# 基于LDA的关键词抽取方法

翁伟

北京大学 教育部计算语言学重点实验室

Email: wei.weng@pku.edu.cn

王厚峰

北京大学 教育部计算语言学重点实验室

Email:wanghf@pku.edu.cn

摘要:这篇论文介绍了一个新颖的关键词组提取方法。该方法使用了 LDA 模型。方法通过 LDA 模型来获得文档的主题信息，通过将这些信息与其它特征信息整合起来，给短语进行了打分，分数高的被挑选为关键词组。实验结果体现了该方法的有效性和可行性。

关键词：关键词提取，LDA，无指导

# Automatic Keyphrase Extraction Based on LDA

Wei Weng

Key Laboratory of Computational Linguistics (Peking University) Ministry of Education, China

Email:wei.weng@pku.edu.cn

Houfeng Wang

Key Laboratory of Computational Linguistics (Peking University) Ministry of Education, China

Email:wanghf@pku.edu.cn

Abstract: This paper proposes a novel approach to extract keyphrases by making use of LDA. This method is implemented by employing topic models to get the topic information, and then combining this information with other features to score all the phrases. Those ranked at the top are selected as keyphrases. Experimental results demonstrate encouraging performance of the proposed approach.

Keywords: Keyphrase extraction, LDA, unsupervised.

## 1 Introduction

Keyphrases are becoming increasingly important in our information era. Firstly, they can facilitate user's fast reading and browsing. Secondly, they can be used in the following NLP tasks: document classification [Krulwich and Burkey, 1996] document clustering [Hammouda 2005] and so on.

Despite the significance of keyphrases, vast majority of documents do not have keyphrases, therefore it is necessary to automatically extract a few keyphrases from a given document to express the main idea of the document. Existing methods can be divided into two categories: supervised and unsupervised.

In supervised approach, keyphrase extraction is regarded as a classification task [Turnkey 1999]. By considering their statistical and linguistic features, a model is trained to predict whether a candidate is keyphrase or not. This method requires a document set with human-assigned keyphrases. However, human labeling is time-consuming. So, in this study, we prefer unsupervised approach.

One existing unsupervised approach view keyphrase extraction as a ranking task [Miracle and Tarau, 2004]. Document is represented as a term graph based on term relatedness, and then page-rank algorithm is used to assign score to each candidate term. The occurrences of terms within a specified window size in the document are often used as a way to compute term relatedness. Wan improves this approach by taking related document into consideration [wan 2008a]. In his method, documents are firstly clustered into a document set. Term scores are then obtained by applying a

page-rank algorithm on the whole set. After that, keyphrases are selected on one document according to that global score.

Another unsupervised approach firstly cluster all the terms [Liu 2009], and then select exemplar terms from each cluster to find keyphrases. In this method, three important properties of keyphrases are brought up: Understandable, Relevant, Good Coverage.

Our method considers these three issues by using LDA, which can help us to find the hidden topic structure. The idea behind LDA is to model documents as arising from multiple topics, where a topic is defined to be a distribution of over a fixed vocabulary of terms [Bali 2009].

The idea of finding keyphrases using topic models is borrowed from human's perception: To know the keyphrases of a particular document, we firstly need to know what the article is mainly about (topics). Then the words representing those topics can be extracted as keyphrases. So this method can yield keyphrases that have all those three properties:

- Relevant to document (topic analysis)
- Good Coverage (keyphrases are selected from all topics)
- Understandable (Filtering and phrase formulating rules are used to ensure this property

## 2 System Overview

In our system, there are 4 major modules. The first module is aiming to process the text so that it can be used as the input of LDA training. The second module analyzes the result and compute score for each word. The third module chooses appropriate combination to generate phrases. The last module outputs the final result set.

