (2008). Linguistically motivated features for enhanced back-of-the-book indexing. Proceedings of ACL-08: HLT, 932–940.

Czoski-Murray, C., M. L. Jones, C. McCabe, K. Claxton, Y. Oluboyede, J. Roberts, J. Nicholl, A. Rees, C. Reilly, D. Young, et al. (2012a). What is the value of routinely testing full blood count, electrolytes and urea, and pulmonary function tests before elective surgery in patients with no apparent clinical indication and in subgroups of patients with common comorbidities: a systematic review of the clinical and cost-effective literature: Appendix 1: Systematic review of clinical effectiveness: Medline search strategies. Health Technol Assess 16(50), 91–93.

Czoski-Murray, C., M. L. Jones, C. McCabe, K. Claxton, Y. Oluboyede, J. Roberts, J. Nicholl, A. Rees, C. Reilly, D. Young, et al. (2012b). What is the value of routinely testing full blood count, electrolytes and urea, and pulmonary function tests before elective surgery in patients with no apparent clinical indication and in subgroups of patients with common comorbidities: a systematic review of the clinical and cost-effective literature: Appendix 11: Search strategies for cost-effectiveness review. Health Technol Assess 16(50), 119–124.

Déjean, H. and J.-L. Meunier (2006). A system for converting pdf documents into structured xml format. In Document Analysis Systems VII, pp. 129–140. Springer.

Dewan, P. (2012). Are books becoming extinct in academic libraries? New Library World 113(1/2), 27–37.

Di Iorio, A., A. G. Nuzzolese, and S. Peroni (2013). Towards the automatic identification of the nature of citations. In 3rd Workshop on Semantic Publishing (SePublica 2013) 10 th Extended Semantic Web Conference Montpellier, France, 26 May 2013, pp. 63.

Dillon, A. (1991). Readers' models of text structures: the case of academic articles. International Journal of Man-Machine Studies 35(6), 913–925.

Dillon, A. (2000). Spatial-semantics: How users derive shape from information space. Journal of the American Society for Information Science 51(6), 521–528.

Dinh, D., L. Tamine, and F. Boubekeur (2012). Factors affecting the effectiveness of biomedical document indexing and retrieval based on terminologies. Artificial Intelligence in Medicine 57(2), 155–167.

Domingos, P. and M. Pazzani (1997). On the optimality of the simple bayesian classifier under zero-one loss. Machine learning 29(2-3), 103–130.

Eglese, R. and G. Rand (1987). Conference seminar timetabling. Journal of the Operational Research Society 38(7), 591–598.

El-Beltagy, S. and A. Rafea (2009). Kp-miner: A keyphrase extraction system for english and arabic documents. Information Systems 34(1), 132–144.

El-Beltagy, S. and A. Rafea (2010). Kp-miner: Participation in semeval-2. In Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 190–193. Association for Computational Linguistics.

Ercan, G. and I. Cicekli (2007). Using lexical chains for keyword extraction. Information Processing & Management 43(6), 1705–1714.

Esposito, F., S. Ferilli, T. M. Basile, and N. Di Mauro (2008). Machine learning for digital document processing: from layout analysis to metadata extraction. In Machine learning in document analysis and recognition, pp. 105–138. Springer.

Esposito, F., D. Malerba, and G. Semeraro (1995). A knowledge-based approach to the layout analysis. In Proceedings of the Third International Conference on Document Analysis and Recognition, 1995, Volume 1, pp. 466–471. IEEE.

Ferilli, S., M. Biba, T. M. A. Basile, and F. Esposito (2009). Combining qualitative and quantitative keyword extraction methods with document layout analysis. In IRCDL, pp. 22–33.

Fox, E. A. (1980). Lexical relations: Enhancing effectiveness of information retrieval systems. In ACM SIGIR Forum, Volume 15, pp. 5–36. ACM.

Frank, E., G. Paynter, I. Witten, C. Gutwin, and C. Nevill-Manning (1999). Domain-specific keyphrase extraction. In Proceeding of 16th International Joint Conference on Artificial Intelligence, Stockholm, Sweden, pp. 668–673. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. [Link].

Frantzi, K., S. Ananiadou, and H. Mima (2000). Automatic recognition of multi-word terms: the c-value/nc-value method. International Journal on Digital Libraries 3(2), 115–130.

Herings, P., G. Van der Laan, and D. Talman (2001). Measuring the power of nodes in digraphs.

Hettich, S. and M. J. Pazzani (2006). Mining for proposal reviewers: lessons learned at the national science foundation. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 862–871. ACM.

Hjørland, B. (2002a). Domain analysis in information science: eleven approaches–traditional as well as innovative. Journal of documentation 58(4), 422–462.

Hjørland, B. (2002b). Epistemology and the socio-cognitive perspective in information science. Journal of the American Society for Information Science and Technology 53(4), 257–270.

Hjørland, B. and L. K. Nielsen (2001). Subject access points in electronic retrieval. Annual Review of lnformation science and Technology (ARIST) 35.

Hofmann, K., M. Tsagkias, E. Meij, and M. De Rijke (2009). The impact of document structure on keyphrase extraction. In Proceeding of the 18th ACM conference on Information and knowledge management, pp. 1725–1728. ACM.

Hofmann, K., M. Tsagkias, E. Meij, and M. de Rijke (2010). A comparative study of features for keyphrase extraction in scientific literature. Available from `http://edgar.meij.pro/wp-content/papercite-data/pdf/cikm-2009-hofmann.pdf`.

Hollingsworth, B., I. Lewin, and D. Tidhar (2005). Retrieving hierarchical text structure from typeset scientific articles–a prerequisite for e-science text mining. In Proc. of the 4th UK E-Science All Hands Meeting, pp. 267–273.

Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge. In Proceedings of the 2003 conference on Empirical methods in natural language processing, pp. 216–223. Association for Computational Linguistics.

Hulth, A. (2004). Combining machine learning and natural language processing for automatic keyword extraction. Ph. D. thesis, Deparment of Computer and Systems Sciences, Stockholm University.

Hulth, A., J. Karlgren, A. Jonsson, H. Boström, and L. Asker (2001). Automatic keyword extraction using domain knowledge. Computational Linguistics and Intelligent Text Processing 2004, 472–482.

Ingoldsby, T. (2009). Physics journals and the ArXiv: What is myth and what is reality? Technical report, American Institute of Physics.

Iwasaki, W., Y. Yamamoto, and T. Takagi (2010). Togodoc server/client system: smart recommendation and efficient management of life science literature. PloS one 5(12), e15305.

Jacquemin, C. and D. Bourigault (2003). Term extraction and automatic indexing, pp. 599–615. Oxford University Press.

Jones, K. S. (1971). Automatic keyword classification for information retrieval.

Joorabchi, A. and A. E. Mahdi (2013). Automatic keyphrase annotation of scientific documents using wikipedia and genetic algorithms. Journal of Information Science 39(3), 410–426.

