KPE: An Automatic Keyphrase Extraction Algorithm

Conference Paper · September 2011

CITATIONS
10

READS
506

3 authors:

Aliaa Youssif
Arab Academy for Science, Technology & Maritime Transport (AASTMT)
95 PUBLICATIONS **875** CITATIONS
SEE PROFILE

Atef Z. Ghalwash
Helwan University
59 PUBLICATIONS **689** CITATIONS
SEE PROFILE

Eslam A. Amer
Misr International University
28 PUBLICATIONS **70** CITATIONS
SEE PROFILE

KPE: An Automatic Keyphrase Extraction Algorithm

Aliaa A.A. Youssif
Computer Science Department
Helwan University
Helwan– Egypt
aliaay@yahoo.com

Atef Z.Ghalwash
Computer Science Department
Helwan University
Helwan– Egypt
atef_ghalwash@yahoo.com

Islam A.Amer
Computer Science Department
MSA University
6-October – Egypt
is_fahim@yahoo.com

Abstract—In this paper, an automatic Keyphrase extraction algorithm (KPE algorithm) is presented for single English document without using controlled vocabularies (corpus). It relies on n-gram filtration technique that filters complicated n-grams. The proposed algorithm proposesa ranking method to weight extracted keyphrases. The ranking method relies onthe position, frequency, as well as χ2-measure for testing the bias of co-occurrence distribution between frequent terms. Experimental results with publically available scientific text dataset show a comparable performance to both supervised algorithms (e.g. KEA) and unsupervised algorithms (e.g. N-gram filtration technique). Meanwhile, evaluations show that the proposed algorithm outperforms existing KEA and N-gram filtration technique methods producing keyphrases with higher precision and recall.

Keywords: Automatickeyphrase extraction, n-gram filtration technique, KEA.

I. INTRODUCTION

Keyphrases (sometimes referred to as keywords or key terms) are a set of significant terms in a text document that capture the main topics covered in a document. Keyphrase extraction is a fundamental step for various tasks of Natural Language processing (NLP)e.g. document classification, document clustering, and text summarization [1].They can be used in information retrieval systems as descriptions of documents returned by a query, as the basis for search indexes, as a way of browsing a collection, and text mining and so on. It is also a core task for internet based advertising systems such as Google's AdSense and Yahoo! Contextual Match. They display advertisements based on keyphrases found in web pages. Improvements on the quality of keyphrases extracted from the web page directly leads to approximately 10% higher click-through advertisement [2].

Keyphrases are usually assigned manually,professional indexers assign keyphrases to documents from a "controlled vocabularies" relevant to document domain. As digital libraries that contains huge number of scientific documents increases in size every day, and the majority of such documents don't contains author assigned keyphrases; Automatic keyphrase extraction techniques are potentially of great benefit as scientific documents contains a broad range of human activities and information, also to manually assign keyphrases to each document is atedious process that requires knowledge of the subject matter which is not feasible at all.

II. RELATED WORKS

There are many approaches for extracting keyphrases. The simplest one is to use term frequency criteria (e.g. TFxIDF [3]); however it generally yields poor results [4]. State-of-the-art approaches in this area uses supervised leaning methods, where a system is trained using controlled vocabularies (a dictionary, thesaurus, or a list of terms) to recognize keywords in a text, based on lexical and syntactic features.

KEA [5] [6] one of the most prominent approach of this class. In this approach the candidate keyphrases are weighted based on three features: TFxIDF, distance (number of words that precede the first occurrence of the keyphrase divided by number of words in documents, and keyphrase frequency (number of times a candidate keyphrase occurs as keyphrase in training documents). The classifier is trained using naïve Bayes learning algorithm. Thus KEA calculates only some simple statistical properties of candidate keyphrases and unifies each candidate keyphrase independently from otherterms in document.

In contrast to this approach, the proposed algorithm is completely unsupervised which depends solely on the structure of the document and statistical relations between terms in document. Another difference is that KEA depends on training set that may provide poor results if such training set doesn't contains vocabularies that are highly related to the document while our algorithm doesn't require training.

978-1-4244-9646-4 /11/$26.00 ©2011 IEEE

Recently, many new methods have been proposed which extend classical approaches with new features computed over Wikipedia corpus. For example [7] introduced node degree feature that use Wikipedia links to calculate the

distance between candidate terms in document (i.e. number of links of Wikipedia articles between two candidate keyphrases). So candidate keyphrases with high node degree are those that have many related keyphrases in document.

