SJTULTLAB: Chunk Based Method for Keyphrase Extraction

Letian Wang
Department of
Computer Science & Engineering
Shanghai Jiao Tong University
Shanghai, China
koh@sjtu.edu.cn

Fang Li
Department of
Computer Science & Engineering
Shanghai Jiao Tong University
Shanghai, China
fli@sjtu.edu.cn

Abstract

In this paper we present a chunk based keyphrase extraction method for scientific articles. Different from most previous systems, supervised machine learning algorithms are not used in our system. Instead, document structure information is used to remove unimportant contents; Chunk extraction and filtering is used to reduce the quantity of candidates; Keywords are used to filter the candidates before generating final keyphrases. Our experimental results on test data show that the method works better than the baseline systems and is comparable with other known algorithms.

1 Introduction

Keyphrases are sequences of words which capture the main topics discussed in a document. Keyphrases are very useful in many natural language processing (NLP) applications such as document summarization, classification and clustering. But it is an expensive and time-consuming job for users to tag keyphrases of a document. These needs motivate methods for automatic keyphrase extraction.

Most existing algorithms for keyphrase extraction treat this task as a supervised classification task. The KEA algorithm (Gordon et al., 1999) identifies candidate keyphrases using lexical methods, calculates feature values for each candidate, and uses a machine-learning algorithm to predict which candidates are good keyphrases. A domain-specific method (Frank et al., 1999) was proposed based on the Naive Bayes learning scheme. Turney (Turney, 2000) treated a document as a set of phrases, which the learning algorithm must learn to classify as positive or negative examples of keyphrases. Turney (Turney, 2003) also presented enhancements to the KEA keyphrase extraction algorithm that are designed to increase the coherence of the extracted keyphrases. Nguyen and yen Kan (Nguyen and yen Kan, 2007) presented a keyphrase extraction algorithm for scientific publications. They also introduced two features that capture the positions of phrases and salient morphological phenomena. Wu and Agogino (Wu and Agogino, 2004) proposed an automated keyphrase extraction algorithm using a nondominated sorting multi-objective genetic algorithm. Kumar and Srinathan (Kumar and Srinathan, 2008) used n-gram filtration technique and weight of words for keyphrase extraction from scientific articles.

For this evaluation task, Kim and Kan (Kim and Kan, 2009) tackled two major issues in automatic keyphrase extraction using scientific articles: candidate selection and feature engineering. They also re-examined the existing features broadly used for the supervised approach.

Different from previous systems, our system uses a chunk based method to extract keyphrases from scientific articles. Domain-specific information is used to find out useful parts in a document. The chunk based method is used to extract candidates of keyphrases in a document. Keywords of a document are used to select keyphrases from candidates.

In the following, Section 2 will describe the architecture of the system. Section 3 will introduce functions and implementation of each part in the system. Experiment results will be showed in Section 4. The conclusion will be given in Section 5.

2 System Architecture

Figure 1 shows the architecture of our system. The system accepts a document as input (go through arrows with solid lines), then does the preprocessing job and identifies the structure of the document. After these two steps, the formatted document is sent to the candidate selection module

Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, pages 158–161,
Uppsala, Sweden, 15-16 July 2010. ©2010 Association for Computational Linguistics

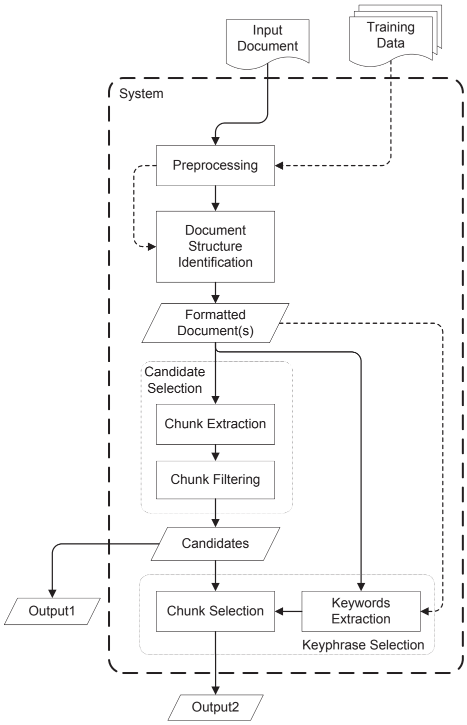

Figure 1: System architecture

which first extracts chunks from the document, then uses some rules to filter the extracted chunks. After candidate selection, the system will choose top fifteen (ordered by the position of the first occurrence in the original document) chunks from the candidates as the keyphrases and output the result ("Output1" in Figure 1) which is our submitted result. The candidates will also be sent to keyphrase selection module which first extracts keywords from the formatted document, then uses keywords to choose keyphrases from the candidates. Keywords extraction needs some training data (go through arrows with dotted lines) which also needs first two steps of our system. The result of keywords selection module will be sent to "Output2" as the final result after choosing top fifteen chunks.