Justeson, J. S. and S. M. Katz (1995). Technical terminology: some linguistic properties and an algorithm for identification in text. Natural language engineering 1(1), 9–27.

Katayama, T., M. D. Wilkinson, K. F. Aoki-Kinoshita, S. Kawashima, Y. Yamamoto, A. Yamaguchi, S. Okamoto, S. Kawano, J.-D. Kim, Y. Wang, et al. (2014). Biohackathon series in 2011 and 2012: penetration of ontology and linked data in life science domains. Journal of Biomedical Semantics 5(1), 5.

Kim, J., D. X. Le, and G. R. Thoma (2001). Automated labeling in document images. In Proc. SPIE: Document Recognition and Retrieval VIII, Volume 4307, pp. 111–22.

Kim, S. and M. Kan (2009). Re-examining automatic keyphrase extraction approaches in scientific articles. In Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications, pp. 9–16. Association for Computational Linguistics.

Kim, S., O. Medelyan, M. Kan, and T. Baldwin (2010). Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 21–26. Association for Computational Linguistics.

Kiss, T. and J. Strunk (2006). Unsupervised multilingual sentence boundary detection. Computational Linguistics 32(4), 485–525.

Kleinberg, J. M., R. Kumar, P. Raghavan, S. Rajagopalan, and A. S. Tomkins (1999). The web as a graph: Measurements, models, and methods. In Computing and combinatorics, pp. 1–17. Springer.

Klingbiel, P. H. (1973). A technique for machine-aided indexing. Information Storage and Retrieval 9(9), 477–494.

Krauthammer, M. and G. Nenadić (2004). Term identification in the biomedical literature. Journal of biomedical informatics 37(6), 512–526.

Lazar, J., A. Allen, J. Kleinman, and C. Malarkey (2007). What frustrates screen reader users on the web: A study of 100 blind users. International Journal of human-computer interaction 22(3), 247–269.

Lei, K., H. Tang, and Y. Zeng (2013). Keywords extraction via multi-relational network construction. In Advances in Computational Science, Engineering and Information Technology, pp. 33–45. Springer.

Lesk, M. E. (1969). Word-word associations in document retrieval systems. American documentation 20(1), 27–38.

Lewis, D. D. (1995). Evaluating and optimizing autonomous text classification systems. In Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 246–254. ACM.

Liakata, M., S. Saha, S. Dobnik, C. Batchelor, and D. Rebholz-Schuhmann (2012). Automatic recognition of conceptualization zones in scientific articles and two life science applications. Bioinformatics 28(7), 991–1000.

Liu, F., Y. Liu, C. Busso, S. Harabagiu, and V. Ng (2011). Identifying the Gist of Conversational Text: Automatic Keyword Extraction and Summarization. Ph. D. thesis, The University of Texas as Dallas.

Liu, M., R. Calvo, A. Aditomo, and L. Pizzato (2012). Using wikipedia and conceptual graph structures to generate questions for academic writing support.

Lopez, P. and L. Romary (2010). Humb: Automatic key term extraction from scientific articles in grobid. In Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 248–251. Association for Computational Linguistics.

Lopez, P., L. Romary, et al. (2010). Grisp: A massive multilingual terminological database for scientific and technical domains. In LREC 2010.

Louis, A. and A. Nenkova (2011). Automatic identification of general and specific sentences by leveraging discourse annotations. In Proc. of IJCNLP, pp. 605–613.

Lourenço, A., R. Carreira, D. Glez-Peña, J. R. Méndez, S. Carneiro, L. M. Rocha, F. Díaz, E. C. Ferreira, I. Rocha, F. Fdez-Riverola, et al. (2010). Biodr: Semantic indexing networks for biomedical document retrieval. Expert Systems with Applications 37(4), 3444–3453.

Luong, M.-T., T. D. Nguyen, and M.-Y. Kan (2011). Logical structure recovery in scholarly articles with rich document features. J. of Digital Library Systems. Forthcoming.

MacGregor, J., K. Stranack, and J. Willinsky (2014). The public knowledge project: Open source tools for open access to scholarly communication. In Opening Science, pp. 165–175. Springer.

Magnini, B. and G. Cavaglia (2000). Integrating subject field codes into wordnet. In LREC.

Mao, S., A. Rosenfeld, , and T. Kanungo (2003). Document structure analysis algorithms: a literature survey. In Proc. SPIE Electronic Imaging, Volume 5010, pp. 197–207.

Marcus, M. P., M. A. Marcinkiewicz, and B. Santorini (1993). Building a large annotated corpus of english: The penn treebank. Computational linguistics 19(2), 313–330.

Medelyan, O. (2009). Human-competitive automatic topic indexing. Ph. D. thesis, The University of Waikato.

Medelyan, O., I. Witten, and D. Milne (2008). Topic indexing with wikipedia. In Proceedings of the AAAI WikiAI workshop, Volume 1, pp. 19–24. AAAI Press.

Mihalcea, R. and P. Tarau (2004). Textrank: Bringing order into texts. In Proceedings of EMNLP, Volume 4. Barcelona: ACL.

Milios, E., Y. Zhang, B. He, and L. Dong (2003). Automatic term extraction and document similarity in special text corpora. In Proceedings of the sixth conference of the pacific association for computational linguistics, pp. 275–284. Citeseer.

Miller, G. A., R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller (1990). Introduction to wordnet: An on-line lexical database*. International journal of lexicography 3(4), 235–244.

Milne, D. and I. H. Witten (2013). An open-source toolkit for mining wikipedia. Artificial Intelligence 194, 222–239.

Nenadić, G., I. Spasić, and S. Ananiadou (2002). Automatic acronym acquisition and term variation management within domain-specific texts. In Third International Conference on Language Resources and Evaluation (LREC2002), pp. 2155–2162.

Nguyen, T. and M. Kan (2007). Keyphrase extraction in scientific publications. In Proceedings of the 10th international conference on Asian digital libraries: looking back 10 years and forging new frontiers, pp. 317–326. Springer-Verlag.

Nguyen, T. and M. Luong (2010). Wingnus: Keyphrase extraction utilizing document logical structure. In Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 166–169. Association for Computational Linguistics.

Park, Y., R. J. Byrd, and B. K. Boguraev (2002). Automatic glossary extraction: beyond terminology identification. In Proceedings of the 19th international conference on Computational linguistics-Volume 1, pp. 1–7. Association for Computational Linguistics.

Patrick Wilson (1968). Two kinds of power: An essay on bibliographic control, Volume 5. University of California Press.

Paukkeri, M.-S., I. T. Nieminen, M. Pöllä, and T. Honkela (2008). A language-independent approach to keyphrase extraction and evaluation. In COLING (Posters), pp. 83–86.

Pentz, E. (2001). Crossref: a collaborative linking network. Issues in science and technology librarianship 29.

Pettifer, S., P. McDermott, J. Marsh, D. Thorne, A. Villeger., and T. K. Attwood (2011). Ceci n'est pas un hamburger: modelling and representing the scholarly article. Learned Publishing 24(3), 207–220.

