The Impact of Document Structure on Keyphrase Extraction

Katja Hofmann k.hofmann@uva.nl
Manos Tsagkias e.tsagkias@uva.nl
Edgar Meij edgar.meij@uva.nl
Maarten de Rijke mdr@science.uva.nl

ISLA, University of Amsterdam
Science Park 107, 1098 XG Amsterdam

ABSTRACT

Keyphrases are short phrases that reflect the main topic of a document. Because manually annotating documents with keyphrases is a time-consuming process, several automatic approaches have been developed. Typically, candidate phrases are extracted using features such as position or frequency in the document text. Document structure may contain useful information about which parts or phrases of a document are important, but has rarely been considered as a source of information for keyphrase extraction.

We address this issue in the context of keyphrase extraction from scientific literature. We introduce a new, large corpus that consists of full-text journal articles, where the rich collection and document structure available at the publishing stage is explicitly annotated.

We explore features based on the XML tags contained in the documents, and based on generic section types derived using position and cue words in section titles. For XML tags we find sections, abstract, and title to perform best, but many smaller elements may be beneficial in combination with other features. Of the generic section types, the discussion section is found to be the most useful for keyphrase extraction.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.7 Digital Libraries

General Terms

Algorithms, Experimentation, Measurement

Keywords

Keyphrase extraction, Scientific literature search

1. INTRODUCTION

Keyphrases are short phrases that indicate the main topic of a document. Initially, curator-assigned keyphrases were used to facilitate information access [7, 10] but today keyphrases are increasingly important for exploratory search—to quickly get an overview of the contents of a collection, and for discovering information objects in the case of under-specified information needs. In the context of scientific literature search, keyphrases appear to be one of the clues that researchers use to make relevance decisions, and have been found beneficial for exploring digital libraries [1, 5]. As manual assignment of keyphrases is a tedious process, various methods to automatically suggest keyphrases have been proposed [13], that select phrases based on features capturing usage of the phrase.

In this paper we analyze the use of features based on document structure for keyphrase extraction from scientific documents. Despite the large amount of structured and semi-structured documents available on the web and in organizations, there is little work on exploiting document structure for keyphrase extraction. We hypothesize that document structure provides useful cues for keyphrase extraction because the structural elements follow conventions that direct the reader to important parts of the document content.

We conduct our experiments on a new document collection. This collection is substantially larger than any collections previously considered for this task and the clean document structure available at the publishing stage is preserved.

The remainder of this paper is organized as follows. In Section 2 we give a brief overview of existing keyphrase extraction approaches. We describe our document collection in Section 3 and detail our approach in Section 4. We present and analyze our results in Section 5 and end with a concluding section.

2. RELATED WORK

Various approaches to keyphrase extraction have been explored in the past, which can be divided into unsupervised methods and methods that apply supervised learning. Unsupervised methods filter or rank candidate phrases according to a scoring function, either using a single feature (ranking criterion), or a combination of features [4, 12]. Approaches using supervised learning train a machine learning algorithm to predict whether a phrase is a keyphrase or not [6, 13, 15].

In this paper we focus on unsupervised keyphrase extraction using rankings of individual features that are based on different elements of document structure. Considering the growing amount of structured and semi-structured data available online and in organizations, it is surprising that document structure has rarely been considered for keyphrase extraction with the exception of [9, 14]. Wang and Peng [14] use features such as *title frequency* and *paragraph frequency* to extract keyphrases from web pages. Nguyen and Kan [9] use similar features to extract keyphrases from scientific publications, and also use the frequency of term occurrences in generic section types. In both cases, however, structural features are used in combination with several other features and it is not clear how they contribute to the final system performance.

3. DOCUMENT COLLECTION

We run our experiments on a collection of scientific journal articles provided by Elsevier. The collection consists of 14,724 articles

CIKM'09, November 2–6, 2009, Hong Kong, China.

from 26 journals in the Food Informatics and Computer Science domains published between 1995 and 2005. The rich document structure available at the publishing level is preserved in this data set and is provided in the form of XML markup. Document annotations serve different functions, e.g., there are elements indicating article abstracts, individual sections, lists and list items, italicized terms, individual elements of mathematical formulas, references and citations, etc. Overall, there are more than 100 XML codes.

