Keyphrase Extraction in Scientific Articles: A Supervised Approach

Pinaki BHASKAR[1] *Kishorjit NONGMEIKAPAM*[2] *Sivaji BANDYOPADHYAY*[1]
(1) Department of Computer Science and Engineering, Jadavpur University, Kolkata – 700032, India
(2) Department of Computer Science and Engineering, Manipur Institute of Technology, Manipur University, Imphal, India
pinaki.bhaskar@gmail.com, kishorjit.nogmeikapa@gmail.com, sivaji_cse_ju@yahoo.com

ABSTRACT

This paper contains the detailed approach of automatic extraction of Keyphrases from scientific articles (i.e. research paper) using supervised tool like Conditional Random Fields (CRF). Keyphrase is a word or set of words that describe the close relationship of content and context in the document. Keyphrases are sometimes topics of the document that represent the key ideas of the document. Automatic Keyphrase extraction is a very important module for the automatic systems like query or topic independent summarization, question-answering (QA), information retrieval (IR), document classification etc. The system was developed for the Task 5 of SemEval-2. The system is trained using 144 scientific articles and tested on 100 scientific articles. Different combinations of features have been used. With combined keywords i.e. both author-assigned and reader-assigned keyword sets as answers, the system shows a precision of 32.34%, recall of 33.09% and F-measure of 32.71% with top 15 candidates.

KEYWORDS : Keyphrase Extraction, Topic Extraction, Information Extraction, Summarization, Question Answering (QA), Document Classification.

Proceedings of COLING 2012: Demonstration Papers, pages 17–24,
COLING 2012, Mumbai, December 2012.

1 Introduction

Keyphrase is a word or set of words that describe the close relationship of content and context in the document. Keyphrases are sometimes simple nouns or noun phrases (NPs) that represent the key ideas of the document i.e. topic. Keyphrases can serve as a representative summary of the document and also serve as high quality index terms (Kim and Kan, 2009). Keyphrases can be used in various natural language processing (NLP) applications such as summarization (Bhaskar et al., 2012a, 2012b, 2010a, 2010b), information retrieval (IR) (Bhaskar et al., 2010c), question answering (QA) (Bhaskar et al., 2012c, 2012d; Pakray et al., 2011), document classification etc. Specially for the query or topic independent summarization system, it's a must needed module. Keyphrase extraction also plays an important role in Search engines.

Works on identification of keyphrase using noun phrase are reported in (Barker and Cornacchia, 2000). Noun phrases are extracted from a text using a base noun phrase skimmer and an off-the-shelf online dictionary.

Keyphrase Extraction Algorithm (KEA) was proposed in order to automatically extract keyphrase (Witten et al., 1999). The supervised learning methodologies have also been reported (Frank et al, 1999).

Some works have been done for automatic keywords extraction using CRF technique. A comparative study on the performance of the six keyword extraction models, i.e., CRF, SVM, MLR, Logit, BaseLine1 and BaseLine2 has been reported in (Chengzhi et al., 2008). The study shows that CRF based system outperforms SVM based system.

The CRF based Keyphrase extraction system is presented in Section 3. The system evaluation and error analysis are reported in Section 4 and the conclusion is drawn in the next section.

2 Preparing the System

2.1 Features Identification for the System

Selection of features is important in CRF. Features used in the system are,

F = {*Dependency, POS tag(s), Chunk, NE, TF range, Title, Abstract, Body, Reference, Stem of word,* W_{i-m}, …, W_{i-1}, W_i, W_{i+1},… , W_{i-n} }

The features are detailed as follows:

i) **Dependency parsing:** Some of the keyphrases are multiword. So relationship of verb with subject or object is to be identified through dependency parsing and thus used as a feature.

ii) **POS feature:** The Part of Speech (POS) tags of the preceding word, the current word and the following word are used as a feature in order to know the POS combination of a keyphrase.

iii) **Chunking:** Chunking is done to mark the Noun phrases and the Verb phrases since much of the keyphrases are noun phrases.

iv) **Named Entity (NE):** The Named Entity (NE) tag of the preceding word, the current word and the following word are used as a feature in order to know the named entity combination of a keyphrase.

v) **Term frequency (TF) range:** The maximum value of the term frequency (max_TF) is divided into five equal sizes (*size_of_range)* and each of the term frequency values is mapped to the appropriate range (0 to 4). The term frequency range value is used as a feature. i.e.

$$\text{size\_of\_range}=\frac{\text{max\_TF}}{5},$$

Thus, Table 1 shows the range representation. This is done to have uniformed values for the term frequency feature instead of random and scattered values.