Pianta, E. and S. Tonelli (2010). Kx: A flexible system for keyphrase extraction. In Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 170–173. Association for Computational Linguistics.

Porter, M. F. (1980). An algorithm for suffix stripping. Program: electronic library and information systems 14(3), 130–137.

Powley, B. and R. Dale (2007). Evidence-based information extraction for high accuracy citation and author name identification. In Large Scale Semantic Access to Content (Text, Image, Video, and Sound), pp. 618–632. Le Centre de Hautes Études Internationales D'informatique Documentaire.

Prokofyev, R., A. Boyarsky, O. Ruchayskiy, K. Aberer, G. Demartini, and P. Cudré-Mauroux (2012). Tag recommendation for large-scale ontology-based information systems. In The Semantic Web–ISWC 2012, pp. 325–336. Springer.

Quinlan, J. R. (1993). C4.5: programs for machine learning, Volume 1. Morgan kaufmann.

Ramakrishnan, C., A. Patnia, E. Hovy, et al. (2012). Layout-aware text extraction from full-text pdf of scientific articles. Source code for biology and medicine 7(1), 1–10.

Ratcliff, J. W. and D. Metzener (1988). Pattern matching: The gestalt approach. Dr. Dobb's Journal, 46.

Ravenscroft, J., M. Liakata, and A. Clare (2013). Partridge: An effective system for the automatic cassification of the types of academic papers. In Research and Development in Intelligent Systems XXX, pp. 351–358. Springer.

Rebholz-Schuhmann, D., H. Kirsch, and F. Couto (2005). Facts from text—is text mining ready to deliver? PLoS Biology 3(2), e65.

Salton, G. (1971). Experiments in automatic thesaurus construction for information retrieval. In IFIP Congress (1), pp. 115–123.

Salton, G. (1975). A theory of indexing. Number 18-22. SIAM.

Salton, G. and C. Buckley (1988). Term-weighting approaches in automatic text retrieval. Information processing and management 24(5), 513–523.

Salton, G. and C. Yang (1973). On the specification of term values in automatic indexing. Journal of documentation 29(4), 351–372.

Salton, G., C.-S. Yang, and C. T. Yu (1975). A theory of term importance in automatic text analysis. Journal of the American Society for Information Science 26(1), 33–44.

Sampson, S. E. (2004). Practical implications of preference-based conference scheduling. Production and Operations Management 13(3), 205–215.

Sánchez, D. and D. Isern (2011). Automatic extraction of acronym definitions from the web. Applied Intelligence 34(2), 311–327.

Sanderson, R., H. Van de Sompel, P. Burnhill, and C. Grover (2013). Hiberlink: towards time travel for the scholarly web. In Proceedings of the 1st International Workshop on Digital Preservation of Research Methods and Artefacts, pp. 21–21. ACM.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In Proceedings of international conference on new methods in language processing, Volume 12, pp. 44–49. Manchester, UK.

Shah, P., C. Perez-Iratxeta, P. Bork, and M. Andrade (2003). Information extraction from full text scientific articles: Where are the keywords? BMC bioinformatics 4(1), 20.

Shotton, D. (2009). Semantic publishing: the coming revolution in scientific journal publishing. Learned Publishing 22(2), 85–94.

Soergel, D. (1994). Indexing and retrieval performance: The logical evidence. JASIS 45(8), 589–599.

Spärck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. Journal of documentation 28(1), 11–21.

Spärck Jones, K. (1983). Intelligent retrieval. In K. Jones (Ed.), Proceedings of Informatics 7, Volume 7, London, pp. 136–142. Aslib.

Spärck Jones, K. and J. I. Tait (1984). Automatic search term variant generation. Journal of Documentation 40(1), 50–66.

Spasić, I., M. Greenwood, A. Preece, N. Francis, and G. Elwyn (2013). Flexiterm: a flexible term recognition method. Journal of biomedical semantics 4(1), 27.

Tenopir, C., D. W. King, S. Edwards, and L. Wu (2009). Electronic journals and changes in scholarly article seeking and reading patterns. In Aslib proceedings, Volume 61, pp. 5–32. Emerald Group Publishing Limited.

Teufel, S. and M.-Y. Kan (2011). Robust argumentative zoning for sensemaking in scholarly documents. Advanced Language Technologies for Digital Libraries 6699, 154–170.

Teufel, S. and M. Moens (2002). Summarizing scientific articles: experiments with relevance and rhetorical status. Computational linguistics 28(4), 409–445.

Thompson, G. M. (2002). Improving conferences through session scheduling. The Cornell Hotel and Restaurant Administration Quarterly 43(3), 71–76.

Torii, M., Z.-z. Hu, M. Song, C. Wu, and H. Liu (2007). A comparison study on algorithms of detecting long forms for short forms in biomedical text. BMC bioinformatics 8(Suppl 9), S5.

Treeratpituk, P., P. Teregowda, J. Huang, and C. Giles (2010). Seerlab: A system for extracting key phrases from scholarly documents. In Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 182–185. Association for Computational Linguistics.

Turney, P. (1997). Extraction of keyphrases from text: evaluation of four algorithms. National Research Council of Canada Publications Archive (NRC/ERB-1051), 1–29.

Turney, P. (1999). Learning to extract keyphrases from text. National Research Council of Canada Publications Archive NRC 41622(NRC/ERB-1057), 1–43.

Turney, P. (2003). Coherent keyphrase extraction via web mining. National Research Council of Canada Publications Archive (NRC 46496), 434–439.

Turney, P. D. (2000). Learning algorithms for keyphrase extraction. Information Retrieval 2(4), 303–336.

van Rijsbergen, C. (1979). Information Retrieval (2nd Edition ed.)., pp. 168–176. London: Butterworths.

van Rijsbergen, C. J. (1977). A theoretical basis for the use of co-occurrence data in information retrieval. Journal of documentation 33(2), 106–119.

Wan, S., C. Paris, and R. Dale (2010). Supporting browsing-specific information needs: Introducing the citation-sensitive in-browser summariser. Web Semantics: Science, Services and Agents on the World Wide Web 8(2), 196–202.

Wan, X. and J. Xiao (2008). Single document keyphrase extraction using neighborhood knowledge. In Proceedings of AAAI, pp. 855–860.

Wang, C. and S. Li (2011). Corankbayes: bayesian learning to rank under the co-training framework and its application in keyphrase extraction. In Proceedings of the 20th ACM international conference on Information and knowledge management, pp. 2241–2244. ACM.

Whitley, L. D. et al. (1989). The genitor algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. In ICGA, Volume 89, pp. 116–123.

Witten, I., G. Paynter, E. Frank, C. Gutwin, and C. Nevill-Manning (1999). Kea: Practical automatic keyphrase extraction. In Proceedings of the fourth ACM conference on Digital libraries, pp. 254–255. ACM.

Wittern, C., A. Ciula, and C. Tuohy (2009). The making of tei p5. Literary and linguistic computing 24(3), 281–296.

