A Ranking Approach to Keyphrase Extraction

Xin Jiang [*]
School of Mathematical
Sciences, Peking University
Haidian District
Beijing, China 100871
jxfeb5@gmail.com

Yunhua Hu
Microsoft Research Asia
No. 49 Zhichun Road
Haidian District
Beijing, China 100080
yuhu@microsoft.com

Hang Li
Microsoft Research Asia
No. 49 Zhichun Road
Haidian District
Beijing, China 100080
hangli@microsoft.com

ABSTRACT

This paper addresses the issue of automatically extracting keyphrases from a document. Previously, this problem was formalized as classification and learning methods for classification were utilized. This paper points out that it is more essential to cast the problem as ranking and employ a learning to rank method to perform the task. Specifically, it employs Ranking SVM, a state-of-art method of learning to rank, in keyphrase extraction. Experimental results on three datasets show that Ranking SVM significantly outperforms the baseline methods of SVM and Naive Bayes, indicating that it is better to exploit learning to rank techniques in keyphrase extraction.

Categories and Subject Descriptors

H.3 [**Information storage and retrieval**]: Content Analysis and Indexing

General Terms

Algorithms, Experimentation

Keywords

Keyphrase extraction, Learning to rank, Ranking SVM

1. INTRODUCTION

We consider automatic extraction of keyphrases from a document. Keyphrases of a document are the words and phrases that can precisely and compactly represent the content of the document. Much work was conducted on keyphrase extraction, with many of them leveraging machine learning techniques. The problem was basically formalized as a classification task in which we classify phrases in the document as *keyphrases* and *non-keyphrases*. Documents as well as manually labeled keyphrases were utilized as training data. Classification methods such as decision tree [3] and Naive Bayes [1] were employed.

Recently, 'learning to rank' technologies have been intensively studied. Different from classification and regression, the goal of learning to rank is to learn a function that can rank objects according to their degree of preference, importance, or relevance. So far, learning to rank methods were mainly applied to document retrieval.

In this paper, we apply Ranking SVM [2] to keyphrase extraction. Specifically, we take ranked phrase pairs as training examples, each of which consists of one keyphrase and one non-keyphrase, and construct an SVM model based on the training data. We then sort the candidate phrases of a new document with the trained model, and select the top ranked candidate phrases as keyphrases. We conducted experiments to verify the effectiveness of our approach with three datasets. The results show that Ranking SVM statistically significantly outperforms the baseline classification methods of SVM and Naive Bayes (KEA).

2. RANKING APPROACH

We point out that keyphrase extraction is by nature a ranking problem rather than a classification problem, and it is better to employ a learning to rank method than a classification method in the task. The reasons are as follows. First, it is more natural to consider the likelihood of a phrase's being a keyphrase in a relative sense than in an absolute sense. Second, information (features) for determining whether a phrase is a keyphrase is also relative. Classification methods use absolute feature values, while (pairwise) ranking methods use differences of feature values. It is the differences that have the discriminative power across documents.

In this paper, we employ Linear Ranking SVM. As baseline, we also use Linear SVM. The two methods actually use the same set of features, but different feature values. To test the discriminative abilities of the two methods, we conducted analysis on some randomly selected data. Figure 1 (a)(b) shows the scatterplots of the instances in a two-feature space (TF-IDF score of phrase and position of phrase's first occurrence in document) for the classification and ranking data. We can see that it would be easier for the ranking approach to separate the positive and negative instances. We further conducted a linear discriminant analysis on the data using all features. Figure 1 (c)(d) shows the distributions of data projected on the discriminant direction (direction with the optimal classification separability) for classification and ranking. The overlaps between the densities clearly indicate that instances are much more separable in the ranking case than in classification case. Therefore,

[*] This work was conducted at Microsoft Research Asia when the first author visited there.

SIGIR'09, July 19–23, 2009, Boston, Massachusetts, USA.

