Automatic Keyword Extraction From Any Text Document Using N-gram Rigid Collocation

Bidyut Das, Subhajit Pal, Suman Kr. Mondal, Dipankar Dalui, Saikat Kumar Shome

Abstract*— An unsupervised method for extracting keywords from a single document is proposed in this paper. A fuzzy set theoretic approach, fuzzy n-gram indexing, is used to extract n-gram keywords. It is noticed that n-gram keyword renders a better result as compared to mono-gram keyword, but for some documents the most relevant keyword is mono-gram. This paper focuses on a keyword extraction approach which neither requires a dictionary or thesaurus nor does it depend on the size of text document. The algorithm is efficient enough to dynamically determine the mono-gram, bi-gram as well as n-grams keywords for different documents.

***Index Terms*— Keyword extraction; n-gram collocation, fuzzy set; information retrieval, natural language processing.**

I. INTRODUCTION

Keyword plays a crucial role in extracting the correct information as per user requirements. Everyday thousands of books and papers are published which makes it very difficult to go through all the text material; instead there is a need of good information extraction method which can find the required text document. As such effective keywords are a necessity. Since keyword is the smallest unit which express meaning of the entire document, many applications can take advantage of it such as automatic indexing, text summarization, information retrieval, classification, clustering, filtering, cataloging, topic detection and tracking, information visualization , report generation , web searches etc. [1-4].

Keyword extraction is the task of identifying the most relevant words or phrases in a document [5-6]. There are several types of keyword extraction, which may be broken into three categories: statistical methods, linguistic methods, and mixed methods. Statistical methods tend to focus on non-linguistic features of the text such as term frequency, inverse document frequency, and position of a keyword. The benefits of purely statistical methods are their ease of use, limited computation requirements, and the fact that they do generally produce good results. However, as is shown in a number of the articles herein, methods which pay attention to linguistic features such as part-of-speech, syntactic structure and semantic qualities tend to add value, functioning sometimes as filters for bad keywords [7].

Manuscript received on May, 2013.

Bidyut Das, Dept. Of IT, Haldia Institute Of Technology, Haldia, India.
Subhajit Pal, Students Of IT Dept., Haldia Institute Of Technology, Haldia, India.
Suman Kr. Mondal, Students Of IT Dept., Haldia Institute Of Technology, Haldia, India.
Dipankar Dalui, Students Of IT Dept., Haldia Institute Of Technology, Haldia, India.
Saikat Kumar Shome, Scientist, CSIR-Central Mechanical Engineering Research Institute, Durgapur.

Some of the linguistic methods are in fact mixed methods, incorporating some linguistic methods with common statistical measures such as term frequency and inverse document frequency [8].

Recently, numerous documents have been made available electronically. Domain-independent keyword extraction, which does not require a large corpus, has many applications. For example, if one encounters a new Web page, one might like to know the contents quickly by some means, e.g., highlighting keywords. If one wants to know the main assertion of a paper at hand, one would want to have some keywords. In these cases, a keyword extraction without a corpus is very useful. Word count is sometimes sufficient for document overview [9]; however, a more powerful tool is desirable. The paper represents a fuzzy based, unsupervised, domain-independent, and language-independent algorithm for extracting n-gram keywords from any individual text document.

II. MOTIVATION

A collocation is just a set of words occurring together more often than by chance in a text document. Collocation is a syntactic and semantic unit whose exact and unambiguous meaning or connotation cannot be derived directly from the meaning or connotation of its components [10]. In other word Collocation is lexically and/or pragmatically constrained recurrent co-occurrences of at least two lexical items which are in a direct syntactic relation with each other. Collocation can be of two types i.e. rigid or flexible. Rigid collocation are those n-grams that always occur side by side and appear in same order whereas flexible collocation are n-grams that can have intervening words placed between them or can occur in different order [11]. Our motivation is based on top ranked N-gram rigid collocations that carry the clue of the document and can be treated as keywords.