OpenNLP[1] and KEA[2] are used in chunk extraction and keywords extraction respectively.

3 System Description

3.1 Preprocessing

In preprocessing, our system first deletes line breaks between each broken lines to reconnect the broken sentences while line breaks after title and section titles will be reserved. Title and section titles are recognized through some heuristic rules that title occupies first few lines of a document and section titles are started with numbers except abstract and reference. The system then deletes brackets blocks in the documents to make sure no keyphrases will be splitted by brackets blocks (e.g., the brackets in "natural language processing (NLP) applications" could be an obstacle to extracting phrase "natural language processing applications").

3.2 Document Structure Identification

Scientific articles often have similar structures which start with title, abstract and end with conclusion, reference. The structure information is used in our system to remove unimportant contents in the input document. Based on the analysis of training documents, we assume that each article can be divided into several parts: *Title*, *Abstract*, *Introduction*, *Related Work*, *Content*, *Experiment*, *Conclusion*, *Acknowledgement* and *Reference*, where *Content* often contains the description of theories, methods or algorithms.

To implement the identification of document structure, our system first maps each section title (including document title) to one of the parts in the document structure with some rules derived from the analysis of training documents. For each part except *Content*, we have a pattern to map the section titles. For example, the section title of *Abstract* should be equal to "abstract", the section title of *Introduction* should contain "introduction", the section title of *Related Work* should contain "related work" or "background", the section title of *Experiment* should contain "experiment", "result" or "evaluation", the section title of *Conclusion* should contain "conclusion" or "discussion". Section titles which do not match any of the patterns will be mapped to the *Content* part. After mapping section titles, the content between two section titles will be mapped to the same part as the first section title (e.g., the content between the section title "1. Introduction" and "2. Related Work" will be mapped to the *Introduction* part).

In our keyphrase analysis, we observed that most keyphrases appear in the first few parts of a document, such as *Title*, *Abstract*, and *Introduction*. We also found that parts like *Experiment*, *Acknowledgement* and *Reference* almost have no

[1] http://opennlp.sourceforge.net/

[2] http://nzdl.org/Kea/

keyphrases. Thus, *Experiment*, *Acknowledgement* and *Reference* are removed by our system and other parts are sorted in their original order and outputted as formatted document(s) (see in Figure 1) for further process.

3.3 Candidate Selection

The purpose of candidate selection is to find out potential keyphrases in a document. Traditional approaches just choose all the possible words sequences and filters them with part-of-speech tags. This approach may result in huge amount of candidates and lots of meaningless candidates for each document.

Our system uses chunk based method to solve these problems.

> "A chunk is a textual unit of adjacent word tokens which can be mutually linked through unambiguously identified dependency chains with no recourse to idiosyncratic lexical information."[3]

Our approach significantly reduces the quantity of candidates and keep the meanings of original documents. For example, for an article title, "Evaluating adaptive resource management for distributed real-time embedded systems", the traditional method will extract lots of meaningless candidates like "adaptive resource" and "distributed real-time", while our method just extract "adaptive resource management" and "distributed real-time embedded systems" as candidates.

3.3.1 Chunk Extraction

The first step of candidate selection is chunk extraction which extract chunks from a document. Four tools in `OpenNLP`, *SentenceDetector*, *Tokenizer*, *PosTagger* and *TreebankChunker*, are utilized in our system. The system first evokes *SentenceDetector* to split the formatted document into sentences. Then uses *Tokenizer* and *PosTagger* to label all the words with part-of-speech tag. At last, *TreebankChunker* is used to extract chunks from the document.

3.3.2 Chunk filtering

Not all the extracted chunks can be the candidates of keyphrases. Our system uses some heuristic rules to select candidates from extracted chunks. The types of rules range from statistic information to syntactic structures. The rules that our system uses are based on some traditional methods for candidate filtering. They are:

1. Any chunks in candidates should have less than 5 words.

2. Any single word chunks in candidates should be found at least twice in a document.

3. Any chunks in candidates should be noun phrases.

4. Any chunks in candidates must start with the word with the part-of-speech tag (defined in `OpenNLP`) NN, NNS, NNP, NNPS, JJ, JJR or JJS and end with the word with the part-of-speech tag NN, NNS, NNP or NNPS. Chunks that do not match these rules will be removed. Chunks that haven't been removed will be the candidate keyphrases of the document.

