Keyphrase Extraction in Scientific Publications

Thuy Dung Nguyen and Min-Yen Kan

Department of Computer Science, School of Computing,
National University of Singapore, Singapore, 117543
kanmy@comp.nus.edu.sg

Abstract. We present a keyphrase extraction algorithm for scientific publications. Different from previous work, we introduce features that capture the positions of phrases in document with respect to logical sections found in scientific discourse. We also introduce features that capture salient morphological phenomena found in scientific keyphrases, such as whether a candidate keyphrase is an acronyms or uses specific terminologically productive suffixes. We have implemented these features on top of a baseline feature set used by Kea [1]. In our evaluation using a corpus of 120 scientific publications multiply annotated for keyphrases, our system significantly outperformed Kea at the $p < .05$ level. As we know of no other existing multiply annotated keyphrase document collections, we have also made our evaluation corpus publicly available. We hope that this contribution will spur future comparative research.

1 Introduction

Keyphrases are defined as phrases that capture the main topics discussed in a document. As they offer a brief yet precise summary of a document content, they can be utilized for various applications. In an information retrieval (IR) environment, they serve as an indication of document relevance for users, as the list of keyphrases can quickly help determine whether a given document is relevant to their interest. As keyphrases reflect a document's main topics, they can be utilized to cluster documents into groups by measuring the overlap between the keyphrases assigned to them. Keyphrases also be used proactively in IR, in indexing. Good keyphrases supplement full-text indexing by assisting users in finding relevant documents.

Despite these known advantages of keyphrases, only a minority of documents have keyphrases assigned to them. This is because authors provide keyphrases only when they are instructed to do so [1], as manual assignment of keyphrases is expensive and time-consuming.

This need motivates research in finding automated approaches to keyphrase generation. Most existing automatic keyphrase generation programs view this task as a supervised machine learning classification task, where labeled keyphrases are used to learn a model of how true keyphrases differentiate themselves from other possible candidate phrases. The model is constructed using a set of features that capture the saliency of a phrase as a keyphrase.

In this work, we extend an existing state-of-the-art feature set with additional features that capture the logical position and additional morphological characteristics of

keyphrases. Unlike earlier work that aim for a domain-independant algorithm, our work is tailored to scientific publications, where keyphrases manifest domain-specific characteristics. With our extended feature set, we demonstrate a statistically significant performance improvement over the well-known Kea algorithm [1] for scientific publications.

We first review previous approaches in automatic keyphrase generation next. We then describe the overall methodology for our system is described in Section 3, which details our new features used to enhance the baseline feature set. Evaluation, including our compilation of a suitable multiply-annotated corpus, is detailed in Section 4.

2 Related Work

Work on keyphrase generation can be categorized into two major approaches: *extraction* and *assignment*.

Keyphrase Extraction. Keyphrase extraction methods select phrases present in the source document itself. Such approaches usually consist of a candidate identification stage and a selection stage.

In the candidate identification stage, systems restrict the number of candidate phrases for later consideration in order to bound the computational complexity of the latter selection stage. Most systems we surveyed place either a length or phrase type restriction (e.g., noun phrases only). Kim and Wilbur [2] study this stage in more depth, proposing three statistical techniques for identifying content bearing terms, by examining the distributional properties of a candidate versus its context. Tomokiyo and Hurst [3] take a language modeling approach to keyphrase generation by calculating the phraseness of a candidate, which represents the extent to which a word sequence is considered to have a phrasal quality.

The bulk of the work comes in the selection stage, where the program judges whether a candidate is a keyphrase or not. In a supervised learning scenario, this stage critically hinges on the features used to describe a candidate. Barker and Cornacchia [4] used three features to build their model: candidate word length, occurrence frequency, and head noun frequency. Turney's GenEx [5] system computed a vector of nine features to represent candidates. These features captured candidate length and frequency like Barker and Cornacchia's system, but additionally modeled the candidate's position within the document. Frank et al. [1] introduced Kea keyphrasing system. Although they pursued numerous features, their final feature set only used three independent features for classification: 1) the TF$\times$IDF score, 2) the position of the first occurrence, and 3) corpus keyphrase frequency, which measures how many times the candiate was assigned as a keyphrase in other training documents. Despite the reduced size of their feature set, Kea's performance is reported as comparable to GenEx.

Work by Turney [6] noted that candidate selection decisions are not independent. In other words, prior keyphrase selections should have an influence on the remaining selection decisions. He proposed to model the coherence of an entire set of candidate phrases using pointwise mutual information (PMI) between a candidate and k previously selected phrases. However, the PMI for these sets are difficult to obtain without sufficiently large datasets; Turney proposed using web search engine queries to obtain

rough collocation estimates, although this has marked drawbacks in terms of network bandwidth and time inefficiency.