III. MOST POPULAR METHODS FOR EXTRACTING COLLOCATION

A. Point-wise Mutual Information (PMI)

The PMI has been utilized to find the closeness between word pairs [12]. PMI for two events x and y is defined as

$$I(x,y) = \log_2[\frac{P(x,y)}{\{P(x).P(y)\}}].$$

If w_1 and w_2 are written for the first and second word respectively, instead of x and y, then the PMI for the two words w_1 and w_2 is given by

$$I(w_1,w_2) = \log_2[\frac{P(w_1,w_2)}{\{P(w_1).P(w_2)\}}]$$

where $P(w_1,w_2)$ is the probability of two words w_1 and w_2 coming together in a certain text and $P(w_1)$ and $P(w_2)$ are the probabilities of w_1 and w_2

appearing separately in the text, respectively. If $P(w_1, w_2) = P(w_1).P(w_2)$ that is, the two words are independent to each other, then $I(w_1, w_2) = 0$ which indicates that these two words are not good candidates for collocation. A high PMI score signifies the presence of a collocation.

B. T-score

The t-test has been used for collocation discovery to test the validity of a hypothesis [12]. For that purpose, we formulate a null hypothesis H_0 that the two words w_1 and w_2 appear independently in the text. So under the null hypothesis H_0, the probability that the words w_1 and w_2 are coming together is simply given by: $P(w_1, w_2) = P(w_1).P(w_2)$. The null hypothesis has been tested by using t-test. If the null hypothesis is accepted, we conclude that the occurrence of two words is independent of each other. Otherwise, we may conclude that they depend on each other, that is, they form collocations. In t-test we use the null hypothesis that the sample is drawn from a distribution with mean μ, taking sample mean and variance into account. The t-test considers the difference between the observed and expected mean. The t statistic is defined as: $t = \bar{x} - \frac{\mu}{\sqrt{\left(\frac{s^2}{N}\right)}} \approx t_{n-1}(\alpha)$ where x is the sample mean, s^2 is the sample variance, N is the sample size, μ is the mean of the distribution and $t_{n-1}(\alpha)$ denotes a t-distribution with $(n-1)$ degrees of freedom at α level of significance. To apply t-test for testing the independence of two words w_1 and w_2, we assume that $f(w_1)$, $f(w_2)$ and $f(w1, w2)$ are the respective frequencies of the word w_1, w_2 and w_1w_2 in the corpus and N is the total number of words/bigrams in the text document. Then, we have,
$P(w_1) = f(w_1)/N$ (say p_1), $P(w_2) = f(w_2)/N$ (say p_2), $P(w_1w_2) = f(w_1w_2)/N$ (say p_{12}). The null hypothesis is
H_0: $P(w_1, w_2) = P(w_1).P(w_2) = p_1.p_2$
If we select bigrams (word pairs) randomly then the process of randomly generating bigrams of words and assigning 1 to the outcome that the particular word combination for which we are looking for is a collocation and 0 to any other outcome follows a Bernoulli distribution. For the Bernoulli distribution we have Mean $(\mu) = p$ and Variance $(\sigma^2) = p(1-p)$. Thus, if the null hypothesis is true, the mean of the distribution is $\mu = p_1.p_2$. Also, for the sample, we have $P(w_1, w_2) = p_{12}$. Therefore, using Binomial distribution, sample mean $\bar{x} = p_{12}$ and sample variance $s^2 = p_{12}(1 - p_{12})$. Then calculate the value of $|t|$ and compare it with the tabulated value at given level of significance. If the value of $|t|$ for a particular bigram is greater than the value obtained from the table, we reject the null hypothesis, which indicates that the bigram may be considered as a collocation.

