A Survey on Keyword Extraction Approaches

Rasika Arun Londhe[1], Mrs. Vidya Nikam[2]

[1]*T.E. Student, Computer Dept., D. Y. Patil College Of Engineering, Akurdi, Pune*
[2]*Professor, Computer Dept., D. Y. Patil College Of Engineering, Akurdi, Pune*

Savitribai Phule Pune University
University in Pune
India
[1]rasika_londhe@yahoo.in
[2]nikamvidya24@gmail.com

ABSTRACT: This paper focuses on various computing approaches used for the keyword extraction from any document, conversations, Facebook, twitter profiles etc. Keyword extraction techniques are essentially used for finding those words which can best describe the subject of a document. Extracting keywords manually is a very tedious and time-consuming process for a human so much so that it is nearly impractical to achieve it with limited manpower as the size of information is growing at a tremendous rate day by day. So, as a result, automated computing systems are required to do this task skilfully. Keyword extraction task is vital in areas like Text mining, Information Retrieval, and Natural Language Processing. Keyword extraction from text data is a commonly used by search engines to quickly categorize and locate specific data based on supplied keywords.

KEYWORDS: keywords/key phrases, keyword extraction approaches, information retrieval, recommender systems, unsupervised approaches.

I. INTRODUCTION

Keyword/key phrase is a sequence of one or more words that provide a concise yet particular representation of what is the document about. Ideally, keywords represent in brief format the essential content of a document. Keywords may, for example, serve as a dense summary for a document, lead to improved information retrieval, or be the entrance to a document collection. [1] However, relatively few documents provide keywords. This is especially true for spoken documents. Current speech recognition /Automatic speech recognition (ASR) system performance has improved significantly, but there is no rich structural information such as topics and keywords in the transcriptions. Therefore, there is a requirement to automatically generate keywords for the huge amount of written or spoken documents available now.

Also, Users, nowadays are flooded with irrelevant and extraneous information which is nowhere related to their point of interest and which might result in dissatisfaction of the customer. If the process of recommending documents, advertisements or anything in particular, to users, which they find relevant, could be mitigated, it would open tons of new opportunities for businesses, and increase customer retention in fields like advertisement industries, online shopping stores etc. So, in all Keyword extraction approaches are very much essential in providing satisfaction to users.

Extracting keywords benefit readers as they can judge more quickly whether the text is worth reading. Websites creators benefits from keywords as they can group similar content by its topics. Similarly, algorithm programmers benefit from keywords as they reduce the dimensionality of text to the most important features. [2]

A typical keyword extraction algorithm works as follows:

1. Candidate selection: Here, all possible words, phrases terms, or concepts (depending on the task) that can potentially be keywords are extracted.
2. Properties calculation: For each candidate, properties that indicate that it may be a keyword or not are calculated. For example, a candidate appearing in the title of a book is likely to be keyword.
3. Scoring and selecting keywords: All candidates are scored by either combining the properties into a formula or using a machine learning technique to determine the probability of a candidate being a keyword. A score or probability threshold or a limit on the number of keywords is then used to select the final set of keywords.

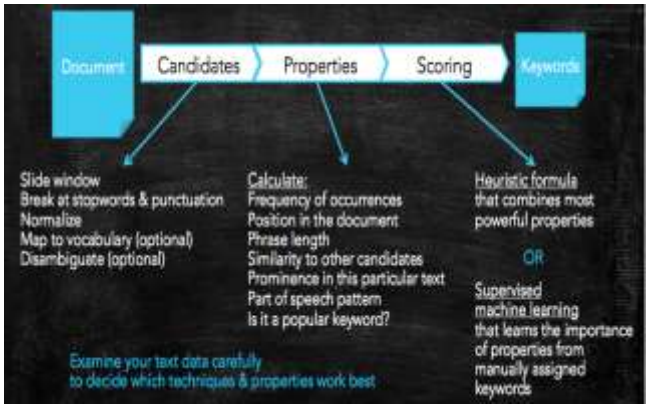

Fig. 1 Process of Keyword Extraction

A set of stop words or common words which do not participate in forming the keywords/key phrases like a, an, the, that, is, of etc. are provided as an input to the keywords extraction algorithm which then eliminates the stop words and forms a meaningful document which are further used for extracting the keywords.