Wikify! [8] Introduced keyphraseness feature of candidate keyphrase which is the probability that the candidate keyphrase is selected as keyphrase in Wikipedia article which is defined as the number of Wikipedia articles which keyphrase appears in and marked up as a link divided by total number of Wikipedia articles where the keyphrase appears.

N-gram filtration Technique [9] which is unsupervised based technique. In this technique sophisticated n-grams (potential candidate keyphrase), are extracted from the words of input document along with their ranking weight in document. This technique go through several steps in order to calculate weights of collected keyphrases (n-grams) (i) preparing dictionary of distinct n-grams using LZ78 data compression algorithm [10],[11] which is required to arrange and capture higher length phrases with lower frequencies, (ii) Refinement of dictionary content; which remove frequent English verbs, (iii) Collect sophisticated n-grams via simple Pattern Filtration algorithm from pre-processed document using separate lists of distinct n-grams (obtained from dictionary), and finally (iv) Term weighting scheme is proposed to calculate the weight of collected n-grams. This technique introduced the use of sentence position (where a given keyphrase occurs first) in document and the position of the keyphrase in the sentence. Evaluations of experimental results showed that N-gram filtration Technique show considerable effectiveness compared to other methods (e.g. KEA) [9].

Although the position of sentence where a given keyphrase occur first has a major impact on the keyphrase'srank; it may lead to pseudo results; as some keyphrases which are not related to the core of the document but appear on an early sentence will got a high rank value.

In KPE, some steps of n-gram filtration technique, which are common among automatic keyphrase extraction algorithms, were used with some modifications to prepare input document for further processing. KPE ranking (weighting) method used many statistical relations between terms in input document as well as relying on the role of χ^2-measure for testing the bias of co-occurrence distribution between frequent terms that is effectively shows remarkable and convenient results compared to KEA and n-gram filtration technique algorithms.

III. PROPOSED ALGORITHM

Scientific domain document are literally more organized than other document, and it is found that the position of sentence where given keyphrase occurs first has an important role, as in such type of documents important terms comes earlier. In a similar manner, position of the phrase in the sentence is a major factor in determining the term weight [9]. Grammatical facts [12], [13] support that noun phrases comes earlier in a sentence. In [6], [14] authors indicate the importance of Noun phrase in keyphrase determination. Likewise, some objects that may be useful as keyphrase comes at almost very end of a sentence. So by combining the frequencies of a keyphrase in the document with the statistical relation about the locations of keyphrase in every sentence in a document could enhance ranking of keyphrase in a given document.

Entire working flow Process of KPE algorithm was outlined in Figure-1. The proposed algorithm is divided into eight stagesnamely; input preprocessing, preparing dictionary distinct items, refine dictionary entries, Filtering n-grams,normalizing dictionary entries, Computation of Chi-square test for keyphrases, and Ranking keyphrases. Description of each stage will be discussed in the following sections.

A. Input Preprocessing.

The objective of this step is to separate sentences in separate lines and organize phrase boundary. This was done by first convert all entire input text to lower case, then splitting the input document into sentences based on splitting criteria of full stop ".". Stop words, trivial words, numbers, apostrophes are removed and replaced by "*". A list of 179 stop words was used, which are available with KEA source code [15]. Some features of porter stemming algorithm [16] is applied (i.e. suffix "ing" is not stemmed).

B. Preparing dictionary distinct items using LZ78 algorithm

The objective of this step is to remove any redundancy of keyphrases as to arrange and capture higher length phrases with lower frequencies. Its final outcome is a dictionary that contains only distinct items. This is done by applying LZ78 data compression technique [10], [11] with some modification in that words are used instead of characters, and space is used as separator character between words.For fully understanding of how this stage works, consider the following example:

Assume that the following two sentences were taken from the processed document:

*Sentence #1:"*w1 w2 w3*w3 w4* w1 w2 w3* w4 w5*"*

Sentence #2:" w4 w5 * w1 w2 w4 * w3 w6 *"*

Where w1, w2, w3, w4, w5, and w6 represents different words, and "*" is the phrase boundary in sentence. Preparations of distinct dictionary for the given sentences were demonstrated in Table 1.