Wu, H. and G. Salton (1981). A comparison of search term weighting: term relevance vs. inverse document frequency. In ACM SIGIR Forum, Volume 16, pp. 30–39. ACM.

Xie, L., L. Xie, and P. E. Bourne (2009). A unified statistical model to support local sequence order independent similarity searching for ligand-binding sites and its application to genome-based drug discovery. Bioinformatics 25(12), i305–i312.

Yang, Y., N. Bansal, W. Dakka, P. Ipeirotis, N. Koudas, and D. Papadias (2009). Query by document. In Proceedings of the Second ACM International Conference on Web Search and Data Mining, pp. 34–43. ACM.

Yates, J. and W. J. Orlikowski (1992). Genres of organizational communication: A structurational approach to studying communication and media. Academy of management review 17(2), 299–326.

Yoo, Y.-H. and J.-H. Kim (2013). Mathematical formula recognition based on modified recursive projection profile cutting and labeling with double linked list. In Robot Intelligence Technology and Applications 2012, pp. 983–992. Springer.

You, W., D. Fontaine, and J. Barthès (2012). An automatic keyphrase extraction system for scientific documents. Knowledge and Information Systems 34, 691–724.

Yu, C. T., C. Buckley, K. Lam, and G. Salton (1983). A generalized term dependence model in information retrieval. Technical report, Cornell University.

Zanibbi, R., D. Blostein, and J. R. Cordy (2003). A survey of table recognition: Models, observations, transformations, and inferences. International Journal of Document Analysis and Recognition 7, 1–16.

Zervanou, K. (2010). Uvt: The uvt term extraction system in the keyphrase extraction task. In Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 194–197. Association for Computational Linguistics.

Zervanou, K. and J. McNaught (2004). A domain-independent approach to ie rule development. In LREC.

Zesch, T. and I. Gurevych (2009). Approximate matching for evaluating keyphrase extraction. In Proceedings of the 7th International Conference on Recent Advances in Natural Language Processing, pp. 484–489. Citeseer.

Zhang, L. (2012). Grasping the structure of journal articles: Utilizing the functions of information units. Journal of the American Society for Information Science and Technology 63(3), 469–480.

Zhang, W., W. Feng, and J. Wang (2013). Integrating semantic relatedness and words' intrinsic features for keyword extraction. In Proceedings of the Twenty-Third international joint conference on Artificial Intelligence, pp. 2225–2231. AAAI Press.

Zhang, Y., N. Zincir-Heywood, and E. Milios (2005). Narrative text classification for automatic key phrase extraction in web document corpora. In Proceedings of the 7th annual ACM international workshop on Web information and data management, pp. 51–58. ACM.