Class	Range
0 to size_of_range	*0*
*size_of_range + 1 to 2*size_of_range*	*1*
*2*size_of_range + 1 to 3*size_of_range*	*2*
*3*size_of_range + 1 to 4*size_of_range*	*3*
*4*size_of_range + 1 to 5*size_of_range*	*4*

TABLE 1 – Term frequency (TF) range

vi) **Word in Title:** Every word is marked with T if found in the title else O to mark other. The title word feature is useful because the words in title have a high chance to be a keyphrase.

vii) **Word in Abstract:** Every word is marked with A if found in the abstracts else O to mark other. The abstract word feature is useful because the words in abstracts have a high chance to be a keyphrase.

viii) **Word in Body:** Every word is marked with B if found in the body of the text else O to mark other. It is a useful feature because words present in the body of the text are distinguished from other words in the document.

ix) **Word in Reference:** Every word is marked with R if found in the references else O to mark other. The reference word feature is useful because the words in references have a high chance to be a keyphrase.

x) **Stemming:** The Porter Stemmer algorithm is used to stem every word and the output stem for each word is used as a feature. This is because words in keyphrases can appear in different inflected forms.

xi) **Context word feature:** The preceding and the following word of the current word are considered as context feature since keyphrases can be a group of words.

2.2 Corpus Preparation

Automatic identification of keyphrases is our main task. In order to perform this task the data provided by the SEMEVAL-2 Task Id #5 is being used both for training and testing. In total 144 scientific articles or papers are provided for training and another 100 documents have been marked for testing. All the files are cleaned by placing spaces before and after every punctuation mark and removing the citations in the text. The author names appearing after the paper title was removed. In the reference section, only the paper or book title was kept and all other details were deleted.

3 CRF based Keyphrase Extraction System

3.1 Extraction of Positional Feature

One algorithm has been defined to extract the title from a document. Another algorithm has been defined to extract the positional feature of a word, i.e., whether the word is present in title, abstracts, body or in references.

Algorithm 1: Algorithm to extract the title.

Step 1: Read the line one by one from the beginning of the article until a '.'(dot) or '@' found in the line. ('.'(dot) occurs in author's name and '@' occurs in author's mail id).

Step 2: If '.' found first in a line then each line before it is extracted as Title and returned.

Step 3: If '@' found first in a line then extract all the line before it.

Step 4: Check the extracted line one by one from beginning.

Step 5: Take a line; extract all the words of that line. Check whether all the words are not repeated in the article (excluding the references) or not. If not then stop and extract all the previous lines as Title and return.

Algorithm 2: Algorithm to extract the Positional Features.

Step 1: Take each word from the article.

Step 2: Stem all the words.

Step 3: Check the position of the occurrence of the words.

Step 4: If the word occurs in the extracted title (using algorithm 1) of the article then mark it as 'T' else 'O' in title feature column.

Step 5: If the word occurs in between the word ABSTRACT and INTRODUCTION then mark it as 'A' else 'O' in abstracts feature column.

Step 6: If the word occurs in between the word INTRODUCTION and REFERENCES then mark it as 'B' else 'O' in body feature column.

Step 7: If the word occurs after the word REFERENCES then mark it as 'R' else 'O' in references feature column.

3.2 Generating Feature File for CRF

The features used in the keyphrase extraction system are identified in the following ways.

Step 1: The dependency parsing is done by the Stanford Parser[1]. The output of the parser is modified by making the word and the associated tags for every word appearing in a line.

Step 2: The same output is used for chunking and for every word it identifies whether the word is a part of a noun phrase or a verb phrase.

Step 3: The Stanford POS Tagger[2] is used for POS tagging of the documents.

[1] http://nlp.stanford.edu/software/lex-parser.shtml

Step 4: The term frequency (*TF*) range is identified as defined before.

Step 5: Using the algorithms described in Section 3.1 every word is marked as *T* or *O* for the title word feature, marked as *A* or *O* for the abstract word feature, marked as *B* or *O* for the body word feature and marked as *R* or *O* for the reference word feature.

Step 6: The Porter Stemming Algorithm[3] is used to identify the stem of every word that is used as another feature.

