고정키어구 추출을 통한 디지털 문서의 도메인 특정 주석

이람 파티마, 이영구, 이승룡
경희대학교
e-mail : {iram.fatima, sylee}@oslab.khu.ac.kr, yklee@khu.ac.kr

Domain Specific Annotation of Digital Documents through Keyphrase Extraction

Iram Fatima, Young-Koo Lee, Sungyoung Lee,
Dept. of Computer Engineering, Kyung Hee University, Korea

Abstract

In this paper, we propose a methodology to annotate the digital documents through keyphrase extraction using domain specific taxonomy. Limitation of the existing keyphrase extraction algorithms is that output keyphrases may contain irrelevant information along with relevant ones. The quality of the generated keyphrases by the existing approaches does not meet the required level of accuracy. Our proposed approach exploits semantic relationships and hierarchical structure of the classification scheme to filter out irrelevant keyphrases suggested by Keyphrase Extraction Algorithm (KEA++). Our experimental results proved the accuracy of the proposed algorithm through high precision and low recall.

1. Introduction

A challenge in processing digital documents is to manipulate and search relevant information as volume of available information is continuously increasing. So there is a growing need in helping people to better find, filter and manage these resources. Keyphrases express the primary topics and themes of a document precisely [1]. They are useful in text clustering and classification [2], content-based retrieval , automatic text summarization, thesaurus construction, representing search results, and navigation [1-4]. It is widely used for organizing digital data and providing thematic access to them. In keyphrase extraction, the phrases occurring in a document are analyzed to identify apparently significant according to the specific taxonomy and the document is aligned according to its contents that correspond to the elements of taxonomy. Existing approaches for keyphrase generation are: *keyphrase extraction* and *keyphrase assignment* [4,5]. In keyphrase extraction, significant keyphrases in a document are identified through properties such as frequency and length. While in the latter approach, keyphrases are generated by semantically aligning a document with taxonomy. The quality of the generated keyphrases by the existing approaches has not been able to meet the required level of applications [5,6].

The proposed methodology is a novel approach of refinement, comprising two major processes (a) extraction and (b) refinement. Extraction of keyphrases is the prerequisite of refinement process. We adopt KEA++ (Key Phrase Extraction Algorithm) [7-11] for extracting keyphrases. Refinement process refines the result set of keyphrases returned by KEA++ using different levels of taxonomy. We observed in our analysis that the hierarchical levels of taxonomy and their generalization and specialization play vital role in both training and extraction process of KEA++. Refinement rules exploit the semantics of different levels and select keyphrases located on the most relevant levels. Experiments have been performed on dataset of 100 documents collected from the ACM Computing Surveys[1]. Experimental results show increase in precision from 0.14 to 0.38 and decrease in recall from 0.42 to 0.38 at the fourth level of the ACM Computing Classification[2]

The rest of the paper is organized as follows. Section 2 explains the proposed methodology of automatic keyphrase refinement. Results from ACM Computing surveys dataset are given in Section 3. Conclusion together with possible future work discusses in section 4.

2. Proposed Methodology

Our proposed methodology processes the returned results of KEA++ [7-11] by exploiting different hierarchical level of taxonomy. It comprises two main steps: (a) extraction and (b) refinement. We adopted KEA++ for extraction after customized parameter setting according to the structure of taxonomy. ACM computing classification has been used as

1 http://surveys.acm.org/
2 http://www.acm.org/about/class/1998/

taxonomy in the SKOS format using UTF-8 encoding. It is used for the implementation and testing purpose of our algorithm, while our contribution is adoptable for other classification systems.

Refinement process of extracted keyphrases is based on refinement rules, which emphasize on different hierarchical level of taxonomy and associated semantic relations among them. These refinement rule set discard the irrelevant keyphrases and retain the most relevant ones according to available relation within different level of taxonomy. These rules are defines as follows:

Rule I: Adopt Training Level and identify Taxonomy level labels: The training level is the hierarchical level of the taxonomy; we adopt the KEA++ training level during the refinement process. This rule is used to set the level of training in the hierarchy of the taxonomy to extract the refined set of semantic keyphrases. The effective usage of the remaining rules depends on the accurate value of the training level of the taxonomy.

Rule II: Preserving the Training Level Keyphrases: This rule helps in preserving the training level keyphrases. KEA++ results have keyphrases that belong to different levels in the taxonomy. It might have upper level keyphrases and lower level keyphrases which do not contain information as relevant as the training level keyphrases.