IV. PROPOSED METHODOLOGY

Based on fuzzy sets [13], a new approach is proposed to find collocation from text document. As mentioned earlier, a collocation is just a set of words occurring together more often than by chance in a corpus. Collocations are extracted based on the frequency of the joint occurrence of the words as well as that of the individual occurrences of each of the words in the whole text. Intuitively, when a set of words is extracted as a collocation, then the joint occurrence of the words must be high in comparison to that of the constituent individual words. Researchers modeled this intuition as the ratio of the joint word appearance value to the individual word appearance values, either explicitly or implicitly. It is to be realized that the concept of high occurrence is imprecise or vague in nature as it depends on many factors such as the text size, occurrences of other words or word combinations in the text document. Accordingly, the concept of fuzzy set theory is brought here to manage such impreciseness in representing the notion of high occurrences. Here two fuzzy sets, namely High Word Occurrence (HWO) and High Word Pair Occurrence (HWPO) are considered where HWO corresponds to an individual word and HWPO corresponds to as adjacent word combination. Membership functions for such fuzzy sets are decided based on the word appearance statistics in the text. Finally, the said fuzzy membership values are combined to define the Fuzzy Bi-gram Index (FBI). FBI assigns a value in [0, 1] characterizing the degree of an adjacent word pair to be a bi-gram (collocation of length 2). Adjacent bigrams are analyzed later to determine collocation of higher length i.e., n-grams where $(n > 2)$.

A. High Word Occurrence (HWO)

The $F_{iw}(i)$ is the number of words appeared i times in the text document. Usually the value of $F_{iw}(i)$ decreases with the increase of i. It is observed that the value of $F_{iw}(i)$ is considerably high for $i = 1$ with respect to other values of i. The actual value of $F_{iw}(i)$ is replaced by the average of other appearance values. HWO is a fuzzy set which corresponds to an individual word and its membership value represents the degree of being high appearance of the word in the text. Based on the occurrences of all individual words in the text, the membership function of HWO is decided as $\mu HWO(w) = C_{iw}(n)/C_{iw}(N_{max})$ where C_{iw} is the cumulative frequency of the individual word w, n is the number of occurrence of the word w and N_{max} is the maximum number of occurrence of any word in the text. Here, $C_{iw}(n) = \sum_{i=1}^{n} F_{iw}(i)$ when $F_{iw} \neq 0$. Obviously $0 < \mu HWO(w) \leq 1$ and μHWO is 0 for all those words which are absent in the text document and it is 1 for the words which occurred maximum number of times in the text document.

B. High Word Pair Occurrence (HWPO)

The $F_{wp}(i)$ is the number of word pairs occurred i times in the text document. For the same purpose the actual value of $F_{wp}(1)$ is replaced by the average of other appearance values. HWPO is the second fuzzy set which corresponds to a word pair and its membership value represents the degree of being high occurrence of the word-pair in the text document. Based on the occurrence of all word pairs in the text document, its membership function is decided as
$\mu HWPO(w1, w2) = C_{wp}(m)/C_{wp}(M_{max})$ where C_{wp} is the cumulative frequency of the word pair $(w1, w2)$, m is the number of occurrence of the word pair $(w1, w2)$ and M_{max} is the maximum number of occurrence of any word pair in the text document. Here, $C_{wp}(m) = \sum_{i=1}^{m} F_{wp}(i)$ when $F_{wp} \neq 0$. Obviously $0 \leq \mu HWPO(w1, w2) \leq 1$ and $\mu HWPO$ is 0 for all those word pairs which are absent in the text document and it is 1 for the word pairs which occurred maximum number of times in the text document.

C. Bi-grams and Fuzzy Bi-gram Index (FBI)

The membership values of above two fuzzy sets HWO and HWPO are combined to define a collocation measure, named as Fuzzy Bi-gram Index (FBI). It is to be noted that the degree of a word pair of $(w1, w2)$ to be a bi-gram is directly proportional to $\mu HWPO(w1, w2)$ and inversely proportional to the values of $\mu HWO(w1)$ and $\mu HWO(w2)$. Accordingly,

$FBI(w1, w2) = \mu HWPO(w1, w2)[1 - \alpha(\mu HWO(w1) + \mu HWO(w2))]$ where $\alpha \in [0, 0.5]$.

