Volume 6, Issue 5, May 2016 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Novel Extension for Automatic Keyword Extraction

Ambar Dutta[*]

Department of Computer Science and Engineering, Birla Institute of Technology,
Mesra, Jharkhand, India

Abstract*— Keywords in a document represent subset of words or phrases from the document for describing its meaning. Manual assignment of quality keywords is error-prone, time-consuming and expensive. In the literature, there exist a number of algorithms and systems to help people by automatically extraction of keywords. In this paper, we have proposed an improvement of an existing automatic keywords extraction algorithm (RAKE) and compared it with the original algorithms with the help of different textual data.

Keywords*— Keyword Extraction, Stemming, stop word generation.

I. INTRODUCTION

In the modern society, everyday a huge amount of data is created in form of electronic articles, webpages, emails and web search results. Keywords represent a set of words or phrases from a textual document which enables us to describe the meaning of the text. Various text mining applications are benefitted from it. Keyword search is a powerful tool which helps us to scan a large document collection efficiently without having any knowledge about the syntax and understanding complex semantics of a structured query language. Manual assignment of high quality keywords is often time-consuming, error-prone and expensive. As a result, researchers have proposed a number of algorithms and systems for automatic keywords extraction. Automatic keyword extraction (AKE) enables us to identify a small set of words, key phrases, keywords, or key segments from a textual document. Therefore, keywords extraction from documents is now considered as one of the core technologies of all automatic processing for documents. AKE should be performed systematically with minimal or no human intervention, depending on the model. In this paper, an excellent algorithm, Rapid Automatic Keyword Extraction (RAKE), is studied in detail and an extension of the RAKE is suggested.

The remainder of the paper is organized as follows. A short discussion of some of the existing automatic keyword extraction techniques is given in Section II. Section III deals with the description of a popular keyword extraction scheme, Rapid Automatic Keyword Extraction (RAKE). Proposed improvement and related discussions are presented in Section IV. Finally, the concluding remarks are given in Section V.

II. AUTOMATIC KEYWORD EXTRACTION – A SHORT REVIEW

Researchers have already proposed a number of keyword extraction algorithms which can be classified into three classes [12]: simple statistics, linguistics and machine learning-based. Simple statistics based approaches are simple to understand, have limited prerequisites and tend to focus on non-linguistic features of the text. The keywords in the document can be identified using these statistical information of the words. Cohen [2] indexed the document automatically using N-Gram statistical information. Other statistical methods used include word frequency, term frequency [10], word co-occurrences [8] etc. The purely statistical methods are easy to use and they generally generate good results.

Linguistics approaches use the linguistic features of the words, sentences and document. These techniques pay more attention to linguistic features. Hulth [5, 6] examined various techniques of incorporating linguistics into keyword extraction. It is seen from the experimental results that performance of the automatic keyword extraction is significantly improved by the use of linguistic features.

The machine learning based algorithms work as follows. First a set of training documents is provided to the system, each of which has a range of human-chosen keywords as well. Then the gained knowledge is applied to extract keywords from documents. These methods used naive Bayes formula, support vector machine [14] for domain-based extraction of technical key phrases. Suzuki et al. [12] extracted keywords from radio news using spoken language processing techniques.

Early approaches for automatic keyword extraction were based on the evaluation of corpus-oriented statistics of individual words. Salton et al. [11] selected an index vocabulary by statistically discriminating words across a corpus and obtained good result. Andrade et al [1] presented an algorithm on the basis of the comparison of word frequency distributions within a text. Hulth [6] compared the effectiveness of three term-selection approaches: noun-phrase chunks, n-grams, and POS tags. Mihalcea and Tarau [10] described a system that applies a series of syntactic filters to identify POS tags that are used to select words to evaluate as keywords. A graph-based ranking algorithm is applied to rank words based on their associations in the graph, and then top ranking words are selected as keywords. Matsuo and Ishizuka [9] applied a chi-square measure to calculate how selectively words and phrases co-occur within the same sentences as a particular subset of frequent terms in the document text. The chi-square measure is applied to determine the bias of word co-occurrences in the document text which is then used to rank words and phrases as keywords of the document. It is revealed from the related works that the automatic keyword extraction is faster and less expensive than human intervention.