Advantages of keyword extraction are:

I. To maintain multiple hypotheses about users information requirements/needs.
II. To present a small sample of recommendations based on the most likely ones.
III. To extracting a relevant and diverse set of keywords, cluster them into topic specific queries ranked by importance, and present users a sample of results from these queries.

Recommender systems attempt to predict items in which a user might be interested, given some information about the user's and items' profiles. Most existing recommender systems use content-based or collaborative filtering methods or hybrid methods for keywords extraction. [3][4]

Recommender systems addresses the problem of using consumer opinion about products/items, expressed online in free-form text, to generate product recommendations. [4]

Various approaches for keyword extraction are:

1. Lexical chain analysis
2. Graph based keyword extraction
3. Neural based approach
4. Word co-occurrence

II. MOTIVATION

Information is the most powerful weapon in the modern society. Every day the internet is overflowed with a huge amount of data in form of e-newspaper articles, emails, web-pages, reviews and search results. Often, information received by users is incomplete, such that further search activities are required to enable correct interpretation and usage of this information. Keyword Extraction is a usable and powerful tool which enables efficient scanning of large document collections for defining the particular subject of the document. It helps users to know that whether a document is of their use or not.

III. LITERATURE WORK

i. Lexical chain analysis:

A lexical chain is a sequence of parallel/interrelated words, independent of the grammatical structure of the text and in effect, it is a list of words that relates sentences via thesaurally- related nouns. Lexical chains are computed by grouping/chaining set of words that are semantically related. [5]

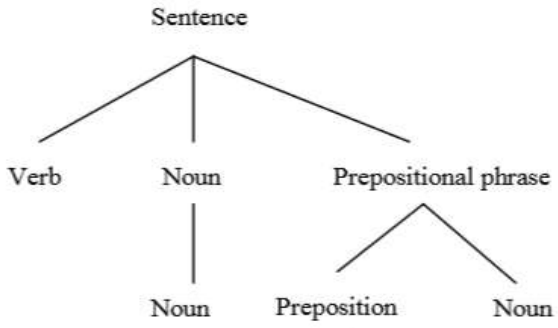

Fig. 2 A Lexical Chain format

Steps to be followed to create a lexical chain are:

1. Select a set of candidate keywords.
2. For each candidate keyword, find an appropriate chain relying on relatedness criterion among members of the chains.
3. If it is found, insert the word in the chain and update it accordingly.

In this approach keywords are extracted using the following features that are:

a. First occurrence position
b. Word frequency
c. Last occurrence position
d. Lexical chain score of a word
e. Direct lexical chain score of a word
f. Lexical span score of a word
g. Direct lexical span score of a word

Lexical Chain Score of a Word:

A word can be a part of more than one lexical chain. The score is assigned to these words. Then the word that has the maximum score in any lexical chain is chosen as the lexical chain score of the word.

Direct Lexical Chain Score of a Word:

This is calculated by scoring only the relations that belong to the word.

Lexical Span Score of a Word:

The lexical span score of a lexical chain depends upon the portion of the text that is covered by the lexical chain. This c portion of the text is considered to be the distance between the first occurrence position of a word and the last occurrence position of a word. The span score is computed by finding the difference between these two positions.

Direct Lexical Span Score of a Word:

The score of the chain with maximum score can be considered as the direct lexical span score of the word. This score can be computed as same as the lexical chain span score except that the words that are directly related with the word in the lexical chain.

For example consider the following sentences:

I like iPhones. Apple Company just launched a new iPhone. But, because I like iPhones that doesn't mean that I am going to buy one, I'll buy Google Pixel.