Input Sentences:
" w1 w2 w3 * w3 w4 * w1 w2 w3 * w4 w5 *"*
" w4 w5 * w1 w2 w4 * w3 w6 *"*

Input sentence	Pattern	Action	Dictionary
w1 w2 w3 w1 w2 w3	w1 w2 w2 w3	Add to dictionary Add to dictionary	w1 w2 w2 w3
w3 w4	w3 w4	Add to dictionary	w3 w4

w1 w2 w3 w1 w2 w3	w1 w2 w1 w2 w3	Word exist Add to dictionary	- w1 w2 w3
w4 w5	w4 w5	Add to dictionary	w4 w5
w4 w5	w4 w5	Word exist	-
w1 w2 w4 w1 w2 w4	w1 w2 w1 w2 w4	Word exist Add to dictionary	- w1 w2 w4
w3 w6	w3 w6	Add to dictionary	w3 w6
-	-	quit	

Table 1: Distinct Dictionary Preparation of words using LZ78 data compression technique

C. Refine dictionary entries

The objectives of this step is to refine the contents of the dictionary of phrases by eliminating frequent English verbs as these words are less important to be fit as a keyphrases for document. For such purpose a list of frequent verbs obtained from [17], [18] is used as a guide.

D. Filtering n-grams

According to experimental observations, it was remarked that n-grams that have minimal occurrence (occurred once) were likely unimportant ones. In this stage, n-grams that have lowest frequencies were deleted from dictionary.

E. Compute term occurrence weight

In this stage keyphrases are assigned two different weight values, the first value, *frequency weight* which is based on the frequency of keyphrases in the distinct dictionary (i.e., occurrence times in processed document). The second value, *influence weight*; which is based on times of frequency count of keyphrases inside the sentences based on grammatical facts [9] which support idea that noun phrases comes earlier in the sentence or keyphrases may come at the end of sentence. The latter weight is calculated as: If N_i is the number of words in sentence $S_i$and P_0 isthe index of starting word of phrase P, then theoccurrence of phrase P is taken into account if P_0satisfy the following condition:

$$0 \quad \leq P_0 < \frac{N_i}{2} \quad or \quad P_0 > \left(\frac{(3\ x\ N_i}{4}\right)$$

F. Normalizing dictionary entries

At this stage, Normalization step is carried out to concatenate related keyphrases that are part of each other into one keyphrase, a number of heuristics were applied to detect that two separate keyphrases are related to each other. The main function of such stage is to reduce dictionary entries.

G. Computation of Chi-square test for keyphrases

The objective of this stage is to determine if the co-occurrence of terms in keyphrases are highly dependent on each other or just co-occurred by chance, therefore a term with co-occurrence bias may have an important meaning in document. In order to evaluate the significance of bias, χ^2 test is used for evaluating biases between expected frequencies and observed frequencies. For each term, co-occurrence frequency with frequent terms is regarded to value; if the occurrence of frequent terms G is independent from term w, a null hypothesis occur that means that terms co-occurred by chance, which is expected to reject such co-occurrence. The statistical value for χ^2 is defined as in [19]:

$$\chi 2(w) \ = \sum_{g \in G} \frac{(\ freq(w,g) - n_w\, p_g)^2}{n_w\, p_g}$$

Where $p_g$is sum of terms in sentences where g appears divided by total number of terms in document, n_w denote total number of terms in sentences where w appears, $n_w p_g$ denotes co-occurrence expected frequency, and $freq(w,g)$ denotes frequency of term w and term g, and ($freq(w,g) - n_w\, p_g$) is the difference between observed and expected frequency [19]. If $\chi^2(w) > \chi^2_\alpha$, null hypothesis is rejected at significance level α.

H. Ranking keyphrases

The objective of this stage is to assign a value to keyphrases based on importance of it in the document.KPE algorithmmodified the ranking function used in n-gram filtration technique [9]. Usage of position was modified as well as boosting the term frequency and terminfluence frequency (refer to E). Ranking of keyphrase i was calculated as:

$$Rank(i) = \sum P\ x\ \frac{occurrence(i)}{(L)} + log_2\left(\frac{w + chi(i) + 1}{w - p + 1}\right) \quad (1)$$

Where P is the position of keyphrase which is calculated as $(L - L_s)$ where L is total lines in document, L_s is the first sentence where keyphrase i occurs, this value is multiplied with the occurrence times of keyphrase i in document divided by L as some terms that may appear in first sentences will get large value dependent on their importance in document, p denotes influence weight. The formula $log_2\left(\frac{w+chi(i)+1}{w-p+1}\right)$ is to encourage weight of phrases based on its frequency and its influence weights and its χ^2 value, which is denoted in equation (1) as *chi (i)*.