Rule III: Stemming the Lower Level General Keyphrases: In the ACM Computing Classification, there is the *general* category of keyphrases on each level of the hierarchy. If a keyphrase is aligned on a lower level than the training level (e.g., C.2.3.0), and associated with the general category in the lower level, then we stem the lower level keyphrase to its training level (e.g., C.2.3) keyphrases

Rule IV: Preserving the Lower Level Keyphrases: If the result set of KEA++ contains no training level keyphrases, then we preserve the lower level keyphrases from the result set of KEA++.

Rule V: Identifying and Preserving the Training Level Equivalent Keyphrase: Different keyphrases aligned to separate categories of the ACM taxonomy can be semantically equivalent, e.g., *Control Structures and Microprogramming* (B.1) is equivalent to *Language Classifications* (D.3.2). If the upper level has equivalent keyphrases of the training level, then preserve the training level keyphrase before discarding the upper level keyphrase

Rule VI: Removing Redundant Keyphrases: After applying above rules, the result might contain redundant keyphrases (i.e., C.2.3, C.2.3, D.4.5). Remove the redundant keyphrases from the set of refined keyphrases (i.e., C.2.3, D.4.5).

Keyphrases returned by the KEA++ either has training level terms or only contain upper and lower level keyphrases. After identify the taxonomy level labels, our algorithm search for training level keyphrases and behave accordingly as shown in Algorithm. 1. Finally, redundancy is removed followed by the refined set of keyphrases from digital documents.

Pseudo code: Refinement Process

Input:
- **Training**
  - (a) Set the parameters of the KEA++ by keeping in view keyphrase length in taxonomy and documents type.
  - (b) Documents along with their semantic keyphrase and taxonomy
- **Dataset for Extraction:**
  - (a) Documents with unknown keyphrases

Output: Set of refined keyphrases

```
TrainLevel    ←  KEA++ TrainLevel
resultSet []  ←  returned keyphrases by KEA++[]
resultSet []  ←  level labels (Resultset [])
for resultSet[] <> empty do
 if (resultSet(training level)) then
   if (keyphrase level = lower level keyphrases) then
     processSet[] = preserving lower level keyphrases
   else
     set processSet ← identifying and preserving training level
                      equivalent
     processSet[] ← remove redundant keyphrases
     refineSet[] ← processSet[]
 else
    if (keyphrase level = training level) then
          refineSet[] ← processSet[]
    else
       if (keyphrase level = upper level) then
         processSet[] ← identifying and preserving training
                        level equivalent keyphrases
       else
         processSet[] ← stemming lower level general
         keyphrases
 processSet[] ← remove redundant keyphrases
 refineSet[]  ← processSet[]
return refineSet[]
```

Algorithm 1. Keyphrase Refinement Algorithm

3. Experimental Results and Evaluation

In this section, the results of manual annotation, KEA++, and the proposed refinement algorithm are compared. The precision of the refinement algorithm is tested on various hierarchical levels as provided in the manual annotations of the datasets. Datasets for the experiments are composed of documents from the ACM Computing Surveys[3] . They are mostly aligned on the fourth level of the ACM Computing Classification. Experiment is performed on 100 documents, 70 documents were used for training and 30 were used for testing. We implement refinement algorithm using Jena API written in Java with system Pentium(R) Dual-Core 2 GHz computer, 2 GB Memory, and windows XP professional operating system.

The number of KEA++ returned keyphrases lies between 1 to 20, manual annotation vary from 1 to 8 while refinement algorithm range from 0 to 12 as shown in Figure 1. The statistical analysis presents that the number of keyphrases returned per average number of documents in the refinement algorithm is closer to the manual annotation.