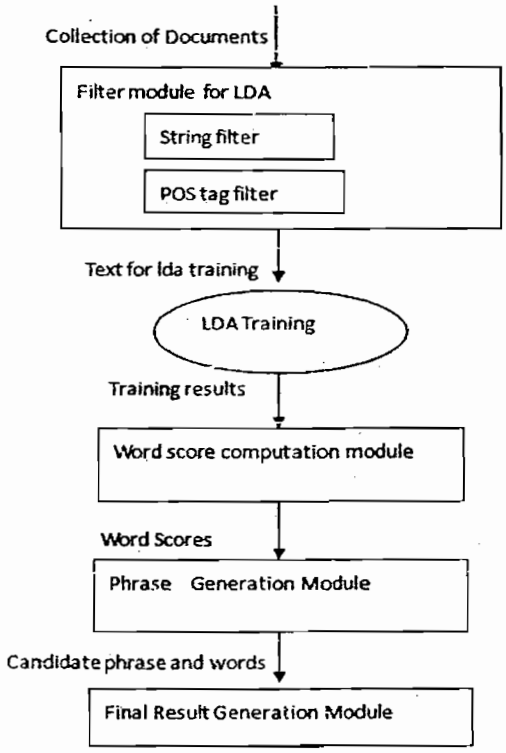


Fig 1

## 3 System Design

### 3.1 Filter Module for LDA

For the training of LDA, each document is represented as a bag of words. The main task for this module is to choose words that can express actual meanings. Considering single-character word and word with certain POS tags are often meaningless words. We adopt following filtering rules:

- All the single character word is filtered.
- Words that are prepositions and conjunctions are filtered.

### 3.2 Word Score Computation Module

The graphical model representation of LDA is as follows:

Nodes denote random variables, edges denotes random variables, Shaded nodes denote observed random variable; unshaded nodes denote hidden random variables.[Bali 2009]

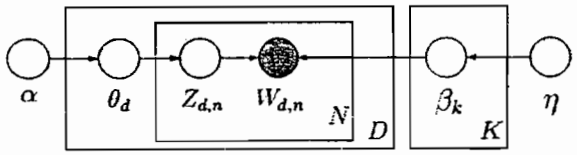


Fig 2

To compute the score of a particular word, we use the following information that this model outputs:

- Topic-document distribution.
- topic assignment for each word

Then a basic score of each word is computed by using the following methods:

For a word w in document d, s(w) represents the score of that word, $t_i$ means the probability of particular topic, $C_i$ means the total times that the word has been assigned to that topic. Then score is computed in this formula:

$$S(w)=\sum_0^i t_i * C_i$$

### 3.3 Phrase Generation Module

In this stage, phrase is defined as a Chinese character string sequence that has occurred in the text containing two or more Chinese words. To narrow down the scope of selecting phrases, only the words ranks at the top are taken into consideration for forming phrase. If we want to choose N keyphrases, the top 2N words are taken into consideration to form phrases. Very long phrase is not commonly considered to be keyphrases, so we only consider bigram and trigram here. All possible phrases (combinations of those 2n words) are scored in this way:

S(p)=F*sum of the score of its word.

F is the frequency of the phrase which appears in the text.

Then we filter those phrases by applying those two rules:

1. For those candidates whose score is less or equal to zero will be filtered.
2. Sorting those candidates in a descent order by its score. A set containing valid phrase is maintained. Each phrase is checked to be added into that set according to that descent order. The

rule for checking into that set is as follows: if no word that constitutes that phrase has appeared in any phrase in valid set, then the phrase should be checked in.

### 3.4 Final Result Generation Module

After all valid phrases have been selected. The final result set is obtained in this way: the total number of result is given, so the total number of result is fixed. We set a threshold on the score of the phrases to check whether it should be in the final result set. The remaining positions are left for word. We then check along the ranking list of word in a descended order. If one word has appeared in the phrase that we have chosen, we will go to check the next one. If it has not, the word is checked into the final result. This procedure goes on until the number of keyphrases in the result set reaches the expected value.

## 4 Experiments and Result Analysis

We conduct the experiment on a document set containing 5430 news articles. All of them are news reports on a website (news.163.com) from March to April in 2009. We have not tried many sets of parameters on the LDA model training. For the hyper parameter of LDA, alpha is set to be 0.5 and beta is set to be 0.1. The total topic number is set to be 80 and iteration steps are 15000.

