A Practical Keyword Recommendation Method based on Probability in Digital Publication Domain

Yuejun Li[1,2], Xiao Feng[1], Shuwu Zhang[1]

[1]Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[2]Department of Computer Science and Technology, Shandong Jianzhu University, Jinan, 250101, China

273253612@qq.com, xiao.feng@ia.ac.cn, shuwu.zhang@ia.ac.cn

Abstract. The increase of information and knowledge has brought great challenge in knowledge management which includes knowledge storage, information retrieval and knowledge sharing. In digital publication domain, books are segmented into items that focus on target topic for dynamic digital publication. The management of items has great need to annotate items automatically instead of annotating by editor manually. This paper proposed probability based and hybrid method to recommend meaningful keywords for items. Experiment shows that the methods we proposed get more than 90% precision, recall and f1 value on the digital publication dataset which outperforms the traditional extraction based and tfidf similarity based method in keyword recommendation.

Keywords: keywords recommendation; probability based keywords recommendation; digital library

1 Introduction

Keywords summarize a document concisely and give a high-level description of the document content [1]. Keyword has been used in various domains which include document summarization [2], document classification, document clustering [3], document retrieval, topic search, and document analysis [1]. As the development of digital libraries and publication, there is a need to assemble new books or resources by taking advantage of books which having been published and stored in digital libraries. Editors of the press segment the book into hundreds of items which is subject to one's own topic. As each item is to some extent semantically independent from each other and correspond to one topic or more topics, previously, editors need to assign several keywords to the item manually according to its meaning which is a time-consuming process. When more new books come, the workload of editors to designate new keywords be-

comes heavier. So an automatic keyword recommendation mechanism is needed to faster the process of making items of books.

Numerous methods have been proposed to automatically extract keywords from a text. Keyword extraction technique tries to extract words that can summarize the text mostly which means the keyword must come from the text content.

In digital libraries and publication domain, there are huge amount of books accumulated that can be used as corpus of developing automatic keyword recommendation system. The problem is when a new item comes, what keywords should be recommended. Traditional keyword extraction method like the TextRank [4] can extract important words from item, but in many cases, the keyword may even not appear in the content of the item. Under such circumstances, traditional keyword extraction can't solve the problem independently and it needs a supervised method to conduct the keyword recommendation process. The focus of this paper is to solve the item keyword recommendation problem using supervised keyword recommendation algorithm.

2 Keyword Extraction Methods

Earlier techniques mainly focus on the word frequencies of the text or the TFIDF values to determine the weight of the candidate words [5]. Although the frequency of word can imply the importance of the word in some cases, there are still some cases that the important words appear only few times. To overcome the problem of frequency-based keyword extraction method, graph-based method is proposed which is inspired by the PageRank [6] via building network of words/phrases and ranking the node using some kind of centrality measure, variants of the graph based method include [2] and HITS[7]. Semantic method is supposed to bring meaningful information to keyword extraction. Semantic relation of words can be found with help of WordNet or Wikipedia and HowNet to recommend semantically similar words of original words in the text. Topic modeling methods which include LSA, PLSA, and LDA are used to mind hidden topic to improve accuracy and coverage ability of keywords.

Keyword extraction can be formulated as a supervised classification problem. The word or phrase to be classified is represented as a vector of features which may include tf-idf [1] values, length or occurrence position [8]. A training set which is annotated as positive and negative should be provided, and during the testing phase, the candidate keywords should be formulated as a feature vector to be classified. Variants of machine learning method are used which include SVM[1], decision trees, conditional random fields [9]. The shortcoming of the supervised method is that it needs a manually constructed training set which is time-consuming and hard to get.

3 TFIDF-Similarity based Keyword Recommendation

Traditional extraction method can extract keywords from the content itself. But when the content of the document is not long enough it will be difficult to extract useful keywords from the document directly. Recommending existed keywords to new documents can be implemented with the help of tfidf similarity based keyword recommendation technology which is described in [10].