Appendix A

PDFX LOG AND OUTPUT EXAMPLE

The following is an example of PDFX's log output for the processing of the biomedical publication of Xie et al. (2009). The sequence of steps followed is the one in Table 3.3, that was also detailed in Section 3.2.3.

```
                              *** BODY ***
Document body font: ('Times-Roman', 9.0)
Document body font not found on page 3/8
Document body font not found on page 8/8
Body font statistics: Occurred in 6/8 pages.
page 3 - Reg set as possible (font diff: 1.0)    - PDB chains used...
page 3 - Reg set as possible (width diff: 216.0) - PDB ID
page 3 - Reg set as possible (width diff: 213.0) - Chain ID
[...]
page 8 - Reg set as possible (font diff: 2.0) - Laskowski,R.A. (20...
page 8 - Reg set as possible (font diff: 2.0) - Siggers, T.W. (200...
                           *** end BODY ***
                             *** IMAGES ***
page 3: 1 (merged) images
page 4: 2 (merged) images
page 5: 2 (merged) images
                         *** end IMAGES ***
                                *** DOI ***
DOI found: 10.1093/bioinformatics/btp220
Meta-info from CrossRef:
> surnames: [Xie, Xie, Bourne]
```

```
> title: A unified statistical model to support local sequence ord...
                          *** end DOI ***
                          *** AUTHORS ***
Found confident author word: Xie (by doi)
Author region(s): Lei Xie 1,* , Li Xie...
                      *** end AUTHORS ***
                            *** TITLE ***
Title font candidate: ('HelveticaNeue-Bold', 16.0)
No. of title candidates: 1
Title candidate passed DOI test (similarity 100%)
Title: A unified statistical model to support local sequence order...
                        *** end TITLE ***
                        *** OUTSIDERS ***
                    *** end OUTSIDERS ***
                               *** H1 ***
Found confident H1 by cue: INTRODUCTION
Found confident H1 by cue: RESULTS
Found confident H1 by cue: DISCUSSION
H1 font candidate statistics: [(('HelveticaNeue-Bold', 10.0), 3)]
Chosen H1 font: ('HelveticaNeue-Bold', 10.0)
H1 'confident': 1 INTRODUCTION
H1 'confident': 2 METHODS
H1 'confident': 3 RESULTS
H1 'confident': 4 DISCUSSION
H1 'confident': ACKNOWLEDGEMENTS
H1 'confident': REFERENCES
Found H1 in upper-case. Looking for possbile intruders and merged H2s
                          *** end H1 ***
                        *** ABSTRACT ***
Found cue word: ABSTRACT
Abstract: Functional relationships between proteins that do not sh...
                      *** end ABSTRACT ***
                          *** CAPTIONS ***
Page 3 - caption candidate 'confident' -  Table 1. PDB chains u...
Page 3 - caption candidate 'confident' -   Fig. 1. Fitting of t...
Page 4 - caption candidate 'confident' -   Fig. 2. The derived ...
Page 4 - caption candidate 'possible'  - Figure 2 shows the der...
Page 5 - caption candidate 'confident' -   Fig. 3. Percentage o...
Page 7 - caption candidate 'possible'  -  table 2 are amongst t...
Page 7 - caption candidate 'confident' -  Table 2. Top 18 most ...

7 total caption candidates
```

```
Caption font: ('Times-Roman', 8.0)
Label/text delimiters: {'. ': 5, ' ': 2}. Chosen: '. '
5 total captions
Page 3 - Removing intersecting region:
                    (body, possible) Table 1. PDB ch...
Page 3 - Removing intersecting region:
                    (body, possible)  Fig. 1. Fittin...
                  *** end CAPTIONS ***
                          *** H2+ ***
Found numbering pattering in H1s
H2, confident: 2.1 Representation o...
H2, confident: 2.2 Scoring function...
H2, confident: 2.3 Statistical mode
H2, confident: 2.4 Benchmark data
H2, confident: 2.5 Performance eval...
H2, confident: 2.6 Construction of ...
H2, confident: 2.7 Determination of...
H2, confident: 3.1 EVD for the SOIP...
                        *** end H2+ ***
               *** AUTHOR FOOTNOTES ***
          *** end AUTHOR FOOTNOTES ***
                     *** REFERENCES ***
Reference heading candidates:
    > [('REFERENCES', ('HelveticaNeue-Bold', 10.0))]
Reference heading found on page 7
Removing intersecting region (h1, confident): REFERENCES

Reference block, confident: Altschul,S.F. et al. ...
No bracket or number delimiter found. Assuming name delimiter.
page 7 - Completed bib-item (by name): AltschulS.F. et al. ...
page 7 - Completed bib-item (by name): Andreeva,A. and Murz...
page 7 - Completed bib-item (by name): Artymiuk,P.J. et al....
page 7 - Completed bib-item (by alignment): Barker J.A. and...
[...]
Reference block, confident: Bonnac,L. et al. (20...
page 7 - Completed bib-item (by name): Bonnac,L. et al. (20...
page 7 - Completed bib-item (by name): Brakoulias,A. and Ja...
[...]
Reference block, confident: Laskowski,R.A. et al...
page 8 - Completed bib-item (by name): Laskowski,R.A. et al...
page 8 - Completed bib-item (by name): Levitt, M. and Gerst...
[...]
```

```
Reference block, confident: Siggers, T.W. et al....
page 8 - Completed bib-item (by name): Siggers, T.W. et al....
page 8 - Completed bib-item (by name): Smith,P.A. and Romes...
[...]
                  *** end REFERENCES ***
                    *** OTHER BODIES ***
Set Method section possible body to conf: We represented prote...
Set Method section possible body to conf: 2.1.1 Local protein ...
Set Method section possible body to conf: After two structures...
Set Method section possible body to conf: pa ij = cos(a ij ), ...
Set Method section possible body to conf: We first download th...
Set Method section possible body to conf: The druggability of ...
                *** end OTHER BODIES ***
                         *** TABLES ***
Table for caption: Table 1. PDB chains ...
 -> Scores: [('down_single_column_not_conf', 36.666),
             ('down_2_columns_not_conf', 0)]
Created: page_003.table_1 >> variant: down_single_column_not_conf

Table for caption: Table 2. Top 18 most...
 -> Scores: [('down_2_columns_unassigned', 7.173),
             ('down_2_columns_not_conf', 7.173),
             ('down_single_column_unassigned', 5.055),
             ('down_single_column_not_conf', 5.055)]
Created: page_006.table_2 >> variant: down_2_columns_unassigned
                      *** end TABLES ***
                         *** INTEXTS ***
Bibliographic in-text reference pointers (ITRPs)
page 1 - ITRP found in: 'the structural compa...':
         > Levitt and Gerstein, 1998
page 1 - ITRP found in: 'the structural compa...':
         > Gerlt and Babbitt, 2001
[...]
page 7 - ITRP found in: 'a number of proteins...':
         > Smith and Romesberg, 2007
page 7 - ITRP found in: 'a number of proteins...':
         > Raman and Chandra, 2008
Figure/Table references
page 3 - Figure/Table reference found in: 'Table' + '1.'
page 3 - Figure/Table reference found in: 'Figure' + '1,'
[...]
page 5 - Figure/Table reference found in: 'Table' + '2,'
```

```
page 7 - Figure/Table reference found in: 'Table' + '2).'
                    *** end INTEXTS ***
                      *** EQUATIONS ***
Candidate label statistics: [('numeric', 7), ('alphanumeric', 2)]
Label justifications: [('Right', 8), ('Left', 1)]
Primary label justification: Right
page 2 - Removing/Splitting intersecting region (body, confident):
         > After two structures...
page 2 - Formula, confident:   S ij = (M ij x pa ij...
page 2 - Label: (1)
page 2 - Removing/Splitting intersecting region (body, confident):
         > Where M ij is the re...
page 2 - Formula, confident:   pa ij = cos(a ij ), ...
page 2 - Label: (2)
page 2 - Formula, confident:   pd ij = { I 1.0, d i...
page 2 - Label: (3)
page 3 - Removing/Splitting intersecting region  (, ):
         > P(s > S) = 1-exp(-ex...
page 3 - Formula, confident:   P(s > S) = 1-exp(-ex...
page 3 - Label: (4)
page 3 - Removing/Splitting intersecting region  (, ):
         > S 2 -μ Z = (5) σ
page 3 - Formula, confident:   S 2 -μ Z = σ
page 3 - Label: (5)
page 3 - Removing/Splitting intersecting region  (, ):
         > μ = a * ln(N) 2 +b *...
page 3 - Formula, confident:   μ = a * ln(N) 2 +b *...
page 3 - Label: (7)
                  *** end EQUATIONS ***
                    *** SAVE-2-XML ***
page 1 - Logically merging
         > ('body', 'confident') >> the structural compa...
         with previous:
         > ('body', 'confident') >> Evolutionary and fun...
page 2 - Logically merging
         > ('body', 'confident') >> Poisson processes (D...
         with previous:
         > ('body', 'confident') >> the structural compa...
[...]
                  *** end SAVE-2-XML ***
```

A stripped version of the XML produced as a result of the preceding processing is given below. All XML elements are marked in bold. The DoCO tags assigned to the various logical regions are visible as `class` attributes of the respective XML elements.

```
<pdfx>
  <meta>
    <doi>10.1093/bioinformatics/btp220</doi>
  </meta>
  <article>
    <front class="DoCO:FrontMatter">
     <outsider class="DoCO:TextBox" type="header">
       BIOINFORMATICS
     </outsider>
     <outsider class="DoCO:TextBox" type="header">
       Vol. 25 ISMB 2009, pages i305-i312
     </outsider>
     <title-group>
       <article-title class="DoCO:Title">
         A unified statistical model to support local sequence ...
       </article-title>
     </title-group>
     <contrib-group class="DoCO:ListOfAuthors">
       <contrib contrib-type="author">
         <name>Lei Xie</name><aff>1</aff><aff>*</aff>
       </contrib>
       <contrib contrib-type="author">
         <name>Li Xie</name><aff>2</aff>
       </contrib>
       <contrib contrib-type="author">
         <name>Philip E. Bourne</name><aff>1</aff><aff>2</aff>
       </contrib>
     </contrib-group>
     <footnote class="DoCO:Footnote">
       1 San Diego Supercomputer Center and 2 Skaggs School ...
     </footnote>
     <footnote class="DoCO:Footnote">
       * To whom correspondence should be addressed.
     </footnote>
     <abstract class="DoCO:Abstract">
       Functional relationships between proteins that do not share
       global structure similarity can be established by [...]
       Contact: <email>lxie@sdsc.edu</email>
     </abstract>
```

```
</front>
<body class="DoCO:BodyMatter">
  <section class="deo:Introduction">
    <h1 class="DoCO:SectionTitle" page="1" column="1">
      1 INTRODUCTION
    </h1>
    </section>
    <region class="DoCO:TextChunk" page="1" column="1">
      Evolutionary and functional relationships between proteins
      can be reliably inferred by the comparison of their sequences
      and benefits from a well-understood extreme value distribution
      (EVD) model that can be applied on a large scale
      (<xref ref-type="bibr" rid="R1" class="deo:Reference">
      Altschul et al., 1997</xref>; <xref ref-type="bibr" rid="R16"
      class="deo:Reference">Claverie, 1994</xref>; [...]
      <marker type="column" number="2"/><marker type="block"/>
      the structural comparison score follows an EVD if a summarized
      structural alignment score, rather than a root mean square[...]
    </region>
    <outsider class="DoCO:TextBox" type="footer">
      © 2009 The Author(s) This is an Open Access article dis[...]
    </outsider>
    <outsider class="DoCO:TextBox" type="header">
      L.Xie et al.
    </outsider>
    <section class="deo:Methods">
      <h1 class="DoCO:SectionTitle" page="2" column="2">
        2 METHODS
      </h1>
      <section class="DoCO:Section">
        <h2 class="DoCO:SectionTitle" page="2" column="2">
          2.1 Representation of protein structures
        </h2>
        <region class="DoCO:TextChunk" page="2" column="2">
          We represented protein structures using Delaunay
          tessellation of Cα atoms that are characterized
          by geometric potentials [...]
        </region>
      </section>
      [...]
    </section>
    [...]
```

```
      <section class="DoCO:Bibliography">
        <h1 class="DoCO:SectionTitle" page="7" column="1">
          REFERENCES
        </h1>
        <ref-list class="DoCO:BiblioGraphicReferenceList">
          <ref rid="R1" class="deo:BibliographicReference" page="7"
               column="1">
            Altschul,S.F. et al. (1997) Gapped BLAST and PSI-BLAST:
            a new generation of protein database search programs.
            [...]
          </ref>
          <ref rid="R2" class="deo:BibliographicReference" page="7"
               column="1">
            Andreeva,A. and Murzin,A.G. (2006) Evolution of protein
            fold in the presence of functional constraints.
            Curr. Opin. [...]
          </ref>
          [...]
        </ref-list>
      </section>
    </body>
  </article>
</pdfx>
```