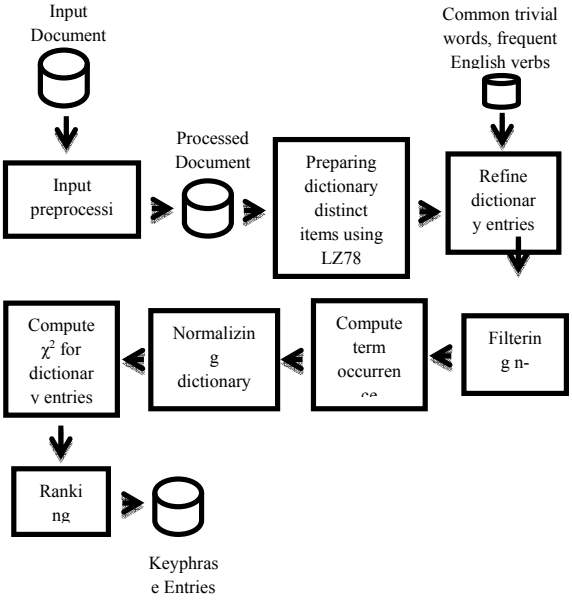

Fig. 1: Workflow of KPE algorithm

IV. EVALUATION

A. Dataset

The dataset used is total of 215 full length text documents from ACM digital library related to computer science subjects publically available for download from [20]. All documents are converted to text form and then author's specific keyphrases were removed.

B. Evaluation measures

KPE algorithm is compared with two different algorithms KEA [5] [6] and n-gram filtration technique [9]. Keyphrase extraction algorithms are evaluated by their retrieval performance, namely precision and recall.

All algorithms were evaluated according to *precision*, *recall*, and resultant *F-measure* for top twenty entries resulted from each algorithm.

Precision was calculated as a fraction of terms automatically extracted by a method that were also extracted by humans [1]:

$$precision = \frac{|\{manually\ selected\} \cap \{automatically\ selected\}|}{\{automatically\ selected\}}$$

Where {manually selected} denoting the set of all termsidentified for a document by humans, and {machine-selected} denoting the set of all terms extracted for the same document by an automatic method.

Recall was calculated as the fraction of the manually extracted key terms that were also extracted by an automatic method [1]:

$$recall = \frac{|\{manually\ selected\} \cap \{automatically\ selected\}|}{\{manually\ selected\}}$$

The weighted harmonic mean of precision and recall, F-measure was calculated traditionally as [1]:

$$F - measure = \frac{2\,.precision.recall}{precision + recall}$$

C. Overall Effectiveness

A sample output is given in Table 2 which contains top ten results returned by KPE, n-gram filtration technique, and KEA algorithm. The table is organized as follows: document title with its URL is presented in the top table, then a list of keywords as appeared in document, top-ten results output of each algorithm.

Overall performance shown in Table 3 indicates that KPE outperforms much better than other domain specific algorithm KEA and domain independent algorithm N-gram filtration technique. On average KPE has the highest precision value, the highest F-measure value which indicates it presents more accurate results compared to its other algorithms.

TABLE 2: Top ten extracted keyphrases by all algorithms

Algorithm/ criteria	Average Precision	Average Recall	Average F-measure
KEA	0.49	0.64	0.56
N-gram Filtration	0.48	0.68	0.56
KPE	0.66	0.70	0.68

V. CONCLUSION

In this paper an effectivealgorithm (KPE) for automatic keyphrase extraction was presented for English documents. Experimental results showed that KPE algorithm outperforms supervised (KEA) and unsupervised (n-gram) filtration techniques since it produces keyphrases from document with higher precision, recall, and F-measures.

Contrary to all supervised algorithms KPE requires neither training nor controlled vocabularies during keyphrase extraction process. Moreover, the proposed algorithm relies on position, frequency, as well as χ2-measure as an indication of the relatedness strength between terms in a given keyphrase. KPE is also an enhancement of n-gram filtration technique since it filters pseudo keyphrases. Like other unsupervised algorithms, KPE is domain independentalgorithm thatcandeal with documents in different domains withoutthe need to controlled vocabularies.