Here $FBI(w1, w2) \in [0, 1]$, and it provides a measure to be a bi-gram. That is, the more is the value of $FBI(w1, w2)$, the more is the possibility of $(w1, w2)$ to be a bi-gram. A word pair of $(w1, w2)$ is identified as a bi-gram if the value of $FBI(w1, w2)$ is greater than a threshold value which is taken as 0.5 in this experiment. One may decide any other threshold value depending on requirements.

D. N-grams and Fuzzy N-gram Index (FNI)

N-gram keywords are formed after all bi-grams have been extracted and corresponding FBI have been assigned to them. If two bi-grams appear in the text adjacently we further analyze it to form n-grams. For example consider that "cricket world" and "world cup" are two adjacent bi-grams (i.e. there is no other word in between them in the original text). So for further analysis we check whether the last word of the first candidate and the first word of the second candidate are same; if this condition is satisfied we check if both the bi-grams have their respective FBI > 0.5 or not. The two bi-grams are combined to form a tri-gram only if all these conditions are proved to be true. The fuzzy index of the newly created tri-gram is calculated by averaging the index value of constituent bi-grams. This entire procedure is recursively applied to extract n-gram keywords.

E. Keyword Extraction

The algorithm proposed in this paper has three distinct phases. The first phase extracts a list of bi-grams and arranges them in descending order based on their FBI. The second phase is responsible for extracting the n-grams and arranges them in descending order of respective FNI. In the third phase the algorithm checks whether it is necessary to include monogram keywords. There are some documents for which the most relevant keywords are monograms. For example a document on "Tiger", "Sun", "football", etc. must have monogram keywords like 'tiger', 'sun', and 'football' respectively; any bi-gram or n-gram keyword will be less relevant for such documents. On the other hand, for documents on "Static web page" or "cricket world cup", there is no mono-gram keyword which can give proper clue about the document. For this type of documents, bi-gram or tri-gram keywords best serve the purpose, thus mono-grams must not be included. This logic is implemented by using a frequency measure approach. We include those mono-grams whose frequency is greater than eight times the highest frequency of Bi-grams. The final result set contains the top 5 mono-grams (if included by the algorithm), top 10 bi-grams and the top 10 n-grams. To improve the efficiency of the algorithm, an additional effort is made to find some highly relevant n-gram keywords and if found these are placed before the bi-grams in the final result set. Any n-gram having FNI greater than the average FBI of the top 10 bi-grams is considered as highly relevant n-gram keyword. Therefore the result set of any document will have less than or equal to 25 keywords and it will be arranged according to relevance of the keyword (i.e. the most relevant keyword will appear at the beginning whereas the less relevant keywords will appear at the bottom of the list.)

V. EXPERIMENTAL RESULTS

A. Data Set

The proposed method is applied on a large number of text documents (including web documents and scientific articles). The data set consists of 500 web documents related to various topics from different websites. These include Wikipedia on "Android", "Indian National Congress", "Cricket World Cup" etc. The extracted keywords are arranged in descending order based on their respective FNI. Then the top ranked keywords

B. Preprocessing

The text document is preprocessed in such a way that the frequency of co-occurrence word-pair is easily counted from the text document and reduces the effort of extracting collocation. The preprocessing technique consists of the following steps. The HTML tags and scripts are removed if it is a web document. Secondly special characters including numeric digits are eliminated. Thirdly, the stop-words are removed. It is noticed that if any special character or stop-word is present in between any two consecutive words then they are not consider as a co-occurring word-pair. Therefore each line is broken wherever a stop-word or a special character appears. Finally discard the line that contains only a single word because single word cannot form collocation. Then the word pair is identified from each line and the frequency is counted.

C. Evaluation Method

Precision, Recall and F-measure are used to evaluate the performance of our experimental work [14]. Precision can be seen as a measure of exactness or fidelity, whereas recall is a measure of completeness. In keyword extraction, the precision is the proportion of true positive (i.e., keyword of the desired type) among the n keywords and how many of all suitable keywords that could have been extracted from the text are actually found in the n-best list are called recall. Higher the source of precision and recall, better the algorithm is. To compare the result set generated by our algorithm we had to manually extract some keywords based on intuition. The experiment was performed with 10 people. Each participant was asked to manually extract the keyword from 10 different web documents. Then these manually extracted result sets were compared with the result sets generated by the algorithm. Then the Precision values were calculated for each of those text documents. Altogether a total of 100 documents have been tested upon. The outcome of the experiment was intuitively satisfying.