Given a document set $D\{d_1,d_2,...,d_n\}$, every document is annotated with several keywords: $d_i=\{text,tagset\}$, where the tagset comprised of several keywords and all the keywords form a keyword library T. Once there comes a new document q, we need to recommend proper existed keywords based on its content. The process can be described in two steps:

Step1. Compute P(t|q,T,D) , which is the probability of every keyword in keyword library T, comparing the new document q with document in D.

Step2. Sort P(t|q,T,D) in descending order and select the top k keywords as the final recommendation.

P(t|q,T,D) can be formulated as follows:

$$P(t\mid q,T,D)=\frac{keyWeight(t,q,D)}{\sum_{t\in T}keyWeight(t,q,D)} \tag{1}$$

Where $keyWeight(t,q,D)$ is the weight of keyword t according to the similarity of document t with all document in D.

$$keyWeight(t,q,D)=\sum_{d\in D}DocSim(q,d)\times isTag(t,d) \tag{2}$$

$DocSim(q,d)$ is the similarity of new document q and document d of corpus D and we select the cosine similarity measure to compute similarity.

$$DocSim(q,d)=\frac{q\cdot d}{\|q\|\times\|d\|}=\frac{\sum_{i=1}^{n}q_i\times d_i}{\sqrt{\sum_{j=1}^{n}q_i^2}\times\sqrt{\sum_{j=1}^{n}d_i^2}} \tag{3}$$

$$isTag(t,d)=\begin{cases}1 & t\in d\\ 0 & t\notin d\end{cases} \tag{4}$$

When d is annotated with keyword t, then isTag(t,d) is given the value 1, otherwise is given to 0.

The q and d vector is the TF-IDF value of each word in document q and d.

$$q=(tfidf(w_1),tfidf(w_2),...tfidf(w_n)) \tag{5}$$

4 Probability-based Keyword recommendation

4.1 Problem definition

Our keyword recommendation method of items in dynamic publication domain can be formulated as follows. The training set is composed of items annotated with keywords by editors.

TraininSet={[Tags(1),Item(1),ClassId(1)],[Tags(2),Item(2),ClassId(2)],…,[Tags(i), Item(i),ClassId(i)),…, [Tags(n), Item(n), ClassId(n)]},

where Tags(i) is the keyword set assigned to item i by editors. Tags(i) = (key(1), key(2), … , key(m)) where key(i) is the keyword and the keyword number m varies from one to ten or more. In digital publication areas, the keyword number for each item often varies from 3 to 5. ClassId(i) is the category id of item i which suggests that the item belongs to the class i. We utilize the text classify technology to classify the item first in order to narrow the range of recommending keywords because the training set is usually very large, direct keyword recommendation would face the problem that there are thousands of keywords to be evaluated and to find the best keywords in them is a difficult thing. Due to fact that the items of the training set have the information of classification, when a new item comes, through the process of classification, an item first can be classified to proper category, and then the keywords in the category can be recommended to the new item base on its content.

The process can be described as follows: First we run the classification algorithm to find the category of item with the result category k; Given a new item, our aim is to compute the probability of every keyword in category k and it can be described as follows: we compute the probability $p(k_i \mid item)$, where k_i is the keyword from category k. Then we sort the list of $p(k_i \mid item)$ and select the top k as the candidate keywords of item.

4.2 Probabilistic Modeling

The Bayes probability theory is used in modeling the probability of keywords k_i that mostly delegate the item. Given a new item we need to compute the probability $p(k_i \mid item)$:

$$p(k_i \mid item) = \frac{p(item \mid k_i) \times p(k_i)}{p(item)} \tag{6}$$

The probability of every new item is no different from each other, so we can ignore the probability $p(item)$, and the probability

$$p(k_i \mid item) \propto p(item \mid k_i) \times p(k_i) \tag{7}$$

Every item is a fragment of text composed of words/phrases, and we make a hypothesis that every word/phrases is independent from each other which we called bag of words model. The probability of item given the keyword k_i can be calculated as follows:

$$p(item \mid k_i) = \prod_{j=1}^{m} p(w_j \mid k_i) \tag{8}$$

where w_j is the term of item, and $p(w_j \mid k_i)$ is the probability of every term w_j of item when annotated keyword k_i occurs.