The documents' authors have annotated 8,479 (58%) of the documents with between 1 and 146 keyphrases (mean: 6.33, mode: 4). More than 75% of the documents with keyphrases have between 3 and 6 keyphrases. Keyphrases are between 1 and 142 terms long. We ignore outliers, automatically removing keyphrases of length greater than 10 terms. The remaining keyphrases have an average of 2.13 terms. In the documents we analyzed there are 53,651 keyphrases; of these, 38,222 (71.24%) also occur in the document content and can (theoretically) be extracted automatically.

For our experiments we use a subset of the collection, consisting of the 5,504 documents for which both the document's full text and manually annotated keyphrases are available. We make this selection, as opposed to also including articles for which only keyphrases and abstract are provided, to avoid any possible bias.

4. APPROACH

We follow a two-step approach to keyphrase extraction: (i) extract candidate phrases from the document text and (ii) rank the candidate list according to features assumed to reflect the phrases' likelihood of being assigned as a keyphrase. Our main focus is on the second step. We view the task of identifying keyphrases from a set of candidate phrases as a problem of ranking candidate phrases according to their probability of being selected as a keyphrase. This probability can be expressed as $p(t = K|D)$, the probability p of event K that a phrase t drawn from document D is assigned as a keyphrase to this document. In the current paper we specifically focus on using *TFIDF* to estimate this probability for each structural element. We call each individual estimation a *feature*.

4.1 Experimental Setup and Preprocessing

In our experimental setup we model the scenario where we have a number of existing journal issues available for analysis and training, and want to predict keyphrases for a set of unseen documents published in subsequent issues. We first split the collection by publication date and journal issue. Then, we take the first 20% of the issues per journal as *development* set, the next 60% as *training* set, and the last 20% as the *test* set. The development set is used for our experiments to select a good candidate selection approach, the training set is used to evaluate individual features.

All documents in the collection are pre-processed and indexed as follows. The XML documents were parsed and the textual content of each XML element was indexed using Lucene.[1] (as separate fields, with stemming and without stopword removal). In this way elements can be searched, and we can also retrieve the full text content of a specified element.

We apply sentence-splitting and PoS-tagging using an off-the-shelf software package.[2] Stemming is performed using the implementation of the snowball stemmer included with Lucene.

4.2 Candidate Selection

The first step in keyphrase extraction is to select candidate phrases from the document text. We compare existing methods in order to identify a method that achieves a good trade-off between precision and recall. Ideally, we want to obtain a large set of candidate phrases to maximize recall. However, a large number of candidate phrases results in a large processing overhead, as features for all candidates have to be calculated.

We evaluate the following candidate selection approaches:

- All n-grams: for each sentence, we generate all possible subsequences of up to n words.
- Filtered n-grams: we generate all n-grams, as above, and accept those that follow certain PoS patterns [5].
- PoS patterns: we extract all PoS patterns that occur as keyphrases in the first 10% of the corpus and use this list of patterns to filter the n-grams generated from each sentence [6].

After candidate selection, we obtain a list of candidate phrases per document. The next step is to extract features for each phrase.

4.3 Features

Many document types, or genres, exhibit a characteristic style and form. E.g., news articles typically have a headline summarizing the article, an indication of the news source, location and date, etc. Both authors and readers are aware of these conventions and use them to effectively process the document content. Similarly, scientific papers are subject to constraints that have developed over time, and are, for example, enforced through the review process. This results in a certain degree of standardization [2].

We use the document structure available in our collection in two ways. First, we model the content of markup elements to identify whether some of these elements are useful for keyphrase extraction, and which elements are the most informative. Second, we focus on section structure and augment the markup with position and clues in the section headers to identify main section types.

4.3.1 XML markup

We assume that the types of markup elements in our document collection have semantics that are similar across documents. Overall, some types of elements may be more likely to contain content representative of the document content than others. E.g., a reader may be more likely to find information about the document topic in the title than in the author's contact information. In this case the XML markup may serve different purposes. It can explicate structure corresponding to the conventions mentioned above (marking the document title, abstract, sections and section headings, lists, figure captions, etc.). But the specific markup format has been developed by a publishing house for use during the electronic publishing process and may not necessarily be relevant for readers. The markup ranges from coarse (e.g., section) to very fine granularity (e.g., individual symbols within a mathematical formula, individual cells of a table). Thus, some markup may be helpful for identifying important phrases in the document, whilst others are not. We include every XML element type that was found to contain at least one keyphrase at least once in our development set.