Appendix B

KPEX LOG AND OUTPUT EXAMPLES

The following is KPEX's log output of the steps carried out for extracting the Top 15 keyphrases shown in Table 4.9, for the publication Krauthammer and Nenadić (2004). KPEX was run over the PDFX output of the article PDF, with a region weight combination of 1111111110, to leave out the Bibliography section.

```
                        *** POS Tag ***
                    *** end POS Tag ***
              *** Parse for Candidates ***

Number of keyphrase candidates: 5634
Examples [POS tags]:
   approved names ['VVN NNS']
   most important terms ['RBS JJ NNS']
   amount of published work ['NN IN VVN NN']
   more complex morpho-syntactic features ['JJR JJ JJ NNS']
   linguistically related words ['RB VVN NNS']
   Boeckmann ['NP']
         *** end Parse for Candidates ***
                     *** Get Acronyms ***
                 *** end Get Acronyms ***
                *** Filter Candidates ***
(stoplisted prefix):
   most important terms
   same corpus
   other molecular class
```

```
   [...]
(stoplisted suffix):
   considered part
   EURALEX 96.
   molecular class
   [...]
(stoplisted):
   errors
   types
   abstract
   [...]

Number of keyphrase candidates: 3901
           *** end Filter Candidates ***
          *** Stemming/Lemmatisation ***
       *** end Stemming/Lemmatisatin ***
              *** Get Frequencies ***
            *** end Get Frequencies ***
              *** Merge Frequencies ***
(casing):
   Term Recognition (3.0) -> term recognition (35.0)
   Acronyms (1.0) -> acronyms (27.0)
   Acronym (3.0) -> acronym (27.0)
   [...]
(lemmatised):
   acronyms (28.0) -> acronym (30.0)
   Protein Name (8.0) -> protein names (22.0)
   dictionaries (12.0) -> dictionary (13.0)
   [...]
('of' variants):
   recognition of acronyms (1.0) -> acronyms recognition (1.0)
   identification of terms (2.0) -> term identification (38.0)
   [...]
(unknown plurals/hyphens/spaces):
   F-scores (3.0) -> F-score (8.0)
   information-extraction (1.0) -> information extraction (8.0)
   SwissProt (2.0) -> Swiss-Prot (6.0)
   [...]

Deleting: Automatic ['NP'] - not a candidate when lowercased
Deleting: Biomedical ['NP'] - not a candidate when lowercased
[...]
```

```
Abbreviations:
NLP (20.0) -> natural language processing (3.0)
ATR (19.0) -> automatic term recognition (5.0)
POS (11.0) -> part-of-speech (2.0)
EFs (7.0) -> expanded forms (2.0)
NPs (6.0) -> noun phrases (2.0)
[...]

Number of keyphrase candidates: 3584
           *** end Merge Frequencies ***
                *** Modified C-value ***
Ignoring nested term: science journal articles
     >> same freq as: biological science journal articles
Ignoring nested term: extensive list of gene names
     >> same freq as: extensive list of gene names from FlyBase
Ignoring nested term: gene names from FlyBase
     >> same freq as: extensive list of gene names from FlyBase
[...]
            *** end Modified C-value ***
               *** Structure Weights ***
           *** end Structure Weights ***
                  *** Prepare Output ***
(too long): 7 - Probabilistic Term Variant Generator for Biomedical Terms
(too long): 6 - important step towards final term identification
(too long): 6 - successful identification of terms from literature
(too long): 6 - increasingly large body of biomedical articles
```

Having applied the region weight combination 1111111110, terms that occur only in the Bibliography section will have their score set to 0. These will now be removed because of being under the minimal score threshold (1.0).

```
Deleting terms below score threshold:
   protein structures
   Informatics
   Annual ACM SIGIR
   Boeckmann
   Inform
   Kluwer
   [...]

Number of keyphrase candidates: 1131
```

Finally, KPEX proceeds to remove any nested terms ranked lower than their longer variants, making also sure that none of these remain in the list of Top 15 keyphrases that will be output.

```
Removing nested terms:
   protein
   recognition
   gene
   automatic term
   phrases
   [...]

Lower-ranked nested terms removed: 175
Number of keyphrase candidates: 956
              *** end Prepare Output ***

Final output:
   term recognition
   term identification
   automatic term recognition (ATR)
   natural language processing (NLP)
   protein names
   term mapping
   term classification
   gene names
   gene and protein names
   biomedical literature
   information extraction (IE)
   term identification process
   expanded forms (EFs)
   term occurrences
   Gene Ontology (GO)
```

Table B.1: Extended view of the keyphrases extracted by KPEX from the two scientific articles references in Table 4.9. KPEX was run over the PDFX output of the article PDFs with a region weight combination of 1111111110 and was asked to output 60 terms, as opposed to 15. Because of this, the top extracted keyphrases differ slightly between the two tables. This behaviour is explained in Section 4.2.5.