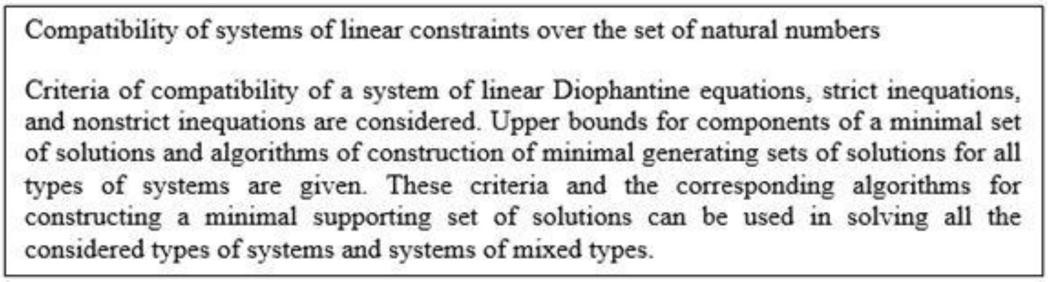
III. RAPID AUTOMATIC KEYWORD EXTRACTION (RAKE)

RAKE [11] is based on the observation that keywords frequently contain multiple words but rarely contain standard punctuation or stop words. Reviewing the manually assigned keywords, there is only one keyword that contains a stop word. Stop words are typically dropped from indexes within IR systems and not included in various text analyses as they are considered to be uninformative or meaningless. This reasoning is based on the expectation that such words are too frequently and broadly used to aid users in their analyses or search tasks. Words that do carry meaning within a document are described as content bearing and are often referred to as content words.

The list of stop words which we have used are listed below:-

Stop word list:- "a", "about", "above", "above", "across", "after", "afterwards", "again", "against", "all", "almost", "alone", "along", "already", "also", "although", "always", "am", "among", "amongst", "amongst", "amount", "an", "and", "another", "any", "anyhow", "anyone", "anything", "anyway", "anywhere", "are", "around", "as", "at", "back", "be", "became", "because", "become", "becomes", "becoming", "been", "before", "beforehand", "behind", "being", "below", "beside", "besides", "between", "beyond", "bill", "both", "bottom", "but", "by", "bye", "call", "can", "co", "con", "could", "couldn’t", "cry", "de", "describe", "detail", "do", "done", "down", "due", "during", "each", "e.g.", "eight", "either", "eleven", "else", "elsewhere", "empty", "enough", "etc", "even", "ever", "every", "everyone", "everything", "everywhere", "except", "few", "fifteen", "fifty", "fill", "find", "fire", "first", "five", "for", "former", "formerly", "forty", "found", "four", "from", "front", "full", "further", "get", "give", "go", "had", "has", "hasn’t", "have", "he", "hence", "her", "here", "hereafter", "hereby", "herein", "hereupon", "hers", "herself", "him", "himself", "his", "how", "however", "hundred", "i", "i.e.", "if", "in", "indeed", "interest", "into", "is", "it", "its", "itself", "keep", "last", "latter", "latterly", "least", "less", "ltd", "madam", "made", "mam", "many", "may", "me", "meanwhile", "might", "mill", "mine", "more", "moreover", "most", "mostly", "move", "much", "must", "my", "myself", "name", "namely", "neither", "never", "nevertheless", "next", "nine", "no", "nobody", "none", "no one", "nor", "nothing", "now", "nowhere", "of", "off", "often", "on", "once", "one", "only", "onto", "or", "other", "others", "otherwise", "our", "ours", "ourselves", "out", "over", "own", "part", "per", "perhaps", "please", "put", "rather", "re", "same", "see", "seem", "seemed", "seeming", "seems", "serious", "several", "she", "should", "show", "side", "since", "sincere", "sir", "six", "sixty", "so", "some", "somehow", "someone", "something", "sometime", "sometimes", "somewhere", "still", "such", "system", "take", "ten", "than", "that", "the", "their", "them", "themselves", "then", "thence", "there", "thereafter", "thereby", "therefore", "therein", "thereupon", "these", "they", "thick", "thin", "third", "this", "those", "though", "three", "through", "throughout", "thru", "thus", "to", "together", "too", "top", "toward", "towards", "twelve", "twenty", "two", "un", "under", "until", "up", "upon", "us", "very", "via", "was", "way", "we", "were", "what", "whatever", "when", "whence", "whenever", "where", "where after", "whereas", "whereby", "wherein", "whereupon", "wherever", "whether", "which", "while", "whither", "who", "whoever", "whole", "whom", "whose", "why", "will", "with", "within", "without", "would", "yet", "you", "your", "yours", "yourself", "yourselves", & "the".