Supervised text classification is not the only method for keyword extraction. Probabilistic topic models [7] treat documents as a mixture of topics and topics as a probability distributions over words. Thus, topic models can be considered as generative models for documents, and dually, given a document one can infer the topic(s) responsible for generating that document. While quite potent, topic models also rely on large amounts of training data, and are ineffective for small corpora.

Keyphrase Assignment. In contrast to extraction, *keyphrase assignment* is typically used when the set of possible keyphrases is limited to a known, fixed set, usually derived from a controlled vocabulary or set of subject headings. Here, binary classifiers can be trained for each keyphrase k in the set, and the assignment of keyphrases for a document is given by running all k classifiers and assigning those which indicate a positive result. In essence, keyphrase assignment is the same as traditional multiclass text classification.

For such approaches, as the keyphrases are known *a priori*, mutual information between the keyphrase and other words in the document can be used to do feature selection [8]. If the keyphrases form a ontology with broader, narrower and related term linkages, these relations can also be harnessed to provide additional evidence for inference [9]. Medelyan and Witten [10] used thesaural relations as edges to calculate the connectivity degree of a candidate keyphrase, showing that this feature (in conjunction with others) also statistically improved assignment accuracy. A drawback of the keyphrase assignment method is that it requires a large annotated corpus, as suitable number of training examples need to be found for each possible keyphrase.

3 Methodology

Given the current state of keyphrase generation, we chose to use an extraction based approach, as no suitable compilation of subject headings or ontology exists that aim to facilitate retrieval effectiveness. Extraction-based methods also generate a more diverse set of keyphrases, which we believe would better support relevance assessment. We also chose to use a supervised approach, as other methods require large amounts of annotated corpora, which we did not have.

Among the surveyed related work, the Kea algorithm fits this specification quite well. Kea uses just a few domain-independent features that have been shown to yield robust yet state-of-the-art results. For these reasons, we chose it as the baseline system for comparison.

In developing a keyphrase method for scientific publications, we note that such documents distinguish themselves from others based on their use of technical language as well as their rich document structure. As such, we have tried to capitalize on these features in modeling as well. Key enhancements in our work is to compute such additional features that model keyphrases in terms of their 1) morphological status and 2) document-centric structural character.

Figure 1 shows the outline of our system and highlights our new contributions to keyphrase extraction in gray. Like the baseline system Kea, our system follows a su-

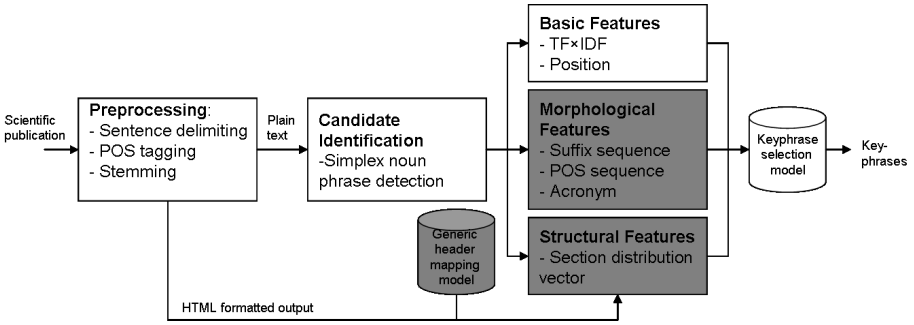

Fig. 1. System architecture. Contributions of this paper are highlighted in gray.

pervised machine learning approach. Training documents are used to generate linguistically motivated features and the extracted annotation from the training data serves as the class label $C = \{keyPhrase, \neg keyPhrase\}$.

Preprocessing is first done to convert the document from PDF to plain text and HTML formats, using the PDF995 utility suite. The plain text form is first processed to delimit sentences, then passed to a modern maximum entropy based part-of-speech (POS) tagger [11].

For candidate identification, all simplex noun phrases (i.e., ones without post modification, such as relative clauses and prepositional phrases) are deemed as keyphrase candidates. Case folding and stemming is also done to conflate statistics for variants, but only after the relevant morphological features for the individual candidate are calculated.

Candidate selection is the primary workhorse for keyphrase extraction. As stated, our key contribution is in introducing two additional sets of features that help to model the document structure of scientific publications as well as the characteristic terminological morphology. All extracted features (detailed in the next three subsections are used as evidence to create a keyphrase model using the standard Naïve Bayes learner implemented in the Weka machine learning toolkit [12].