Document name: A Case Study on How to Manage the Theft of Information **Document URL**: http://portal.acm.org/citation.cfm?id=1107622.1107653		
Author Assigned keyphrases: Information Security, Security Management, Information Security Management		
KPE algorithm 10 - matches found, presented by bold font	**n-gram filtration technique** 6- matches found presented by bold font	**KEA algorithm** 7- matches found presented by bold font
security breach **information security** business making **information security** **information** system, **security** incident confidential **information** theft **information management** personal **information** choicepoint**security** **information** broker	**information** abstract **management** plays handle **security** **security** breach inability determine lack planning ongoing risk risk analysis theft **information management** **information** system	theft of **information** **security** breach **management** issue **information** system cyber crime encryption personal **information** risk analysis confidential **information** **information security**

REFERENCES

[1] Maria Grineva, Maxim Grinev and Dmitry Lizorkin. "Extracting key terms from noisy and multitheme documents". Proceedings of the 18th international conference on World wide web, pp: 661-670, 2009.

[2] W. tau Yih, J. Goodman, and V. R. Carvalho. "Finding advertising keywords on web pages". In WWW '06: Proceedings of the 15th international conference on World Wide Web, pp. 213–222, New York, NY, USA, 2006. ACM

[3] G. Salton and C. Buckley. "Term-weighting approaches in automatic text retrieval". Inf. Process. Manage. 24(5):513–523, 1988.

[4] ShouningQu, Sujuan Wang, Yan Zou, "Improvement of Text Feature Selection Method Based on TFIDF". pp.79-81, Proceedings ofInternational Seminar on Future Information Technology and Management Engineering, 2008.

[5] E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-manning. "Domain-specific keyphrase extraction". pp. 668–673. Morgan Kaufmann Publishers, 1999.

[6] Witten, et al. "KEA: practical automatic keyphrase extraction." Proceedings of the fourth ACM conference on Digital libraries, pp. 254 – 255, 1999.

[7] O. Medelyan, I. H. Witten, and D. Milne. "Topic indexing with Wikipedia". pp. 19-24 In Wikipedia and AI workshop at the AAAI-08 Conference (WikiAI08), Chicago, US, 2008.

[8] R. Mihalcea and A. Csomai. "Wikify!: linking documents to encyclopedic knowledge". In CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, pages 233–242, New York, NY, USA, 2007. ACM.

[9] Niraj Kumar KannanSrinathan." Automatic keyphrase extraction from scientific documents using N-gram filtration technique". pp. 199-208, Proceeding of the eighth ACM symposium on Document engineering, 2008.

[10] Khalid Sayood, Introduction to Data Compression, ELSEVIER, 2nd Edition 2000.

[11] Ida m. Pu. Fundamental data Compression, ELSEVIER, 1st edition 2006.

[12] Moro, A. The raising of predicates. Predicative Noun Phrases and the Theory of Clause Structure, Cambridge Studies in Linguistics, Cambridge University Press, Cambridge, England. 1997.

[13] Hale, K.; Keyser, J. "Prolegomena to a theory of argument structure", Linguistic Inquiry Monograph, 39, MIT Press, Cambridge, Massachusetts. 2002.

[14] ShouningQu, Sujuan Wang, Yan Zou, "Improvement of Text Feature Selection Method Based on TFIDF". pp.79-81, Proceedings ofInternational Seminar on Future Information Technology and Management Engineering, 2008.

[15] Web link for KEA5.0 source code: http://www.nzdl.org./Kea/download.html

[16] Porter. "An algorithm for suffix stripping", Morgan Kaufmann Multimedia Information And Systems Series Program, Vol. 14 No.3, pp. 130-137, 1980.

[17] English Vocabulary: Regular Verbs List (EnglishClub.com)

[18] "Irregular verbs:English – Wiktionary", http://en.wiktionary.org/wiki/Appendix:English_irregular_verbs

[19] Yutaka Matsuo and Mitsuru Ishizuka." Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information". International Journal on Artificial Intelligence Tools, vol. 13, pp. 157-170, 2004.

[20] http://aye.comp.nus.edu.sg/downloads/keyphraseCorpus/