These stop words are used to reduce the huge text to smaller keywords; length of the keyword is length of two stop words. As soon as a stop word match is found it splits from there and make that keyword. The input parameters for RAKE consist of a list of stop words, a set of phrase delimiters, and a set of word delimiters. RAKE used stop words and phrase delimiters to split the document text into candidate keywords, which are sequences of content words as they occur in the text. Co-occurrences of words within these candidate keywords are meaningful and allow us to identify word co-occurrence without the application of an arbitrarily sized sliding window. RAKE begins keyword extraction on a document by parsing its text into a set of candidate keywords. First, the document text is divided into an array of words by the specified word delimiters. This array is further divided into sequences of contiguous words at phrase delimiters and stop word positions. Words within a sequence are assigned the same position in the text and together are considered a candidate keyword. After every candidate keyword is identified and the graph of word co-occurrences is complete, a score is calculated for each candidate keyword and defined as the sum of its member word scores. For example, let us consider the following text:-

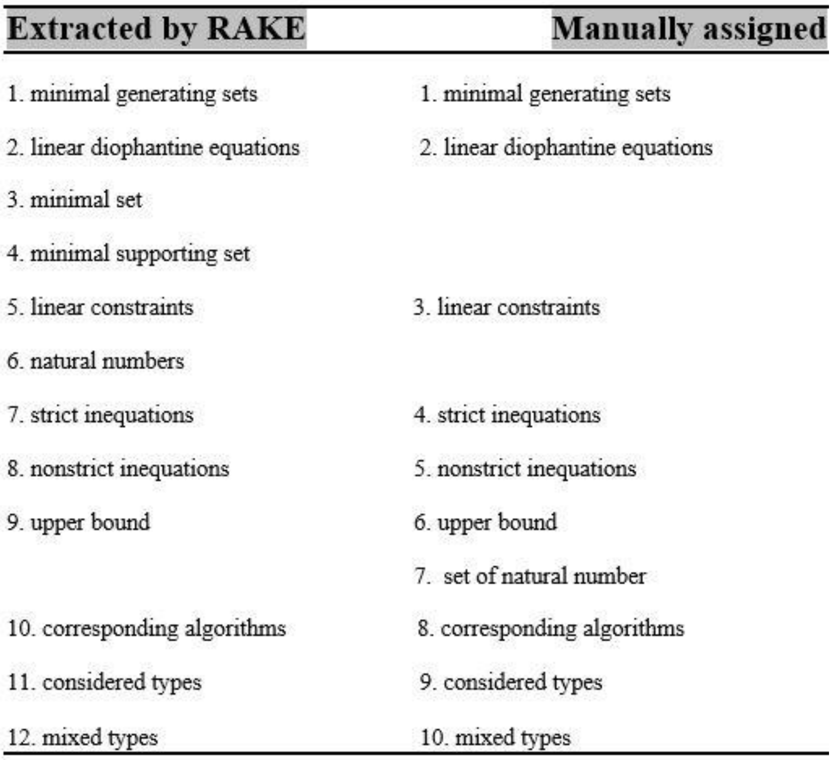


Compatibility of systems of linear constraints over the set of natural numbers

Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered. Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types of systems and systems of mixed types.