4.3.2 Sections

A particularly important structural element of scientific articles are sections and a lot of research is concerned with modeling the discourse structure created through the use of section types [8]. Given the different functions of section types (introduction, results presentation, etc.), the content of some sections may be more representative of a document's topic than others. This hypothesis is supported by Shah et al. [11] who analyzed occurrences of MeSH[3]

[1] http://lucene.apache.org/
[2] http://alias-i.com/lingpipe

[3] Medical Subject Headings—a controlled vocabulary of indexing terms.

terms by section type in 104 articles of a biomedical journal. The authors found a relatively higher concentration of MeSH terms in abstracts and methods sections, and also found qualitative differences between the different sections.

To identify generic section types we make use of two types of cues: (i) position and (ii) characteristic words in section titles [8]. First, we identify top-level sections based on section numbering. Position is then inferred from the ordering of the top-level sections in the document, and we include features for the first N and last N sections (in our case we set N to 10, a number chosen to exceed our estimate of generic section types identifiable based on position).

Characteristic words in section headings were obtained from the most frequent section titles of documents in the development set. All top-level sections containing a cue word were assigned to the corresponding type; Table 1 summarizes the results. For each generic section type we generate probabilistic scores as described before and draw comparisons between section types and the full document text.

Type	cue words	count
Introduction	introduction	954
Background	background, related work	114
Method	method	373
Result	result	415
Discussion	discussion	410
Conclusion	conclusion, concluding, summary	651

Table 1: Generic section types, cue words, and frequency of occurrence on the development set. For 98% of the documents at least one generic section type can be identified using cue words.

4.4 Evaluation

Our choice of evaluation measures is based on our view of keyphrase extraction as a ranking problem. Previously, keyphrase extraction has been evaluated using the number of correctly identified keyphrases and measures typical for classification, such as precision and recall (and sometimes F-score) [6]. These measures are based on evaluating sets, where there is no ordering in the returned positive and negative instances. A problem with these measures is that there typically is a threshold that needs to be determined, either beforehand or through tuning, to control how many keyphrases to return. Depending on the application, different thresholds may be appropriate, and for comparing methods, an arbitrary threshold needs to be chosen. For these reasons we complement these set-oriented evaluation measures with measures that take ranking into account. We propose the use of evaluation measures for ranked lists as they are typically used in Information Retrieval (IR):

- Mean reciprocal rank (MRR) is the averaged inverse of the rank of the first correctly returned keyphrase.
- Precision at N (P@N) is the portion of correctly identified keyphrases returned within the top N results.
- Mean average precision (MAP) is the average precision at N, where N takes on the ranks at which correct keyphrases are returned, averaged over all documents.

As some features cannot be generated for all documents we also report *coverage*: the portion of documents for which keyphrase suggestions were generated. In case coverage is under 100% we average the remaining evaluation measures only over these covered documents. We evaluate the top 100 results of the ranked lists of keyphrase suggestions against the author-annotated keyphrases supplied with the documents. For each individual feature we rank candidate phrases by that feature and evaluate the 100 top-ranked results.

5. RESULTS AND DISCUSSION

In this section we present the results of our analysis. First, we detail the performance of candidate selection methods, then we show the performance of ranking phrases using document structure.

5.1 Candidate Selection

Table 2 shows for each candidate selection approach the number of selected candidate phrases, correct keyphrases, and precision and recall if all phrases were to be considered keyphrases.

Method	candidates	correct	recall	prec.
All 3-grams	4,735,793	2,208	0.7449	0.0006
All 5-grams	9,882,514	2,284	0.7705	0.0003
All 10-grams	21,196,347	2,293	0.7732	0.0001
Filtered 3-grams*	1,437,186	2,186	0.7385	0.0018
Filtered 5-grams	2,825,797	2,256	0.7622	0.0010
Filtered 10-grams	5,896,621	2,265	0.7649	0.0005
PoS patterns	1,206,078	2,166	0.7296	0.0021

Table 2: Performance of different candidate selection methods on the development set. The method used in subsequent experiments is marked with *.