Step 7: In the training data with the combined keyphrases, the words that begin a keyphrase are marked with *B-KP* and words that are present intermediate in a keyphrase are marked as *I-KP*. All other words are marked as *O*. But for test data only *O* is marked in this column.

3.3 Training the CRF and Running Test Files

A template file is created in order to train the system using the feature file generated from the tanning set following the above procedure described in the Section 3.2. After training the C++ based CRF++ 0.53 package[4] which is readily available as open source for segmenting or labeling sequential data, a model file is produced. The model file is required to run the test files.

The feature file is again created from the test set using the above steps as outlined in Section 3.2 except the step 7. For test set the last feature column i.e. Keyphrase column, is marked with 'O'. This feature file is used with the C++ based CRF++ 0.53 package. After running the Test files into the system, the system produce the output file with the keyphrases marked with *B-KP* and *I-KP*. All the Keyphrases are extracted from the output file and stemmed using Porter Stemmer.

4 Evaluation and Error Analysis

The evaluation results of the CRF based keyphrase extraction system are shown in Table 3, where *P, R* and *F* mean micro-averaged precision, recall and F-scores. In second column, *R* denotes the use of the reader-assigned keyword set as gold-standard data and *C* denotes the use of combined keywords i.e. both author-assigned and reader-assigned keyword sets as answers. There are three sets of score. First set of score i.e. Top 5 candidates, is obtained by evaluating only top 5 keyphrases from evaluated data. Similarly Top 10 candidates set is obtained by evaluating top 10 keyphrases and Top 15 Candidates set result is obtained by evaluating all 15 keyphrases.

Team	By	Top 5 Candidates			Top 10 candidates			Top 15 candidates		
		P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
JU_CSE	R	36.33%	15.09%	21.33%	27.41%	22.76%	24.87%	22.54%	28.08%	25.01%
	C	52.08%	17.77%	26.49%	39.69%	27.07%	32.18%	32.34%	33.09%	**32.71%**

TABLE 3 – Result for JU_CSE system with CRF

[2] http://nlp.stanford.edu/software/tagger.shtml
[3] http://tartarus.org/~martin/PorterStemmer/
[4] http://crfpp.sourceforge.net/

The scores for the top 5 candidates and top 10 candidates of keyphrases extracted show a better precision score since the keyphrases are generally concentrated in the title and abstracts. The recall shows a contrast improvement from 17.77% to 33.09% as the number of candidate increases since the coverage of the text increases. The F-score is 32.71% when top 15 candidates are considered which is 17.61% i.e. 2.17 times better from the best baseline model with F-score of 15.10%. Different features have been tried and the best feature we have used in the system is:

F = {*Dependency*, POS_{i-1}, POS_i, POS_{i+1}, NE_{i-1}, NE_i, NE_{i+1}, *chunking, TF range, Title, Abstract, Body, Reference, Stem of word*,W_{i-1}, W_i, W_{i+1}}

Here, POS_{i-1}, POS_i and POS_{i+1} are the POS tags of the previous word, the current word and the following word respectively. Similarly W_{i-1}, W_i and W_{i+1} denote the previous word, the current word and the following word respectively. This POS_i and W_i give a contrasting result when only the word and the POS of the word are considered.

A better result could have been obtained if Term Frequency * Inverse Document Frequency (***TF*IDF)*** range is included (Frank et al., 1999; Witten et al., 1999). *TF*IDF* measures the document cohesion. The maximum value of the *TF*IDF* (max*_TF_IDF)* can be divided into five equal size (*size_of_range)* and each of the *TF*IDF* values is mapped to the appropriate range (0 to 4). i.e.

$$\text{size\_of\_range} = \frac{\text{max\_TF\_IDF}}{5},$$

We have used the Unigram template in the template file CRF++ 0.53 package but the use of bigram could have improved the score.

Conclusion and Perspectives

A CRF based approach to keyphrase extraction has been attempted in the present task for scientific articles. Proper cleaning of the input documents and identification of more appropriate features could have improved the score.

In future we will use MWE as a feature in CRF. Most of the cases the keyphrases are multi-word. We also like to port our system in different domains like news, tourism, health or general.

Acknowledgment

The work has been carried out with support from Department of Electronics and Information Technology (DeitY), MCIT, Govt. of India funded Project Development of "Cross Lingual Information Access (CLIA)" System Phase II.