D. Result Set

The algorithm generates Bi-gram and N-gram (n>2) keywords. In addition to these some mono-gram keywords are also generated. The algorithm dynamically determines whether it is necessary to include mono-gram keywords or not.

The result set consists of at most 25 Keywords. Our extraction method archive overall precision of 95% (approx.). Recall is not taken because the proposed approach focuses on the exactness of top 25 keywords. Some sample result sets has been shown below.

Web Document: Mobile phone
Hyperlink: http://en.wikipedia.org/wiki/Mobile_phone
Mono-gram Keywords: (NOT RELEVANT, HENCE NOT EXTRACTED)
Highly Relevant N-gram Keywords: (NOT FOUND)
Bi-gram Keyword : mobile phone, mobile phones, cell phone, cell phones, sim card, text messaging, gpp family, main article, mobile banking, health organization.
N-gram Keywords: Mobile phone subscribers, cell phone subscribers, Worldwide, Mobile Phone, mobile phone operator, World Health, Organization, Global mobile phone subscribers, Worldwide, Mobile Phone Sales, mobile phone radiation, mobile phone sales, SMS text messaging.

Web Document: Cricket World Cup
Hyperlink: https://en.wikipedia.org/wiki/Cricket_World_Cup
Mono-gram Keywords: (NOT RELEVANT,HENCE NOT EXTRACTED)
Highly Relevant N-gram Keywords: Cricket World Cup
Bi-gram Keyword : world cup, affiliate members, west indies, cricket world, sri lanka, south Africa, international cricket, day international, cricket council, cup qualifier.
N-gram Keywords: World Cup Qualifier, ICC Cricket World Cup, Cricket World Cup final, Day International cricket, international Cricket Council, ICC Cricket World, World Cup final, World Cricket League, World Cup finals.

Web Document: Android
Hyperlink: http://en.wikipedia.org/wiki/Android_(operating_system)
Mono-gram Keywords: android
Highly Relevant N-gram Keywords: open source code Android Open Source
Bi-gram Keyword : Open source, operating system, android devices, google play, ars technica, source code, linux kernel, jelly bean, android open, handset alliance..
N-gram Keywords: Android Open Source Project, Google Play store, Open Handset Alliance, Smartphone market share, mobile operating system, Android market share, Ice Cream Sandwich, mobile operating systems.

VI. CONCLUSION

In this paper we have shown a fuzzy based algorithm to extract keywords from any text document. The strength of our algorithm lies in the fact that it neither requires any corpus nor does it depend on the size of the text document. The algorithm is efficient enough to determine the highly relevant keywords from a document. The efficiency of the algorithm can be further enhanced by using part-of-speech filtering. In the present scenario the number of electronic documents is increasing rapidly, so we believe our method will prove to be very useful in numerous applications where corpus is not available.

REFERENCES

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

[2] S. Jones and G. W. Paynter, "Automatic extraction of document keyphrases for use in digital libraries: Evaluation and applications," Journal of the American Society for Information Science and Technology, vol. 53, no. 8, pp. 653-677, 2002.

[3] K. Coursey, R. Mihalcea, and W. Moen, "Automatic keyword extraction for learning object repositories," in Proc. Conf. Amer. Soc. Inf. Sci. Technol., 2008.

[4] L. Plas, V. Pallotta, M. Rajman, and H. Ghorbel. 2004. Automatic keyword extraction from spoken text. A comparison of two lexical resources: the EDR and WordNet. In Proceedings of the LREC.

[5] Y. Matsuo and M. Ishizuka, "Keyword extraction from a single document using word co-occurrence statistical information," International Journal on Artificial Intelligence Tools, vol. 13, no. 1, pp. 157-169, 2004.