3.1 Baseline feature set

We first review the two domain-independent features used by Kea and also in our enhanced system. Note that we did not use the **keyphrase frequency** feature of Kea, as this feature was reported only effective when sufficiently large training data is provided.

Term frequency $\times$ Inverse document frequency (TF$\times$IDF) - This is the standard salience metric used in information retrieval. Within a single document, frequently occurring terms are given high weight; over an entire corpus, terms that occur in few documents are given high weight. There are many specific formulations of tf$\times$idf; here we use a logarithm to dampen the inverse frequency term:

$$w_{ij} = \frac{f_{ij}}{max(f_{ij})} \times log_2 \frac{N}{df_i} \quad (1)$$

Position of first occurrence - This feature reflects the belief that keyphrases tend to appear at specific locations in the document (e.g., at the beginning). *Position* is calculated as the number of words that precede its first appearance, divided by the number of words in the document.

3.2 Extended structural features

Different logical sections of scientific publications contribute keyphrases at different rates. For example, few true keyphrases appear in experimental results but more occur in the *Abstract* or *Methods* sections. In a sense, the baseline position feature is a coarse-grained approximation of this, as academic publications tend to follow a consistent sequential structure: with an *Abstract*, followed by an *Introduction*, *Related Work*, *Methods*, *Evaluation*, *Conclusions* and *References*. We thus add an additional set of features to add this to our keyphrase model.

Section occurrence vector - We model the distribution of the keyphrase among different logical sections as a vector of frequency features for 14 generic section headers (as shown in Table 1. However, as headers in individual papers may deviate significantly from the norm (e.g., "Discussion" often should map to *Evaluation*, inferring how individual header instances map to generic headers is difficult. We created a maximum entropy (ME) based classifier that used four features – corresponding to 1) section number, 2) relative position, 3) previous section header and 4) current section header – to infer the generic section header (from our own list of 14 headers, as shown below in Table 1) for the input documents. The ME method was evaluated using ten fold cross validation on a corpus of 1020 annotated headers, garnering 938 correct assignments (92% accuracy). We also tried using the same features in a Hidden Markov Model (HMM) framework, but this only achieved 369 correct assignments, accruing a much lower accuracy (36%). We thus employ the ME version of the header mapper on an individual paper's headers (detected using orthography and numbering cues from the HTML converted format) to create the feature vector. Details of this header processing are omitted for space reasons; the interested reader is referred to the first author's thesis [13].

Abstract	Categories and Subject Descriptors	General Terms
Introduction	Background	Methods
Conclusions	References	Evaluation
Related Work	Acknowledgments	Applications
Motivation	Implementation	

Table 1. The 14 generic headers used by our logical section detection module.

3.3 Extended morphological features

Jones and Paynter's study [14] has validated claims that authors often do choose good keyphrases for their own documents. We thus analyzed author-provided keyphrases of

scientific publications to assess what characteristics a good keyphrase should possess. We focused on the linguistics characteristics of keyphrases assigned by authors.

POS sequence - We observed that almost all of the author assigned keyphrases are noun phrases, but whose part-of-speech tag sequence varies. For example, nominal modifiers to the headword feature occur more frequently than adjectival ones (e.g., "additive"/NN versus "additional"/JJ). This trend was observed for both bigram and trigram keyphrases. We use the POS tag sequence of the candidate as a single feature in our extended feature set.

Suffix sequence - In English, suffixes also hint at the terminological status of a candidate. Headwords of keyphrases manifest different suffix distributions than modifiers. We noticed that some suffixes such as $-ion, -ics, -ment$ often appear on headwords while others like $-ive, -al, -ic$ appear on modifiers. We use the sequence of morphological suffixes in a candidate as single feature. This feature partially overlaps with the POS sequence feature but is considerably more fine-grained.

Acronym status - Authors often introduce acronyms for phrases that are used many times in a document, saving space and making reference considerably easier. While there are considerably more sophisticated methods to detect acronyms, we found it sufficed to use use a simple approach. Our approach (Algorithm 1 scans for parenthetical expressions in the text and the preceding text can be considered a correspondance. We use a binary feature to indicate whether a candidate is an acronym.