We models the probability $p(w_j \mid k_i)$ below which is different from that in [11] and more efficient in experiment result.

$$p(w_j \mid k_i) \propto \frac{tfidf(w_j) \times tf(w_j, k_i)}{p(k_i) \times \sqrt{\sum_{j=1}^{m} tf^2(w_j, k_i)}} \tag{9}$$

$tfidf(w_j)$ is the weight of term w_j which can be computed by the typical tfidf formulae.

$$tfidf(w_j) = tf(w_j) \times idf(w_j) \tag{10}$$

Where $tf(w_j)$ is term frequency of term w_j in item and $idf(w_j)$ is the inverse document frequency of term w_j.

$tf(w_j, k_i)$ is the term frequency of w_j in keywords k_i annotated items. $p(k_i)$ is the probability of keyword k_i in all the training set TR of category j and it can be computed as follows:

$$p(k_i) = \frac{tf(k_i, TR_j)}{\sum_i tf(k_i, TR_j)} \tag{11}$$

Where $tf(k_i, TR_j)$ is the term frequency of keywords k_i in the training set TR_j of category j.

In the training period, we first calculate the probability of $p(k_i)$, $p(w_j \mid k_i)$, and stored the result to compute every $p(k_i \mid item)$ of each candidate keyword. Finally we sort $p(k_i \mid item)$ in descending order and select the top N keywords as the final recommendation.

5 A Hybrid Approach of Keyword Recommendation

The tfidf similarity based method and probability based method can utilize previous annotated keywords to precisely recommend meaningful keywords which keyword extraction techniques can't deal with. Keywords extraction method can find the relative words/phrases in the text content of items. In the scenarios of digital publication, some of the time, keywords that describe items do not come from the content directly but are some comprehensive words that describe the domain and character of the item, and some other time, if the item is quite different from existed training data, keywords extraction method would be useful in recommending new keywords to the editor. The editors audit and check the recommended keywords and give feedback to the system that whether the keywords are appropriate or not and give the right keywords to update the training model.

Our proposed algorithm selects a hybrid approach of keyword recommendation which considers both the probability based method and traditional extraction based method mentioned above. The reason we select extraction based method as the partner of probability based method is that we hope it can extract some useful words from the item directly where probability based method may not cover.

Hybrid Approach of Keyword Recommendation

Input: item training set TR with annotated tags, new item to be annotated
Output: recommended N keywords of new item.
step1: for each annotated item in TR
segment item into words(for Chinese words especially), delete stop words and xml tags.
step2: for each category j in TR
for each annotated keyword k_i
calculate $p(k_i) = \frac{tf(k_i, TR_j)}{\sum_{i=1}^{n} tf(k_i, TR_j)}$
serialize all $p(k_i)$
step3: for each category x in TR
for each annotated keyword k_i
for each word w_j in item of TR_k
calculate $p(w_j \mid k_i) \propto \frac{tfidf(w_j) \times tf(w_j \mid k_i)}{p(k_i) \times \sqrt{\sum_{j=1}^{m} tf^2(w_j \mid k_i)}}$

serialize all $p(w_j \mid k_i)$

step4: classify new item into category x;

step5: for each annotated keyword k_i in category x

$$p(item \mid k_i) = \prod_{j=1}^{m} p(w_j \mid k_i)$$

$$p(k_i \mid item) \propto p(item \mid k_i) p(k_i)$$

step6: sort $p(k_i \mid item)$ and select the top N*(1-p) as key_probability.

step7: run TextRank keyword extraction algorithm, select top N*p keywords as key_extraction.
p is the proportion of keywords which extraction based method generated in the final result.

step8: combine key_probability with key_ extraction result.

Fig. 1. Algorithm of hybrid keyword recommendation

Step1 to step3 is the training process of probability-based keyword recommendation. Step4 to step6 first classify the new item to category i, and then calculate the probability of every keyword in category i with respect to the new item. Step7 to step8 select part of the keywords from probability-based keyword recommendation and part of the keywords from the extraction based keyword recommendation method.