The list of keywords extracted by RAKE is given below in TABLE I:

Table I List Of Keywords By Rake And By Manually Intervention



Extracted by RAKE	Manually assigned
1. minimal generating sets	1. minimal generating sets
2. linear diophantine equations	2. linear diophantine equations
3. minimal set	
4. minimal supporting set	
5. linear constraints	3. linear constraints
6. natural numbers	
7. strict inequations	4. strict inequations
8. nonstrict inequations	5. nonstrict inequations
9. upper bound	6. upper bound
	7. set of natural number
10. corresponding algorithms	8. corresponding algorithms
11. considered types	9. considered types
12. mixed types	10. mixed types

IV. IMPROVEMENT OF RAKE AND DISCUSSION

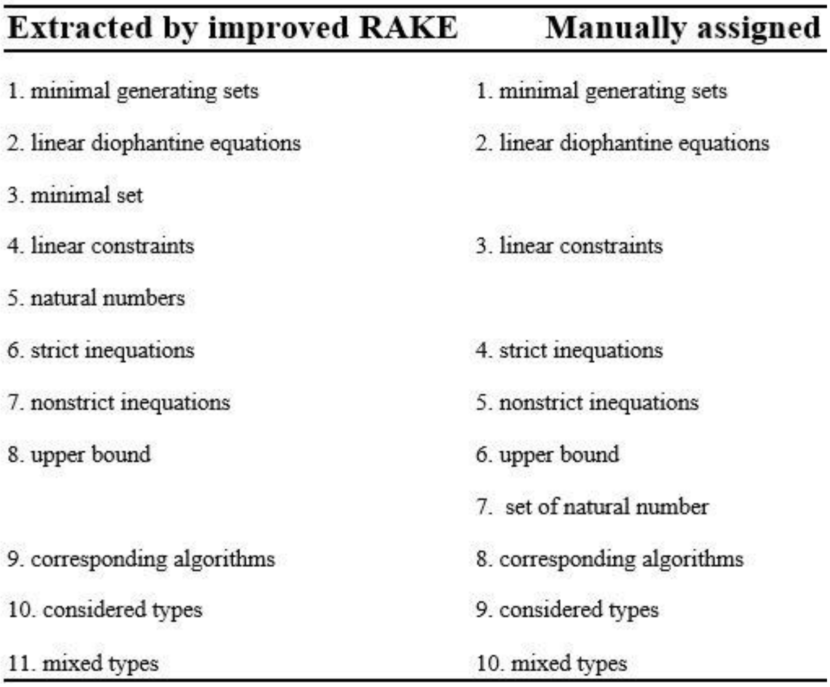
In this section, an effort is given to improve the performance of RAKE method by introducing the synonym word dictionary. A list of different word with their synonyms is kept. Then the list of keywords is checked with the dictionary. If it matches with the dictionary list then it changes the words with the match in such a way that all synonyms will have same word. Then it is called the "stemmer function" which is converting all the plural word to singular one. So the entire keyword plural will be converted to singular. The "stemmer function" is used because in any extracted keyword which is obtained, it checks whether there exists some keywords which differs only because they are plural, for example suppose "set" is a key word and "sets" is another key word then total number of keyword will be two which actually should be one. If plural will be removed then those two keywords will be the same and will be considered one, and finally we can reduce the number of keywords. Stemmer function uses 6 steps to convert all plural to singular. At first step it finds words ending with 'ed' or 'ing' if there is any such word then it removes the ending suffix. In the second step if the ending letter is 'y' then it converts to 'i'. In third step it maps double suffices to single ones. For example '-ization' maps to '-ize' etc... In step four it deals with '-ic-',' –full', '-ness' etc. where similar strategy to step3 is applied. In fifth step it takes off '-ant', '-ence' etc. And finally in sixth step it removes a final '–e'. We have applied all six steps to convert all plural keyword to singular.

In this case, the entire list of keywords are compared after applying stemmer function, if anyone keyword is same then it got eliminated. Thus the list of keyword is reduced further. Implementation of this improvement with the help of above example has produced the following output as shown below in TABLE II. As it can be seen that, after implementing the improvement we are reducing the list of keyword, which is closer to the manually generated keyword list. This is the main achievement of the proposed extension.