References

Barker. K. and Corrnacchia. N. (2000). Using noun phrase heads to extract document keyphrases. In *the 13th Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence*, pages 40-52, Canada.

Bhaskar. P. and Bandyopadhyay. S. (2012a). Language Independent Query Focused Snippet Generation. In *the proceedings of CLEF 2012 Conference and Labs of the Evaluation Forum*, pages 140–142, Rome, Italy, T. Catarci et al. (Eds.): CLEF 2012, LNCS 7488, 2012, Springer-Verlag Berlin Heidelberg 2012.

Bhaskar. P. and Bandyopadhyay. S. (2012b). Cross Lingual Query Dependent Snippet Generation. In *International Journal of Computer Science and Information Technologies (IJCSIT)*, pages 4603 – 4609, ISSN: 0975-9646, Vol. 3, Issue 4.

Bhaskar. P., Banerjee. S., Neogi. S. and Bandyopadhyay. S. (2012c). A Hybrid QA System with Focused IR and Automatic Summarization for INEX 2011. In *Focused Retrieval of Content and Structure: 10th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2011*, Shlomo Geva, Jaap Kamps, and Ralf Schenkel, editors. Saarbruecken, Germany, Revised and Selected Papers. Volume 7424 of Lecture Notes in Computer Science. Springer Verlag, Berlin, Heidelberg.

Bhaskar. P. and Bandyopadhyay. S. (2012d). Answer Extraction of Comparative and Evaluative Question in Tourism Domain. In *International Journal of Computer Science and Information Technologies (IJCSIT)*, pages 4610 – 4616, ISSN: 0975-9646, Vol. 3, Issue 4.

Pakray. P., Bhaskar. P., Banerjee. S., Pal. B., Gelbukh. A. and Bandyopadhyay. S. (2011). A Hybrid Question Answering System based on Information Retrieval and Answer Validation. In *the proceedings of Question Answering for Machine Reading Evaluation (QA4MRE) at CLEF 2011*, Amsterdam.

Bhaskar. P. and Bandyopadhyay. S. (2010a). A Query Focused Automatic Multi Document Summarizer. In *the proceeding of the 8th International Conference on Natural Language Processing (ICON 2010)*, pages 241-250, IIT, Kharagpur, India.

Bhaskar. P. and Bandyopadhyay. S. (2010b). A Query Focused Multi Document Automatic Summarization. In *the proceedings of the 24th Pacific Asia Conference on Language, Information and Computation (PACLIC 24)*, Tohoku University, Sendai, Japan.

Bhaskar. P., Das. A., Pakray. P. and Bandyopadhyay. S. (2010c). Theme Based English and Bengali Ad-hoc Monolingual Information Retrieval in FIRE 2010. In *the proceedings of the Forum for Information Retrieval Evaluation (FIRE) – 2010*, Gandhinagar, India.

Davanzo. E. and Magnini. B. (2005). A Keyphrase-Based Approach to Summarization:the LAKE System at DUC-2005. In *the Document Understanding Conferences*.

Frank. E., Paynter. G., Witten.I., Gutwin. C. and Nevill-Manning. G. (1999). Domain-specific keyphrase extraction. In *the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 668-673, California.

Kim. S. N. and Kan. M. Y. (2009). Re-examining Automatic Keyphrase Extraction Approaches in Scientific Articles. In *the 2009 Workshop on multiword Expressions, ACL-IJCNLP 2009*, pages 9-16, Suntec, Singapore.

Lafferty. J., McCallum. A. and Pereira. F. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *the 18th International Conference on Machine Learning (ICML01)*, pages 282-289, Williamstown, MA, USA.

Turney. P. (1999). Learning to Extract Keyphrases from Text. In *the National Research Council, Institute for Information Technology, Technical Report ERB-1057*. (NRC \#41622).

Witten. I., Paynter. G., Frank. E., Gutwin. C. and Nevill-Manning. G. (1999). KEA:Practical Automtic Key phrase Extraction. In *the fourth ACM conference on Digital libraries*, pages 254-256.

Zhang, C., Wang, H., Liu, Y., Wu, D., Liao, Y. and Wang, B. (2008). Automatic keyword Extraction from Documents Using Conditional Random Fields. In *Journal of Computational Information Systems*, 4:3, pages 1169-1180.