[6] Y. HaCohen-Kerner, "Automatic extraction of keywords from abstracts," in Proc. 7th Int. Conf. Knowledge-Based Intell. Inf. Eng. Syst., 2003, vol. 2773, pp. 843-849.

[7] A. Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In Proceedings of EMNLP, pages 216-223.

[8] A Hulth, 2004, Combining machine learning and natural language processing for automatic keyword extraction. Stockholm University, Faculty of Social Sciences, Department of Computer and Systems Sciences (together with KTH).

[9] H. P. Luhn, *A statistical approach to mechanized encoding and searching of literary information*, IBM Journal of Research and Development, Vol. 1(4), 1957, Pages 309-317

[10] Y. Choueka, *Looking for Needles in a Haystack or Locating Interesting Collocational Expressions in Large Textual Databases*, In Proceedings of RIAO'1988, 1988, pages 609-624.

[11] Goldstein, Ira. Collocations in Machine Translations [Internet]. Version 4. Knol. 2008 Jul 27.

[12] Church, Kenneth W. and Hanks, *Patrick, Word association norms, mutual information and lexicography*, Computational Linguistics, 1990, Vol. 16(1), pages 22-29

[13] R.E. Bellman, L. A. Zadeh, *Decision making in fuzzy environment*, Management Science, Vol. 17(4), 1970, pages 141-164

[14] J. Makhoul, F. Kubala, R. Schwartz and R. Weischedel, *Performance Measures For Information Extraction*, In Proceedings of DARPA Broadcast News Workshop, 1999, pages 249-252

[15] Y. HaCohen-Kerner, Z. Gross, and A. Masa, *Automatic extraction and learning of keyphrases from scientific articles*, Comput. Linguist. Intell. Text Process, pages 657-669, 2005.

[16] Y. HaCohen-Kerner, I. Stern, D. Korkus, and E. Fredj, "Automatic machine learning of keyphrase extraction from short html documents written in Hebrew," Cybern. Syst., 2007, Vol. 38(1), pages 1-21

[17] F. Liu, F. Liu, and Y. Liu, *Automatic keyword extraction for the meeting corpus using supervised approach and bigram expansion*, In Proc. IEEE Workshop Spoken Lang. Technol., 2008, pages 181-184

[18] Q. G. Zhang, D. J. Xue, Z. H. Zhang, J. Y. Zhang, *Automatic Keyword Extraction from Massive Data Sets Based on Feature Combination*, Journal of the China Society for Scientific and Technical Information, 2006, Vol. 25(5), pages 587-593

Bidyut Das is an Asst. Professor of the Haldia Institute of Technology, India. He was born on 8th February, 1982 in India. He received his B.Sc. and M.Sc. degrees in Computer Science from Vidyasagar University, India, in 2002 and 2004 respectively; He did his M.Tech in Information Technology from School of IT, West Bengal University of Technology, India in 2008. He received Gold Medal in M.Sc and Silver Medal in M.Tech. His research interests include topics in natural language processing, text mining, machine learning, pattern recognition and image processing.

Subhajit Pal is currently pursuing B.Tech in IT from Haldia Institute of Technology (Year: 2009-2013). He was born on 14th January, 1991 in India. He completed his schooling form The Assembly of God Church School, Haldia, securing 86.6% in ICSE 2007 and 81.25% in ISC 2009. He is highly innovative, self motivated and has expertise in various programming languages. His research interests lies in the field of text mining, natural language processing, image processing and pattern recognization.

Saikat Kumar Shome received his Bachelor of Technology Degree in Computer Science and Engineering from Future Institute of Engineering and Management, Kolkata, India in 2010 and M.Tech in Mechatronics from Academy of Scientific and Innovative Research in 2012. Presently, he is engaged as a Scientist and persuing PhD at CSIR-Central Mechanical Engineering Research Institute, Durgapur, India. His research areas include Mechatronics, Bio-Medical Image Processing, Network Security and VLSI Signal Processing.