Algorithm 1 Psuedocode for our simple acronym detection algorithm.

```
Retrieve all the texts T_1...T_N within parentheses () in document
for i = 1 to N do
  if length of T_i < 2 then
    Consider T_i as being neither acronym nor definition, continue
  end if
  if (T_i is in upper- or mixed-case) AND length of T_i < MAX then
    Assume T_i is an acronym
    Move toward the left to get its definition def_i
    if def_i exists then
      Record the acronym T_i and its definition def_i
    end if
  else
    Assume T_i is the definition
    Move toward the left to get its acronym acro_i
    if acro_i exists then
      Record the acronym acro_i and its definition T_i
    end if
  end if
end for
```

4 Evaluation

Two main approaches to evaluation present themselves. The first approach involves the manual evaluation of generated keyphrases. Here, subjects are given the document and the generated keyphrase list and asked to rank the relevance of each phrase. A disadvantage of this approach is that it requires manual effort, but more significantly, such an approach does not aid any subsequent evaluation, as the relevant assessment needs to be done from scratch every time. The second approach adopts the standard IR metrics of precision and recall to measure how well the generated keyphrases match a gold-standard assigned keyphrases. We take this second approach, but a question of how to come up with a gold standard arises.

4.1 Data Collection

Evaluating keyphrases has shown to be subjective and difficult. Jones and Paynter (2001) proved that author keyphrases are good representations of the subject of a document. However, generate keyphrase extraction evaluation requires multiple judgments and cannot rely merely on the single set of author-provided keyphrases [10]. Although author assigned keyphrases are usually viewed as a good representation of the subject of a document, they may not be able to cover all the good keyphrases in a document as keyphrase assignment is inherently subjective: keyphrases assigned by one annotator are not the only correct ones.

Unfortunately, we could not find a publicly available scientific document dataset tagged by multiple reliable annotators with keyphrases[1]. We thus constructed our own data set that fits these qualities for the evaluation of our algorithm.

We first found suitable publications and then collected keyphrases from manual annotators. We first used the Google SOAP API to find documents using variants of the query “keywords general terms filetype:pdf”. We downloaded over 250 of these PDF documents for futher processing. Documents were then manually restricted to scientific conference papers, with a length range of 4-12 pages. As our program only deals with textual input, we converted the PDF to plain text using the the PDF995 software suite as it handled two-columned text better than other programs tried. At the end of this process, we had 211 documents in plain text format which were converted successfully without problems.

We then recruited student volunteers from our department to participate in manual keyphrase assignment. Each volunteer was given three of PDF files (with author-assigned keyphrases hidden) to assign keyphrases to. To spur future research on automatic keyphrasing, we are making the full dataset and its details publicly available[2].

4.2 Results

For the experiments reported in this chapter, we used a subset of full dataset consisting of 120 documents, each of which has two keyphrase sets: one by the original author

[1] We considered a corpus of socially “tagged” papers from citeulike.org, but rejected this as authors occasionally choose keyphrases for purposes other than document description.

[2] `http://wing.comp.nus.edu.sg/downloads/keyphraseCorpus`

and the other by our volunteer. For each document, accuracy is the number of matches among keyphrases in the standard set and ten top-ranked extracted phrases.

To ensure clean separation between training and testing documents for our system and the trainable Kea baseline, all results reported here are obtained using ten-fold cross validation.

System	Average # of exact matches	Average score based on weight
Kea (baseline)	3.03	3.61
Our system	3.25 (0.024)	3.84 (0.033)

Table 2. Evaluation results. Statistical signficance over the baseline shown in parentheses (2-tailed paired t-tests).

Table 2 shows the average number of exact matches of the two algorithms with respect to the gold standard in the second column. Aside from an exact match of keyphrase in the gold standard, we can calculate a weighted match score based on the number of keyphrase sets in which the keyphrase appears. Let n be the number of keyphrases set in which a phrase p appears. Its weight $w(p)$ is computed as $w(p) = 1 + ln(n)$. A corresponding average matching score based on this weight is shown also in Table 2 as the third column.

We perform two-tailed paired t-test to see whether the improvements are significant. The corresponding p-values are also shown in the table, which indicate that the results are significant at the $p < 0.05$ level.

4.3 Error Analysis

We performed some post-experimental analysis of the errors created by both systems that lead to the generation of poor keyphrases. Our analysis leads to two problematic areas for future improvement. One difficulty is in deciding whether a general term is a good keyphrase or not. This can be seen in Table 3 document. Phrases such as “data” and “cell” are too general to be useful keyphrases. These phrases appear many times in the document, having high TF$\times$IDF scores, and also appear in important sections, such as the abstract and introduction, which results in their sectionrelated features are the same with those of correct keyphrases.

Another problem area is in generating suitable long keyphrases (i.e., phrases with three words or more). Currently, these are rarely generated by the current methodology. In the sample text, no three-or-more word phrases are generated among in the ten outputs, although they make up 5 of the 14 manually assigned keyphrases in the gold standard set.