Table 1: Keyphrase extraction performances of three methods on three datasets

	Research Paper			Social Tagging			TREC .Gov		
Method	MAP	P@1	P@5	Kendall's Tau	MAP	P@5	Kendall's Tau	MAP	P@5
KEA	0.221	0.249	0.158	0.189	0.431	0.437	0.435	0.552	0.463
SVM	0.279	0.273	0.192	0.222	0.465	0.469	0.445	0.628	0.537
Ranking SVM (pointwise)	**0.288**	**0.311**	**0.195**	**0.250**	**0.503**	**0.495**	0.460	**0.641**	**0.545**
Ranking SVM (pairwise)	-	-	-	-	-	-	**0.512**	0.575	0.498

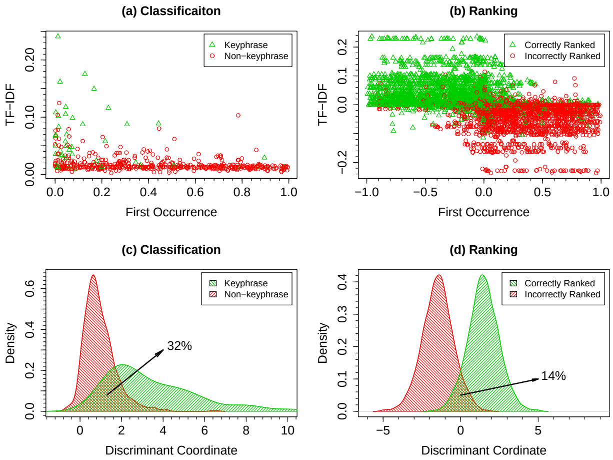

Figure 1: Distributions of instances in classification and pairwise ranking

this indicates that the ranking approach is more likely to achieve better performance than the classification approach in keyphrase extraction.

Our keyphrase extraction method consists of the following steps. First, we use a baseNP identification method [4] to extract all the base noun phrases in the document as candidate phrases. Next, we create a feature vector for each candidate phrase. The features include TF-IDF score, phrase length, position of phrase's first occurrence, phrase's appearance in document title, uniformity of phrase's distribution within document (measured by entropy), frequency of most and least frequent word of phrase, etc. In training, we label keyphrases for randomly selected documents, and employ Ranking SVM to train a ranking function (Linear SVM). In extraction, we utilize the ranking model to rank candidate phrases from a new document and output the top ranked phrases as keyphrases.

3. EXPERIMENTS

We conducted experiments on keyphrase extraction using three datasets.

The first dataset consists of 341 research papers with author-provided keyphrases randomly selected from an academic search engine.

The second dataset was collected from a social tagging website. We randomly selected URL's from the URL list provided at the site and crawled their original web pages (600 web pages in total). Each web page is associated with a list of tags (single words) added by web users. In addition, frequency information of tags is also available, which reflects the popularity of the tags. In classification, the tagged words were treated as keywords. In ranking, candidate words were ranked by frequency and used as ground truth.

For the third dataset, we randomly selected 300 web pages from the TREC .Gov dataset and asked human annotators to label the keyphrases in them. We designed two annotation methods, referred to as *pointwise method* and *pairwise method*. For the pointwise method, 50 candidate phrases were extracted from each document, and the annotators were asked to pick up keyphrases from the candidates. For the pairwise method, 250 pairs of phrases were extracted from each document, and the annotators were asked to identify which phrase is more likely to be a keyphrase than the other. Six annotators participated in the annotation task. Final evaluation results were obtained by majority voting on the judgements. The pointwise data can be used as training and test data for both classification and ranking, while the pairwise data only for ranking.

Three methods, Ranking SVM as well as the two baseline classification methods: Naive Bayes (KEA)[1] and SVM[2], were tested on the datasets. During the learning process, each dataset was separated into training data and test data. The methods were trained with the training data and then evaluated with the test data. In evaluation, the keyphrases of each document were generated by Ranking SVM and the baselines, and the performances of keyphrase extraction were evaluated in three measures: Precision at position n (denoted as P@n), Mean Average Precision (MAP) and Kendall's Tau.

Table 1 shows the 3-fold cross validation experimental results on the three datasets using the three learning methods. From the table, we can see that for all three datasets, Ranking SVM outperforms SVM, and significantly outperforms KEA in terms of all on measures. The improvements are all statistically significant (p-value < 0.05) in sign test. We conclude that the ranking approach can achieve significantly better performance than the classification approach for keyphrase extraction.

4. REFERENCES

[1] E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning. Domain-specific keyphrase extraction. In *IJCAI '99*, pages 668–673, 1999.

[2] R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. In *Advances in Large Margin Classifiers*, pages 115–132. MIT Press, 2000.

[3] P. D. Turney. Learning to extract keyphrases from text. Technical Report ERB-1057, National Research Council, Institute for Information Technology, 2000.

[4] E. Xun, C. Huang, and M. Zhou. A unified statistical model for the identification of English baseNP. In *ACL '00*, pages 109–116.

[1] KEA: http://www.nzdl.org/Kea/

[2] SVM^{light}: http://svmlight.joachims.org/