V. CONCLUSIONS

Keyword extraction is a powerful tool which enables us to scan a large document collections efficiently. Manual assignment of quality keywords is error-prone, time-consuming and expensive. Automatic keyword extraction enables us to identify a small set of words, key phrases, keywords, or key segments from a textual document with the help of which the meaning of the document can be described. In this paper, a popular keyword extraction technique, RAKE is studied in detail and a novel extension of RAKE is proposed so as to improve the performance of RAKE further. Experimental result shows that the proposed improvement generates reduced list of keyword, which is closer to the manually generated keyword list.

Table II List Of Keywords By Improved Rake And By Manually Intervention



Extracted by improved RAKE	Manually assigned
1. minimal generating sets	1. minimal generating sets
2. linear diophantine equations	2. linear diophantine equations
3. minimal set	
4. linear constraints	3. linear constraints
5. natural numbers	
6. strict inequations	4. strict inequations
7. nonstrict inequations	5. nonstrict inequations
8. upper bound	6. upper bound
	7. set of natural number
9. corresponding algorithms	8. corresponding algorithms
10. considered types	9. considered types
11. mixed types	10. mixed types

REFERENCES

[1] Andrade M and Valencia A, “Automatic extraction of keywords from scientific text: application to the knowledge domain of protein families”, Bioinformatics, 1998, 14(7), 600–607.

[2] Cohen J. D., “Highlights: Language and Domain-independent Automatic Indexing Terms for Abstracting”, Journal of the American Society for Information Science, 46(3): 162 – 174, 1995

[3] Engel D, Whitney P, Calapristi A and Brockman F, Mining for emerging technologies within text streams and documents. Proceedings of the 9th SIAM International Conference on Data Mining 2009: Proceedings in Applied Mathematics, vol. 3, pp. 1291-1301. SIAM, Philadelphia, PA, 2009

[4] Gutwin C, Paynter G, Witten I, Nevill-Manning C and Frank E, “Decision Support Systems-Improving browsing in digital libraries with keyphrase indexes”, 27(1–2), 81–104, 1999

[5] Hulth A., “Improved automatic keyword extraction given more linguistic knowledge”, In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'03), 216 – 223, Sapporo, 2003

[6] Hulth A, “Combining machine learning and natural language processing for automatic keyword extraction”, PhD Thesis, Stockholm University, Faculty of Social Sciences, Department of Computer and Systems Sciences, 2004

[7] Jones K, “A statistical interpretation of term specificity and its application in retrieval”, Journal of Documentation, 28(1), 11–21. 1972

[8] Jones S and Paynter G, “Automatic extraction of document keyphrases for use in digital libraries: evaluation and applications”, Journal of the American Society for Information Science and Technology, 53(8), 2002

[9] Matsuo Y and Ishizuka M, “Keyword extraction from a single document using word co-occurrence statistical information”. International Journal on Artificial Intelligence Tools. 13(1), 157–169, 2004

[10] Mihalcea R and Tarau P, “Textrank: Bringing order into texts”, In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 2004

[11] Rose S., Engel D., Cramer N., Cowley W., “Automatic keyword extraction from individual documents”, Text Mining: Applications and Theory edited by Michael W. Berry and Jacob Kogan, John Wiley & Sons Ltd, 3 – 20, 2010

[12] Salton G, Wong A and Yang C, “A vector space model for automatic indexing”, Communications of the ACM, 18(11), 613 – 620, 1975

[13] Suzuki Y., Fukumoto F., Sekiguchi Y., “Keyword extraction of radio news using term weighting with an encyclopedia and newspaper articles”, Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, 373 - 374, New York, USA, 1998.

[14] Whitney P, Engel D and Cramer N, “Mining for surprise events within text streams”. Proceedings of the Ninth SIAM International Conference on Data Mining, Society for Industrial and Applied Mathematics, 617–627, 2009

[15] Zhang C, Wang H, Liu Y, Wu D, Wang B, “Automatic Keyword Extraction from Documents Using Conditional Random Fields”, Journal of Computational Information Systems, 4(3), 1169 – 1180, 2008

[16] Zhang K., Xu H., Tang J., Li J. Z.. Keyword Extraction Using Support Vector Machine. In: Proceedings of the Seventh International Conference on Web-Age Information Management (WAIM2006), Hong Kong, China, 85 – 96, 2006