[3] http://uclab.khu.ac.kr/ext/Dataset_ACM_Computing_Surveys.rar

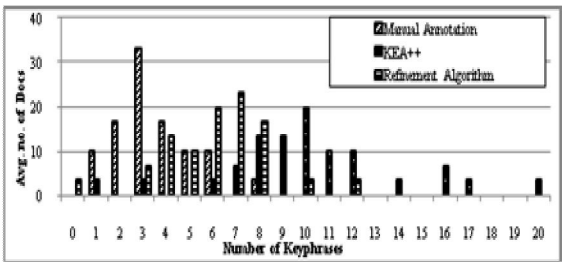

Figure 1. Keyphrase returned per avg. no of documents

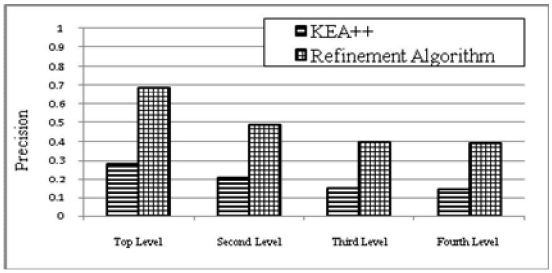

Figure 2. Precision against Total Keyphrases Returned

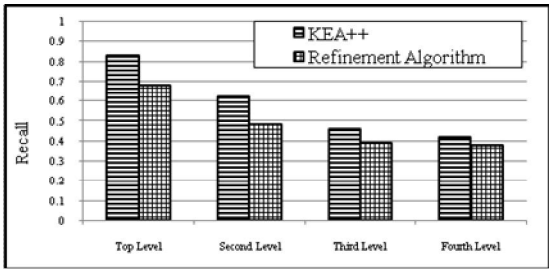

Figure 3. Recall against Total Keyphrases Returned

Fig. 2 shows that refinement algorithm gives more precise results as compare to KEA++ while the recall is low in case of refinement process as depicted in Fig. 3.

In [7] the precision and recall of KEA++ are 0.283 and 0.261 respectively while the average number of manual annotation is 5.4 per document in the dataset of 200 documents. While the precision and recall of KEA++ for our dataset of 100 documents (with 2.35 average number of manual annotation per document) is 0.38. The precision has been improved from 0.14 to 0.38 i.e. 191.9% on the same dataset while recall is decreased from 0.42 to 0.38.

4. Conclusion

Accuracy of extracted keyphrases for the annotation of digital documents is a key challenge. Our proposed approach takes into account semantic relations between terms that appear in the document along with different levels of ACM taxonomy. The refinement algorithm applies the set of rules on the extracted keyphrases returned by KEA++ and improved the accuracy in terms of high precision and low recall. In our future work, we have planned to validate our proposed algorithm for health-care domain, where the accuracy is the major issue during the extraction of keyphrases from the unstructured e-health data.

Acknowledgement

This research was supported by the MKE (Ministry of knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency)" (NIPA-2009-(C1090-0902-0002)).
Also, it was supported by the IT R&D program of MKE/KEIT, [10032105, Development of Realistic Multiverse Game Engine Technology].

References

1. Liu, Z. Li, P. Zheng, Y. Sun, M.: Clustering to Find Exemplar Terms for Keyphrase Extraction. In: Empirical Methods on Natural Language Processing, pp. 257--266, ACL Proceedings, Singapore (2009)
2. O. Roberto, D. Pinto, M. Tovar: BUAP: An Unsupervised Approach to Automatic Keyphrase Extraction. In: 23rd 5th International Workshop on Semantic Evaluation ACL, pp. 174--177, Sweden (2010)
3. Jones, S. Paynter, G.: Automatic extraction of document keyprases for use in digital libraries: evaluation and applications. J of the American Society for Information Science and Technology, vol. 53(8), pp. 653--677, (2002)
4. Turney, P.D.: Coherent keyphrase extraction via web mining. J. of the American Society for Information Science and Technology, pp. 434--439 (2003)
5. Kim, S. N. Kan, M.Y.: Re-examining automatic keyphrase extraction approaches in scientific articles, MWE, pp. 9--16, (2009)
6. I. Fatima, S. Khan, K. Latif, Refinement Methodology for Automatic Document Alignment Using Taxonomy in Digital Libraries, ICSC, pp. 281--286, USA (2009)
7. Medelyan, O. Witten, H. I.: Thesaurus Based Automatic Keyphrase Indexing. Joint Conference on Digital Libraries, pp. 296--297, USA (2006)
8. Medelyan, O. Witten, H. I.: Semantically enhanced Automatic Keyphrase Indexing, WiML, (2006)
9. Medelyan, O., Witten I. H.: Thesaurus-based index term extraction for agricultural documents. In: 6th Agricultural Ontology Service workshop at EFITA, Portugal. (2005)
10. Witten, I. H. Paynter, G. W. Frank, E. Gutwin, C. Nevill-Manning, C.G.: KEA: Practical automatic keyphrase extraction, pp. 254--256 (1999)
11. Witten, H. I. Paynter, G. W. Frank, E. Gutwin: Kea: Practical automatic keyphrase extraction, Design and Usability of Digital Libraries, pp. 129--152, (2005)