Table 1 illustrates three 'topics' (i.e., highly probable words) that were discovered from this collection of news articles. As we can see, all of them have shown a strong connection and demonstrate a meaningful theme.

212 documents in the collection are with human-annotated keyphrases. Each one of them has been given keyphrases by three post-graduate students independently. To evaluate the results, we adopt the following measure: we consider a keyphrase to be correct if it has been labeled as keyphrase by any of the human-annotators. Then the preci- sion and recall are computed by the following formula:

| Topic 0 | Topic 2 | Topic 11 |
|---|---|---|
| 日本 | 经济 | 儿童 |
| 麻生 | 中国 | 手足 |
| 首相 | 增长 | 病毒 |
| 自民党 | 政策 | 死亡 |
| 大臣 | 投资 | 金字塔 |
| 内阁 | 危机 | 报告 |
| 福田 | 发展 | 感染 |
| 总裁 | 问题 | 病例 |
| 报道 | 市场 | 症状 |
| 东京 | 金融 | 阜阳 |
| 选举 | 企业 | 艾滋病 |
| 防卫 | 消费 | 疾病 |
| 政治 | 政府 | 疫情 |
| 支持率 | 影响 | 卫生 |
| 民主党 | 出现 | 患者 |
| 举行 | 出口 | 救治 |
| 国会 | 需求 | 卫生厅 |

**Table 1**

- P= number of correct phrases recognized/number of keyphrases extracted automatically
- R = number of correct phrase recognized / number of unique keyphrases label by human annotators.

The final result is shown in the following table:

| Precision | Recall |
|---|---|
| 0.403 | 0.565 |

**Table 2**

It has to be pointed out that although the precision and recall values are relatively poor comparing to some other NLP tasks, it does not indicate that the performance is poor, because there are no absolute answers for keyphrase. As described in [Wan 2008b], when two annotators were asked to label keyphrase on 308 documents, the Kappa statistic for measuring interagreement among them was only 0.70.

According to our observation of some wrong cases, some keyphrases extracted are the same as manually labeled ones in meaning, but to be counted as incorrect due to only one or two characters mismatched.

Another common problem is failing to recognize names of places, organizations and people. This is limited by the Chinese word segmentation technologies available

## 5 Conclusions and Future Work

In this paper, we propose a novel approach for document keyphrase extraction, which uses topic model. This method does not require any training steps and is simple and efficient in computing.

In future work, we will try to consider more features to improve the scoring system. NP chunks can be used to indentify candidates more accurately .We also tries to explore the feasibility of using this method on single document keyphrase extraction.

## References


[Krulwich and Burkey, 1996] B.Krulwich and C.Burkey. 1996 .Learning user information interests through the extraction of semantically significant phrase. In AAAI 1996 Spring Symposium on Machine Learning in Information Access

[Hammouda 2005] K.M.Hammouda, D.N.Matute and M.S.Kamel.2005 Corephrase: Keyphrase extraction for document clustering. In Proceedings of MLDM2005

[Turney 1999] Peter.D.Turney 1999 Learning to extract Keyphrases from text. National Research Council Canada, Institute for information Technology

[Mihalcea and Tarau,2004] Rada Mihalcea and Paul Tarau.2004. Textrank: Bringing order into texts. In Proceedings of the 2004 Conference On Empirical Methods in Natural Language Processing

[Wan 2008a] Xiaojun wan and Jianguo Xiao. Colabrank: Towards a collaborative approach to single-document keyphrase extraction. In Proceedings of COLING, pages 969-976

[Wan 2008b] Xiaojun wan and Jianguo Xia. Single document keyphrase extraction using neighborhood knowledge. In Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, pages 855-860

[Liu 2009] Zhiyuan Liu, Peng Li. Clustering to Find Exemplar Terms for Keyphrase Extraction. Proceedings of the 2009 Conference on Empirical Method in Natural Language Processing, pages 257-266

[Blei 2009] David M Blei, John D.Lafferty. Topic Models