As expected, we achieve the highest recall using all n-grams but at very low precision. The highest recall that can be achieved on the data set is 77.32%: the missing keyphrases are not contained in the document text and simply cannot be assigned using an extraction-based approach. These typically include morphological variations that are not collapsed through stemming, or words that are broad descriptions of a document topic, too broad to occur in running text and more comparable to generic categories.

In comparison with previous work, our recall score after candidate selection is slightly lower [5, 6] and precision is substantially lower [6]. This stems from the fact that we select candidate phrases from full texts, not just abstracts. *Filtered 3-grams* is the candidate selection method of choice as it combines high recall and a reasonable number of candidate phrases.

5.2 XML Markup

For XML markup features we report $TFIDF$ on the features with the highest scores and high coverage (Table 3). Coverage is relatively low for these features, as many XML markup codes are only used in some articles. As expected, the best-performing elements are the $TFIDF$ scores for abstract, sections, and title. However, there are many other elements that achieve high performance, such as the bibliography ($ce : bibliography - sec$), captions ($ce : caption$), and table headings ($thead$). As such, a high $TFIDF$ of a phrase in cited article title, books, etc. is a good indicator for keyphrases.

For short elements TFIDF may not be optimal for identifying keyphrases. Longer samples are needed to get a meaningful distribution. We think that high-precision elements can be more useful when used in combination with other evidence.

5.3 Section Structure

Generic section types show an interesting pattern (Table 4). "Introduction" seems to be the most general type and is found in almost 80% of the test documents and keyphrase extraction performance is also good. Background has very low coverage and low performance. 'Method", "result" and "conclusion" sections show

feature	coverage	P@10	recall	MRR	MAP
$TFIDF_{fulltext}$	100%	**0.1001**	**0.4051**	0.3366	0.0074
$TFIDF_{ce:abstract-sec}$	100%	0.0957	0.3578	**0.3515**	0.0067
$TFIDF_{ce:bibliography-sec}$	100%	0.0761	0.3319	0.2890	0.0055
$TFIDF_{ce:caption}$	91%	0.0603	0.1777	0.2321	0.0094
$TFIDF_{ce:sections}$	100%	0.0955	0.3768	0.3272	0.0071
$TFIDF_{ce:simple-para}$	100%	0.0935	0.3582	0.3317	0.0067
$TFIDF_{ce:title}$	100%	0.0762	0.2411	0.2078	**0.0267**
$TFIDF_{ce:table}$	70%	0.0512	0.1575	0.1728	0.0056
$TFIDF_{thead}$	50%	0.0320	0.0618	0.1246	0.0149
$TFIDF_{sb:book}$	56%	0.0147	0.0517	0.0839	0.0104
$TFIDF_{sb:edited-book}$	56%	0.0197	0.0619	0.1244	0.0150
$TFIDF_{sb:maintitle}$	98%	0.0839	0.2984	0.3084	0.0093
$TFIDF_{sb:title}$	98%	0.0845	0.3002	0.3076	0.0091

Table 3: Performance of $TFIDF$ of XML markup features. Best performance is achieved with abstract, title, and sections.

medium coverage, and good performance. By far the best is the "discussion" section, which even performs better than when generating scores on the full-text. These results indicate that, when we can identify section types, such information can be very useful for keyphrase extraction.

feature	coverage	P@10	recall	MRR	MAP
$TFIDF_1$	83%	0.0941	0.3098	0.3590	0.0062
$TFIDF_2$	83%	0.0701	0.2310	0.2645	0.0048
$TFIDF_3$	83%	0.0696	0.2217	0.2650	0.0051
$TFIDF_n$	83%	0.0968	**0.3123**	0.3445	0.0066
$TFIDF_{n-1}$	83%	0.0740	0.2508	0.2700	0.0052
$TFIDF_{n-2}$	83%	0.0689	0.2391	0.2517	0.0047
$TFIDF_{INTR}$	79%	0.0956	0.3118	0.3619	0.0063
$TFIDF_{BACK}$	9%	0.0612	0.2652	0.2357	0.0034
$TFIDF_{METHOD}$	33%	0.0960	0.2304	0.3310	0.0072
$TFIDF_{RESULT}$	36%	0.0944	0.2282	0.3481	0.0077
$TFIDF_{DISC}$	40%	**0.1205**	0.3112	**0.4212**	**0.0092**
$TFIDF_{CONCL}$	47%	0.0728	0.2933	0.2748	0.0042

Table 4: Performance of features based on generic section types. The discussion section, as well as first and last sections perform best.