3.4 Keyphrase Selection

Our analysis shows that keywords are helpful to extract keyphrases from a document. Thus, keywords are used to select keyphrases from candidate chunks.

3.4.1 Keywords Extraction

`KEA` is a keyphrase extraction tool, it can also be used to extract keywords with some appropriate parameters. We observed that most keyphrases extracted by `KEA` only contain one word or two words which describe the key meaning of the document, even when the max length is set to 5 or more. There are four parameters to be set, in order to get best results, we set maximum length of a keyphrase to 2, minimum length of a keyphrase to 1, minimum occurrence of a phrase to 1 and number of keyphrases to extract to 30. Then, the output of the `KEA` system contains thirty keywords per document.

As showed in Figure 1, `KEA` needs training data (provided by the task owner). Our system uses formatted documents (generated by the first two steps of our system) of training data as the input training data to `KEA`.

3.4.2 Chunk Selection

After extracting thirty keywords from each document, our system uses these keywords to filter out non-keyphrase chunks from the candidates. The

[3] http://www.ilc.cnr.it/sparkle/wp1-prefinal/node24.html

system completes the task in two steps: 1) Remove candidates of a document that do not have any keywords of the document extracted by KEA; 2) Choose the top fifteen (ordered by the position of the first occurrence in the orginal document) keyphrases as the answer of a document ("Output2" in Figure 1).

4 Experiment Result

Table 1 shows the F-score of two outputs of our system and some baseline systems. The first three methods are the baselines provided by the task owner. TFIDF is an unsupervised method to rank the candidates based on TFIDF scores. NB and ME are supervised methods using Navie Bayes and maximum entropy in WEKA[4]. KEA refers to the KEA system with the parameters that can output the best results. OP1 is our system with the "Output1" as result and OP2 is our system with the "Output2" as result (see Figure 1). In second column, "R" means to use the reader-assigned keyphrases set as gold-standard data and "C" means to use both author-assigned and reader-assigned keyphrases sets as answers.

Method	by	Top05	Top10	Top15
TFIDF	R	10.44%	12.61%	12.87%
	C	11.19%	14.35%	15.10%
NB	R	9.86%	12.07%	12.65%
	C	10.89%	14.03%	14.70%
ME	R	9.86%	12.07%	12.65%
	C	10.89%	14.03%	14.70%
KEA	R	14.55%	17.24%	16.42%
	C	14.45%	17.68%	17.74%
OP1	R	**15.61%**	**17.60%**	**17.31%**
	C	**15.36%**	**18.41%**	**18.61%**
OP2	R	16.08%	18.42%	18.05%
	C	17.91%	20.52%	20.36%

Table 1: The comparison of F-score of our system with other systems.

From the table, we can see that, both two outputs of our system made an improvement over the baseline systems and got better results than the well known KEA system.

We submitted both results of OP1 and OP2 to the evaluation task. Because of some misunderstanding over the result upload system, only the result of OP1 (with bold style) was successfully submitted.

5 Conclusion

We proposed a chunk based method for keyphrase extraction in this paper. In our system, document structure information of scientific articles is used to pick up significant contents, chunk based candidate selection is used to reduce the quantity of candidates and reserve their original meanings, keywords are used to select keyphrases from a document. All these factors contribute to the result of our system.

References

Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-manning. 1999. Domain-specific keyphrase extraction. pages 668–673. Morgan Kaufmann Publishers.

Ian Witten Gordon, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-manning. 1999. Kea: Practical automatic keyphrase extraction. In *Proceedings of Digital Libraries 99 (DL'99*, pages 254–255. ACM Press.

Su Nam Kim and Min-Yen Kan. 2009. Re-examining automatic keyphrase extraction approaches in scientific articles. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications*, pages 9–16, Singapore, August. Association for Computational Linguistics.

Niraj Kumar and Kannan Srinathan. 2008. Automatic keyphrase extraction from scientific documents using n-gram filtration technique. In *DocEng '08: Proceeding of the eighth ACM symposium on Document engineering*, pages 199–208, New York, NY, USA. ACM.

Thuy Dung Nguyen and Min yen Kan. 2007. Keyphrase extraction in scientific publications. In *In Proc. of International Conference on Asian Digital Libraries (ICADL 07*, pages 317–326. Springer.

Peter Turney. 2000. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2:303–336.

Peter Turney. 2003. Coherent keyphrase extraction via web mining. In *In Proceedings of IJCAI*, pages 434–439.

Jia-Long Wu and Alice M. Agogino. 2004. Automating keyphrase extraction with multi-objective genetic algorithms. In *HICSS '04: Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 4*, page 40104.3, Washington, DC, USA. IEEE Computer Society.

[4] http://www.cs.waikato.ac.nz/ml/weka/