6 Experiments and Evaluation

6.1 Dataset

In digital publication domain, we have accumulated huge amounts of items collected from books published by the press. These items are xml texts which mainly contain Chinese words together with some English terminology. The annotated keywords are assigned to items by editors manually with each keyword consisting of one or more words. Items have been classified to a constrained category tree which will be used in the classification process. The dataset has 40147 annotated items in xml format with different category and number of keywords. We split the dataset into two parts: the training set and the test set and use the 10-fold cross-validation to test and validate our method. We evaluate our hybrid and probability based method against the traditional keyword extraction method (like TextRank[4]) and tfidf -similarity based method. We did not use the user study valuation method for that we have enough annotated items to test and the annotated items were annotated by expert editors who have enough authority in tagging work, and it also saves lots of time. The statistics of the corpus for training and testing is listed in table 1.

Table 1. statistics of the corpus for training and testing

Category	number of docs	number of keywords	average doc length	average number of keywords per doc
network security	2199	561	220	5.1
AutoCAD	1931	79	118	5.2
Java	1820	845	178	5.2
Electricity	884	162	52	4.1
photoshop	714	71	173	5.1
SCM	249	32	107	3.2
vehicle maintenance	169	9	123	6
graphics	134	489	129	5.3
…	…	…	…	…
Overall 98 categories	40147	21684	182	5

6.2 Evaluation Metrics

This section presents the evaluation metrics in our experiments which include precision, recall, F1. These metrics, when used in combination, have shown to be effective for evaluation of the effect of our method. Precision, recall, and F1 (F-measure) are well-known evaluation metrics in information retrieval literature [12]. T_r denotes the number of keywords returned by the algorithm when new item comes. We use the original set of keywords as the ground truth T_g.

In out experiment, Precision, recall and F1 measures are defined as follows:

$$precision = \frac{T_g \wedge T_r}{T_r}, recall = \frac{T_g \wedge T_r}{T_g}, F1 = \frac{2 \times precision \times recall}{precision + recall}$$

6.3 Results

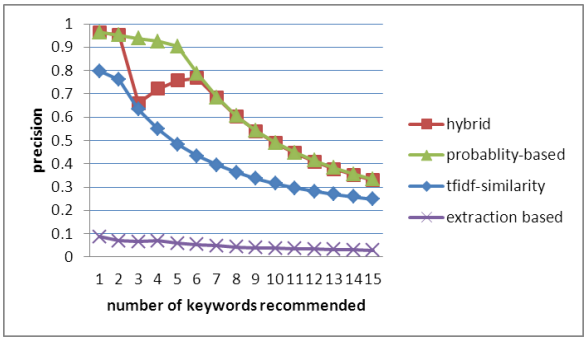

Fig. 2. Precision of four keywords recommendation method

We test the number of keywords recommended to the testing item from 1 to 15 using four different keyword recommendation algorithms which include the hy-

brid and probability based method we proposed. In the digital publication domain, editors often annotate 3 to 5 keywords/phrases per item, so we pay more attention to the result of keywords recommendation that recommend 3 to 5 keywords.

Figure 2, 3, 4 plots the precision, recall and f1 value of the four different keywords recommendation method. We can see that the hybrid and probability based keywords recommendation methods we proposed outperform other methods like the tfidf-similarity based method and traditional keyword extraction method. The probability based method performs better than hybrid method when 3 to 5 keywords are recommended because the hybrid result contains keywords from the result of the traditional extraction method which result in the loss of precision. When one to five keywords are recommended, the probability based method can achieve precision more than 90% which is much higher than the tfidf similarity and extraction based method.

Extraction based method performs worst since previous keyword annotation work of items is done by editors and the keywords annotated mostly are not from the content of the items directly but from a comprehensive understanding of the item. Another reason is that the average length of the items is only 182, and finding appropriate keywords in short item is not very easy for traditional keyword extraction method. But it does not means that we would abandon the extraction method because there are cases that the coming new item is quite different from the training set and the recommended keywords from statistical information may not cover the main idea of the item. Extraction of keywords from the content of item helps editors to have a chance to give personalization keywords to the item.