As expected, the sections based on position have higher coverage than section types identified based on section. Beyond section 3 coverage drops, as there are fewer documents with more than three sections. Performance of the features based on the first and last sections is good. We assume that these sections correspond to "introduction" and "discussion" / "conclusion" (given the scores, discussion is more likely).

In comparison with [11] we see similarities and differences which may be attributed to the field of the documents in question. They found the largest number of keywords on average in the methods and introduction sections, and the highest concentration (keyword over section length) in the abstract and introduction. We also get good performance on the introduction, but find the discussion section to perform best overall.

6. CONCLUSION

In this paper we analyzed the use of features based on document structure for keyphrase extraction from scientific documents. Experiments were performed on a new corpus of scientific documents that is much larger than corpora previously used for this task.

The features based on document structure were modeled probabilistically and evaluated using evaluation measures for rankings. We analyzed features derived from XML markup and on generic section structure. In addition, the performance of existing candidate selection approaches was evaluated on the new corpus.

We found that existing candidate selection methods are able to identify about 75% of the target keyphrases in the full text documents. Precision of candidate selection is very low, at about 1-2%. Ranking these candidates using *TFIDF* on the document full text, we achieve an MRR of 0.33, and precision at 10 of 0.1. Looking at the keyphrase content of more specific XML elements, we find high concentrations of keyphrases for example in title, abstract and bibliography. However, TFIDF does not appear to be useful for ranking phrases extracted from smaller elements.

For section structure we find that section type is a good indicator of keyphrase content of a section. The highest concentration of keyphrases is found in discussion sections, and for these we achieve highest performance when ranking phrases by TFIDF.

Future work will focus on exploring other ways of capturing the information contained in the document structure, and on combining features for keyphrase extraction.

Acknowledgements

This research was supported by the DuOMAn project carried out within the STEVIN programme which is funded by the Dutch and Flemish Governments (http://www.stevin-tst.org) under project number STE-09-12, and by the Netherlands Organisation for Scientific Research (NWO) under project numbers 017.001.190, 640.001.501, 640.002.501, 612.066.512, 612.061.814, 612.061.815, 640.004.802.

References

[1] T. D. Anderson. Studying human judgments of relevance: interactions in context. In *IIiX '06*, 2006.

[2] G. Crookes. Towards a validated analysis of scientific text structure. *Applied Linguistics*, 7(1):57–70, 1986.

[3] A. Dillon. Readers' models of text structures: the case of academic articles. *Intern. J. of Man-Machine Studies*, 35(6):913–925, 1991.

[4] S. El-Beltagy and A. Rafea. KP-Miner: A keyphrase extraction system for english and arabic documents. *Information Systems*, 34(1): 132–144, 2009.

[5] C. Gutwin. Improving browsing in digital libraries with keyphrase indexes. *Decision Support Systems*, 27(1-2):81–104, 1999.

[6] A. Hulth. *Combining Machine Learning and Natural Language Processing for Automatic Keyword Extraction*. PhD thesis, Stockholm University, 2004.

[7] T. Joyce and R. M. Needham. The thesaurus approach to information retrieval. *American Documentation*, 9(3):192–197, 1958.

[8] N. Kando. Text-level structure of research papers: Implications for text-based information processing systems. In *Proc. British Comp. Soc. Ann. Coll. Information Retrieval Research*, 1997.

[9] T. Nguyen and M.-Y. Kan. Keyphrase extraction in scientific publications. *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*, pages 317–326, 2007.

[10] N. Roberts. The pre-history of the information retrieval thesaurus. *J. of Documentation*, 40:271–285(15), 1984.

[11] P. K. Shah, C. Perez-Iratxeta, P. Bork, and M. A. Andrade. Information extraction from full text scientific articles: where are the keywords? *BMC Bioinformatics*, 4(1), 2003.

[12] T. Tomokiyo and M. Hurst. A language model approach to keyphrase extraction. In *Proc. ACL 2003 Workshop on Multiword expressions*, 2003.

[13] P. D. Turney. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2:303–336, 2000.

[14] J. Wang and H. Peng. Keyphrases extraction from web document by the least squares support vector machine. In *Web Intelligence '05*, 2005.

[15] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning. KEA: practical automatic keyphrase extraction. In *DL '99*, 1999.