5 Conclusion

We have presented an improved feature set for the problem of keyphrase extraction in scientific publications. The set adds features for representing logical position of the

Assigned keyphrases	Kea baseline	Our system
Neural network algorithm	Handover	Clusters
3G network	*Soft handover*	*Soft handover*
Visualization capability	3G	Data
Cluster analysis	Clusters	*3G network*
Self organizing map	*3G network*	Interesting clusters
Hierarchical clustering	Cell	Handover attempts
Key performance indicator of handover	Cell pairs	Method
Two-phase clustering algorithm	SHO	*Neural network*
Soft handover (2)	Active set	Measurements
Histograms	Handover measurements	Handover measurement
Decrease in computational complexity		
Mobility management		
Data mining		
neural networks		

Table 3. Author and generated keyphrases for the sample document *Analysis of Soft Handover Measurements in 3G Network* (36.pdf) in our keyphrase corpus. Only the "soft handover" keyphrase was provided by both the author and the volunteer annotator. Output keyphrases that match with assigned keyphrases are presented in italic font.

keyphrase instances with respect to sections of the document, and features to model whether a candidate phrases is an acronym or abbreviation, two salient sources of keyphrases in scientific discourse. Applying the new features in Naïve Bayes model does have a significant improvement against the state-of-the-art baseline Kea [1].

In evaluating our work, we have also compiled a corpus of more than 200 scientific publications, with multiple keyphrase sets. Each publication was annotated by volunteers to provide additional keyphrase coverage aside from the set provided by the original author. Such coverage is essential to the evaluation of keyphrase extraction algorithms in terms of coverage and importance of individual keyphrases. We have made this corpus publicly available and we believe that it will be useful in future work on keyphrase extraction.

Our current work focuses on deployment, in which we apply this keyphrase extraction module automatically over a large set of freely available scientific publications found on the web (i.e., CiteSeer). We are interested in merging such an automated facility with social user tagging. Future work on the extraction algorithm itself will focus on generating longer, more descriptive keyphrases, a key weakness as discussed in our error analysis.

References

1. Frank, E., Paynter, G.W., H.Witten, I., Gutwin, C., Nevill-Manning, C.G.: Domain specific keyphrase extraction. In: Proceedings of the 16th International Joint Conference on Artificial Intelligence. (1999) 668–673
2. Kim, W., Wilbur, W.J.: Corpus-based statistical screening for content-bearing terms. J. Am. Soc. Inf. Sci. Technol. **52** (2001) 247–259
3. Tomokiyo, T., Hurst, M.: A language model approach to keyphrase extraction. In: Proceedings of ACL Workshop on Multiword Expressions. (2003)
4. Barker, K., Cornacchia, N.: Using noun phrase heads to extract document keyphrases. In: Proc. of the 13th Biennial Conf. of the Canadian Society on Computational Studies of Intelligence, London, UK, Springer-Verlag (2000) 40–52

5. Turney, P.D.: Learning to extract keyphrases from text. Technical Report ERB-1057, National Research Council, Institute for Information Technology (1999)
6. Turney, P.D.: Coherent keyphrase extraction via web mining. In: Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03). (2003) 434–439
7. Steyvers, M., Griffiths, T.: Probabilistic topic models. In Landauer, T., Mcnamara, D., Dennis, S., Kintsch, W., eds.: Latent Semantic Analysis: A Road to Meaning. Laurence Erlbaum (2005)
8. Dumais, S.T., Platt, J., Hecherman, D., Sahami, M.: Inductive learning algorithms and representations for text categorization. In: Proc. of 7th International Conference on Information and Knowledge Management (CIKM). (1998) 148–155
9. Pouliquen, B., Steinberger, R., Ignat, C.: Automatic annotation of multilingual text collections with a conceptual thesaurus. In: BUG. (2003)
10. Medelyan, O., Witten, I.H.: Thesaurus based automatic keyphrase indexing. In: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries, New York, NY, USA, ACM Press (2006) 296–297
11. Ratnaparkhi, A.: A maximum entropy part of speech tagger. In: Proc. ACL-SIGDAT Conference on Empirical Methods in Natural Language Processing, Philadelphia (1996)
12. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques. 2nd edn. Morgan Kaufmann, San Francisco (2005)
13. Nguyen, T.D.: Automatic keyphrase generation. Technical report, National University of Singapore (2007)
14. Jones, S., Paynter, G.W.: Human evaluation of Kea, an automatic keyphrasing system. In: ACM/IEEE Joint Conference on Digital Libraries. (2001) 148–156