"Term identification in the biomedical literature" by Krauthammer and Nenadić (2004)	**"Transgenes encompassing dual-promoter CpG islands from the human TBP and HNRPA2B1 loci are resistant to heterochromatin-mediated silencing"** by Antoniou et al. (2003)
term recognition	TATA binding protein (TBP)
term identification	enhanced green fluorescent protein (EGFP)
automatic term recognition (ATR)	CpG island
natural language processing (NLP)	Locus control regions (LCRs)
protein names	methylation-free CpG islands
term mapping	open chromatin
term classification	tissue culture cells
gene names	housekeeping genes
gene and protein names	locus control
protein	dominant chromatin
recognition	dominant chromatin opening
biomedical literature	dominant chromatin opening function
information extraction (IE)	transfected tissue culture cells
term identification process	divergently transcribed promoters
expanded forms (EFs)	position effect variegation (PEV)
automatic term	stably transfected tissue culture cells
term occurrences	chromatin remodeling
Gene Ontology (GO)	transgene
biomedical terms	absence of drug selective pressure
noun phrases (NPs)	chromatin structure
term classification and term mapping	gene expression
Hidden Markov Model (HMM)	Fluorescence in situ hybridization (FISH)
term recognition and classification	chromatin remodeling function
acronym	open chromatin structure
Medline abstracts	genomic clone
biomedical domain	centromeric heterochromatin

Continued on next page

Table B.1 – continued from previous page

Krauthammer and Nenadić (2004)	**Antoniou et al. (2003)**
candidate terms	HNRPA2B1-CBX3 CpG island
term candidates	chromatin
Natural Language	Medical Research Council (MRC)
referent data sources	RNA polymerase II
GENIA corpus	clones
biomedical text	HNRPA2B1-CBX3 CpG island dual-promoter
domain concepts	dominant chromatin remodeling function
functional words	dual divergently transcribed promoters
term recognition and term classification	flanking sequences
precision	human globin
acronym recognition	integrated transgenes
lexical units	transgene copy
machine learning (ML)	human TBP (human TATA binding protein)
named entity (NE)	functional domain
p53 protein	expression pattern
term classes	EGFP reporter
true positives	FACS analysis
recall	HNRPA2B1 promoter
Yeast Protein Database (YPD)	HNRPA2B1-CBX3 dual-promoter
database	chromatin domain
automatic term identification	gene loci
colleagues	genomic fragment
ATR approaches	satellite repeat
ML approach	CpG density map
Term Variant	EGFP reporter gene
UMLS Metathesaurus	TBP-PSMB1 and HNRPA2B1-CBX3 loci
ambiguous terms	centromeric integration events
core terms	dual-promoter CpG island
information retrieval (IR)	human TBP transgene
recognition of protein	satellite repeat sequences
term boundaries	second round PCR
terms to broader biomedical classes	transgene integration site
term ambiguity	ubiquitously expressed genes
dictionary	44-kb genomic clone encompassing TBP

Table B.2 presents a 3-way comparison between the processing results of KP-Miner, TerMine and KPEX. The three systems were run on one article of the physics domain (Boyarsky et al., 2014) and one of the authors was asked to judge the terms according to their relevance for the article. The table notations are as follows:

- ✓A check mark indicates a relevant keyword or keyphrase
- No modification indicates a correct term, but not particularly important
- ~~*A strikeout*~~ indicates an irrelevant term or processing fault

Table B.2: Relevance judgements on keyphrases extracted by KP-Miner (El-Beltagy and Rafea, 2009), TerMine (Frantzi et al., 2000) and KPEX. KPEX was run with the best weight combination obtained in the experiments of Section 4.3 (5311020001).

"An unidentified line in X-ray spectra of the Andromeda galaxy and Perseus galaxy cluster" by Boyarsky et al. (2014)		
KP-Miner	**TerMine**	**KPEX**
~~*kev*~~	✓blank sky dataset	✓dark matter (DM)
✓perseus cluster	✓dm decay line	✓Perseus cluster
✓andromeda galaxy	✓dark matter	✓Andromeda galaxy
perseus	~~*count rate*~~	~~*blank sky*~~
~~*line*~~	~~*blank sky*~~	Perseus
✓dark matter	✓andromeda galaxy	✓DM decay
✓galaxy cluster	pn camera	✓blank sky dataset
~~*observations*~~	~~*table i*~~	X-ray spectra
x-ray spectra	~~*effective area*~~	✓galaxy cluster
~~*weak line*~~	dm distribution	✓DM decay line
✓blank sky dataset	✓dm decay	~~*weak line*~~
~~*exposure*~~	deep blank sky dataset	~~*keV*~~
✓xmm-newton	residual soft proton	DM distribution
~~*decay*~~	✓surface brightness profile	✓Perseus galaxy cluster
~~*leiden*~~	~~*brightness profile*~~	~~*spectra*~~
neutrino	✓average column density	✓surface brightness profile
x-ray	dm decay signal	~~*brightness profiles*~~
particle	~~*p erseus cluster*~~	~~*individual objects*~~
~~*instrumental*~~	~~*fin -fout*~~	spectral resolution
~~*astrophysical*~~	~~*weak line*~~	~~*Boyarsky*~~

Continued on next page

Table B.2 – continued from previous page

KP-Miner	TerMine	KPEX
decay line	*fin fout*	*Ruchayskiy*
fully consistent	*positive residual*	*Future detections or non-detections*
spectral resolution	x-ray spectra	XMM-Newton X-ray observatory
✓instrumental lines	*mos2 camera*	*keV in X-ray spectra*
spectra	column density	*decay*
weak	*ecole polytechnique federale de lausanne*	DM signal
surface brightness	surface brightness	*expected behavior*
mdm	*off-axis angle cleaned exposure arcmin mos1*	✓instrumental lines
energy	*instrumental ca k line.3 combined fit*	*instrumental origin*
consistent	*soft proton*	*keV energy*
fluxes	*hot big bang cosmology paradigm*	*much more convincing evidence*
ukraine	*xmm-newton epic mos1 ccd6 update*	*present day X-ray telescopes*
object	*dark matter column density sdm*	*previous searches*
brightness	*off-axis angle cleaned exposure fov*	✓sterile neutrinos
signal	*comparable flux m31 off-center observation*	*Iakubovskyi*
lifetime	long exposure blank sky dataset	*exposure*
clusters	m31 surface brightness profile perseus	DM decay signal
model	*pc2 cts fov cm2 sec*	✓Milky Way halo

Appendix C

ADDITIONAL KEYPHRASE EXTRACTION RESULTS

Table C.1: Full table of keyphrase extraction results for the SemEval-2010 Test dataset, using the revised evaluation procedure (script and gold-standard list). The scores for KP-Miner, TerMine and TF-IDF when using region weights are also shown. Results obtained by the KPEX system of this dissertation are highlighted.