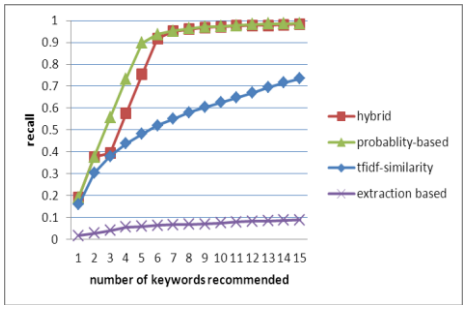

Fig. 3. Recall of four keywords recommendation method

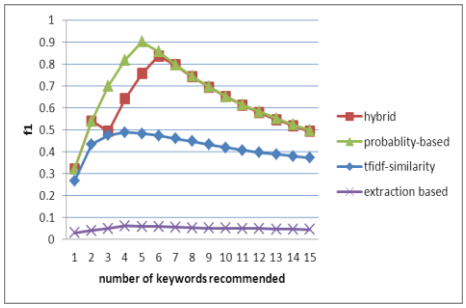

Fig. 4. F1 value of four keywords recommendation method

We achieve a high recall value up to 0.98 when 5 or more keywords are recommended. Recall is rather important for the keyword recommendation process in digital publication domain for that most of the time the recommended keywords are not adopted automatically but needs manual verification and audit.

High recall helps editors to select keywords that are most relevant to the new item in a wider range while low recall limits the scope the editor and if the editor can't find the proper keywords in the recommended keywords list, it would cost the editor lots of time to look through the content of the item and select the keywords manually. From figure 3 we can see that the recall value rises when more keywords are recommended and when 5 or more words are recommended the highest steady recall value are achieved.

We found that when keywords number is five, we achieved the best f1 value(0.9) with pretty high precision and recall because of the large number of training set of items are annotated with five keywords. Less recommended keywords will result in the loss of recall and f1 value but more keywords will result in the loss of precision.

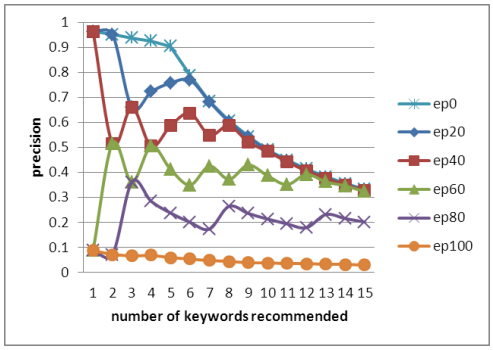

Fig. 5. Precision of Hybrid method with different proportion of extraction based method

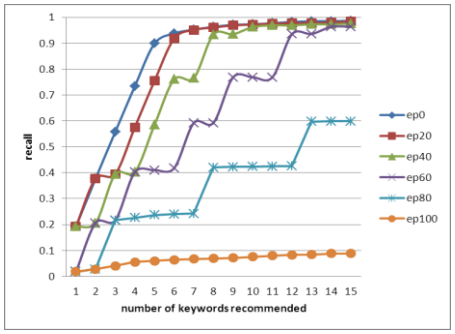

Fig. 6. Recall of Hybrid method with different proportion of extraction based method

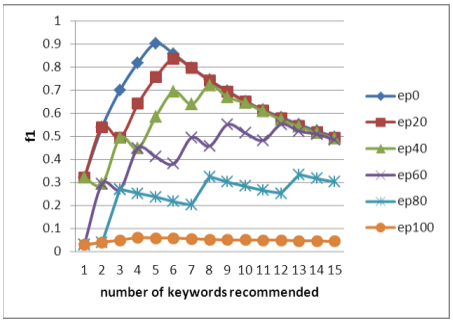

Fig. 7. F1 value of Hybrid method with different proportion of extraction based method

According to the experiment result in figure 5, 6 and 7, when we select 20% percent of the result of extraction based method combining with 80% percent of the result of probability based method, the relatively better precision, recall and

f1 value are achieved. So we select the parameter ep20(0.2) as the p parameter in hybrid algorithm. The series ep0, ep20 to ep100 in figure 5, 6 and 7 means the percentage of extraction based method used in the final keyword recommendation process. ep0 equals with the probability based method and ep100 corresponds to the extraction based method. When editors hope recommend new keywords from the content of the item directly we can use the hybrid approach, otherwise, the probability approach are recommended.