System (input)	Top 5			Top 10			Top 15		
	P	**R**	F_1	**P**	**R**	F_1	**P**	**R**	F_1
KPEX (5311020001)	41.4	41.4	41.4	32.5	32.6	32.6	26.0	29.1	27.4
KP-Miner[1] (0000011100)	39.2	39.2	39.2	32.0	32.1	32.1	25.6	28.6	27.0
KP-Miner (1111111111)	38.8	38.8	38.8	31.7	31.8	31.8	25.5	28.5	26.9
HUMB	38.0	38.0	38.0	31.1	31.2	31.2	25.1	28.0	26.5
KPEX (0110000001)	37.8	37.8	37.8	30.8	30.9	30.9	24.8	27.7	26.2
KPEX (1100000001)	37.0	37.0	37.0	30.7	30.8	30.8	24.7	27.6	26.0
SEERLAB	41.8	41.8	41.8	30.9	31.0	31.0	24.3	27.1	25.6
KPEX (0110000000)	38.0	38.0	38.0	30.9	31.0	31.0	24.3	27.1	25.6
KPEX (1100000000)	37.6	37.6	37.6	30.3	30.4	30.4	24.1	27.0	25.5
WINGNUS	39.0	39.0	39.0	29.8	29.9	29.9	23.7	26.5	25.1
KPEX (0000010000)	38.6	38.2	38.4	30.3	30.1	30.2	23.7	26.2	24.9
KP-Miner (full-text)	37.2	37.2	37.2	29.5	29.5	29.6	23.6	26.4	24.9
ICL	33.6	33.6	33.6	28.5	28.6	28.6	23.3	26.0	24.6

Continued on next page

[1]Apart from the table entry 'KP-Miner (2010)', the KP-Miner version is the one made available for this research, not the original SemEval-2010 one.

Table C.1 – continued from previous page

System (input)	Top 5			Top 10			Top 15		
	P	**R**	F_1	**P**	**R**	F_1	**P**	**R**	F_1
KP-Miner (2010)	36.6	36.6	36.6	29.3	29.4	29.4	23.1	25.8	24.4
KPEX (0000011100)	35.6	35.6	35.6	28.7	28.8	28.7	23.1	25.9	24.4
SZTERGAK (2014)	39.8	39.8	39.8	29.5	29.6	29.6	23.0	25.7	24.3
KPEX (1111111111)	35.4	35.4	35.4	28.7	28.8	28.7	23.0	25.7	24.3
KP-Miner (1100000001)	37.8	37.8	37.8	29.6	29.7	29.7	22.8	25.5	24.1
KP-Miner (0000010000)	36.3	34.6	35.5	29.8	28.3	29.0	23.2	24.4	23.8
TerMine (5311020001)	38.2	38.2	38.2	29.3	29.4	29.4	22.5	25.1	23.7
KP-Miner (1100000000)	38.4	38.0	38.2	29.5	29.3	29.4	22.4	24.8	23.6
KP-Miner (0110000001)	37.4	37.4	37.4	29.0	29.1	29.1	22.3	25.0	23.6
KPEX (full-text)	35.2	35.2	35.2	27.2	27.3	27.2	22.1	24.7	23.3
KX_FBK	34.2	34.2	34.2	27.0	27.1	27.1	22.0	24.6	23.2
KP-Miner (0110000000)	36.6	36.6	36.6	28.8	28.9	28.9	21.7	24.3	22.9
TerMine (0110000001)	37.4	37.4	37.4	27.7	27.8	27.8	21.0	23.5	22.2
SZTERGAK	31.8	31.8	31.8	25.4	25.5	25.5	20.8	23.3	22.0
TerMine (1100000001)	36.4	36.4	36.4	26.8	26.9	26.9	20.7	23.2	21.9
TerMine (0000010000)	34.8	34.4	34.6	26.2	26.0	26.1	20.1	22.3	21.2
Maui	35.6	35.6	35.6	25.6	25.7	25.7	20.1	22.4	21.2
TerMine (0110000000)	39.2	39.2	39.2	26.6	26.7	26.7	19.8	22.1	20.9
DFKI	29.0	29.0	29.0	23.0	23.1	23.0	19.3	21.6	20.4
TerMine (1100000000)	37.6	37.6	37.6	25.8	25.9	25.9	19.3	21.5	20.3
TerMine (0000011100)	31.0	31.0	31.0	23.7	23.8	23.8	18.2	20.3	19.2
TerMine (1111111111)	31.0	31.0	31.0	23.4	23.5	23.4	17.9	20.0	18.9
SJTULTLAB	30.0	30.0	30.0	22.6	22.7	22.6	17.9	20.0	18.9
DERIUNLP	25.4	25.4	25.4	20.6	20.7	20.6	17.6	19.7	18.6
UNICE	27.8	27.8	27.8	22.3	23.4	22.3	17.4	19.5	18.4
TerMine (full-text)	29.4	29.4	29.4	22.2	22.3	22.2	17.4	19.5	18.4
TF-IDF[2] (5311020001)	23.0	23.0	23.0	20.7	20.8	20.7	17.0	19.0	17.9
TF-IDF (0000010000)	22.8	22.6	22.7	19.2	19.1	19.1	16.4	18.2	17.3
TF-IDF (0110000001)	23.8	23.8	23.8	19.7	19.8	19.7	16.2	18.1	17.1
Likey	28.8	28.8	28.8	21.3	21.4	21.3	16.1	18.0	17.0
TF-IDF (1111111111)	22.4	22.4	22.4	19.1	19.2	19.1	16.1	18.0	17.0
TF-IDF (0000011100)	22.2	22.2	22.2	19.1	19.2	19.1	16.1	18.0	17.0

Continued on next page

[2]Apart from the table entry 'TF-IDF (2010)', the TF-IDF implementation is the one of this research, not the original SemEval-2010 baseline.

Table C.1 – continued from previous page

System (input)	Top 5			Top 10			Top 15		
	P	**R**	F_1	**P**	**R**	F_1	**P**	**R**	F_1
UNPMC	17.8	17.8	17.8	18.5	18.6	18.5	16.0	17.9	16.9
TF-IDF (0110000000)	23.6	23.6	23.6	19.7	19.8	19.7	15.9	17.8	16.8
TF-IDF (1100000001)	23.4	23.4	23.4	19.5	19.6	19.5	15.9	17.7	16.8
TF-IDF (1100000000)	23.0	23.0	23.0	19.5	19.6	19.5	15.9	17.7	16.8
TF-IDF (full-text)	22.6	22.6	22.6	19.0	19.1	19.0	15.9	17.7	16.8
ME	23.4	23.4	23.4	18.8	18.9	18.8	14.9	16.7	15.8
NB	23.4	23.4	23.4	18.8	18.9	18.8	14.9	16.7	15.8
JU_CSE	26.8	26.8	26.8	19.6	19.7	19.6	14.6	16.3	15.4
TF-IDF (2010)	22.0	22.0	22.0	17.8	17.9	17.8	14.1	15.7	14.9
UvT	24.2	24.2	24.2	18.2	18.3	18.2	13.5	15.1	14.3
POLYU	23.0	23.0	23.0	16.8	16.9	16.8	13.1	14.7	13.9