7 Conclusions and Future Work

This paper presents probability based and hybrid keyword recommendation algorithm which get at most more than 90% precision, recall and f1 value on the digital publication dataset which outperforms the traditional extraction based and tfidf similarity based method in keyword recommendation. The algorithm is motivated by the keyword annotation problem in digital publication. When there is a new item that is not annotated, the algorithm automatically recommends relative keywords to the editor.

The probability based method utilizes statistical information of annotated training sets to recommend existed annotated keywords to coming items. The hybrid method combines the traditional extraction based method and the probability based method to take advantage of the two methods. Experiments are done on the dataset of items of books provided by the press and show that probability based and hybrid method outperforms the traditional keyword extraction method and tfidf similarity based method. Future work includes experiment on other annotated datasets, improvement on topic model based algorithm and other extraction based algorithms

Acknowledgments. The paper is supported and completed under the financial aid of the National Science-Technology Support Plan Projects " Research and Development of Key Support Technology and Application Demonstration on Dynamic Digital Publishing "(2012BAH88F00, 2012BAH88F02).

References

1. Zhang, K., Xu, H., Tang, J., Li, J.: Keyword Extraction Using Support Vector Machine. In: J.X. Yu, M. Kitsuregawa, and H.V. Leong (eds.) WAIM 2006. LNCS, vol. 4016, pp. 85–96. Springer, Heidelberg (2006)
2. Litvak, M., Last, M.: Graph-Based Keyword Extraction For Single-Document Summarization. In: MMIES '08 Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization, pp. 17–24. Association for Computational Linguistics, Stroudsburg, PA, USA(2008)
3. Tonella, P., Ricca, F., Pianta, E., Girardi, C.: Using keyword extraction for web site clustering. In: Fifth International Workshop on Web Site Evolution, pp. 41–48. IEEE Press, New

York(2003)
4. Mihalcea, R., Tarau, P.: TextRank: Bringing Order into Texts. In: Proceedings of EMNLP 2004, pp 404–411. Association for Computational Linguistics, Barcelona, Spain(2004)
5. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing and Management Journal, *24(5)*, 513–523(1988)
6. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web. In: Proceedings of the 7th International World Wide Web Conference, pp 161–172, Brisbane, Australia(1998)
7. Wan, X.J., Xiao, J.G.: Single Document Keyphrase Extraction Using Neighborhood Knowledge. In: Proceedings of the 23rd national conference on Artificial intelligence(AAAI'08)-Volume 2, pp 855–860. AAAI Press (2008)
8. Poibeau, T., Saggion, H., Piskorski J.: Multi-source Multilingual Information Extraction and Summarization. In: MMIES '08, pp 17–24. Association for Computational Linguistics, Stroudsburg, PA, USA(2008)
9. Zhang, C.Z., Wang, H.L., Liu, Y., Wu, D., Liao, Y., Wang, B.: Automatic Keyword Extraction from Documents Using Conditional Random Fields. Journal of Computational Information Systems,(2008)
10. Tuarob, S., Pouchard, L.C., Giles, C.L.: Automatic tag recommendation for metadata annotation using probabilistic topic modeling. In: Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries, vol.15, pp. 239-248. ACM(2013)
11. Ni N., Liu K., Li Y.D.: Study of Automatic Keywords Labeling for Scientific Literature. Journal of Computer Science, 39(9), (2012)
12. Manning, C. D., Raghavan, P., Schtze, H.: Introduction to Information Retrieval. Cambridge University Press, NewYork, NY, USA,(2008)
13. Jiang, X., Hu, Y., Li H.: A Ranking Approach to Keyphrase Extraction. In: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, pp 756–757. ACM(2009)