Those 3 sentences are related through:

IPhone ->Apple ->Google pixel

An algorithm named as KEA uses lexical analysis technique and calculates feature values for each candidate, and uses a machine-learning algorithm to predict which candidates are good and appropriate key phrases. [6]

ii. Graph based keyword extraction:

In graph based approach, basically a document is designed as graph where vertices represents terms (words) and edges represents their relations. After the basic text preprocessing like classification into candidate words and stop words removal, a graph is built. The graph contains a single vertex for each distinct word even if it appears more than once in the text. Thus each vertex is unique.

Once a graph is built, clusters of words are identified by locating maximally-connected sub-graphs within the document graph. Candidate keywords are then identified by locating vertices within the graph that have edges/links between two separate clusters. These candidate keywords are then ranked by the probability that for each of the clusters they join, that word was the word used to join the two clusters (effectively, the most common word used to join these clusters). [7]

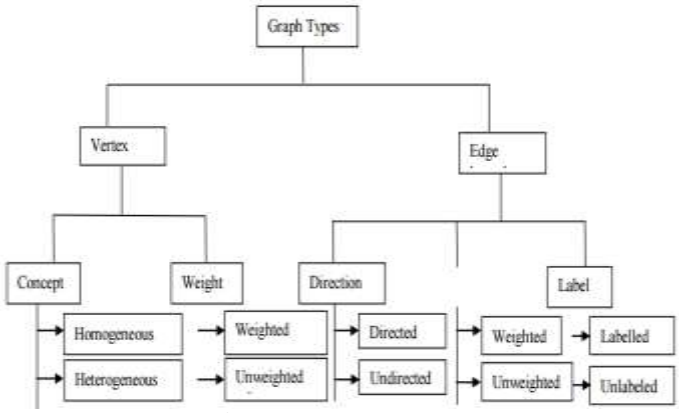

Fig. 3 Types of Graph Based Keyword Extraction Approach

Extraction of keywords from a document using graph based approach works as follows:

1. Graph Construction
2. Word determination
3. Keyword generation

1. Graph Construction:

There is a directed edge from the vertex corresponding to the word x to the vertex corresponding to word y, if a word x immediately precedes a word y in the same sentence somewhere in the document. An edge cannot be created when the sentence terminating punctuation marks are present between two words. [8]

2. Word determination:

Once the graph is constructed, the important word determination step is followed, for which certain centrality measures are applied to assign the rank to each node in a graph. In graph based theory, centrality measures refer to indicators which identify the most important vertices within a graph and that approach is used for ranking the vertices. In the domain of keyword extraction, various centrality measures are used for the task of ranking the words in a text. [9]

3. Keyword generation:

This process includes two steps:

i. Keywords are extracted from the document. The sequence of adjacent words is considered as keyword candidates.

ii. The score of a candidate keyword k is computed by summing the scores of the words it contains normalized by its length+1.

The score can be given by the equation

Score (k) = ($\sum_{word \epsilon k}$ Score (word)) / (length (k) +1) After scoring the keywords, redundant keywords are eliminated and the resulting keywords are ranked by the descending scores of the keywords.

iii. Neural based approach

Before applying neural based approaches, it is important to decide features for the classification.

The different input features required for the classification are as follows:

a. TF (Term Frequency): TF is the occurrence of a word in single document.
b. IDF (Inverted Document Frequency): IDF is the measure of importance of the word in the sample documents.
c. ITF (Inverted Term Frequency): ITF denotes the total frequency of the term/word in sample documents.
d. T (Title): T denotes the presence of the word in the title of the given document.
e. FS (First Sentence): FS denotes the presence of the word in the first sentence of the given document.
f. LS (Last Sentence): LS denotes the presence of the word in the last sentence in the given document.

The features TF, IDF, and ITF, are represented in integers greater than or equal to zero, while T, FS, and LS, are represented in binary values (0 and 1). [10]

The different output features required for the classification are as follows:

a. K (Keyword): If the word is judged as keyword, K is one, else it is zero.
b. N (Non-keyword): If the word is judged as non-keyword, K is zero, else it is one.[11]

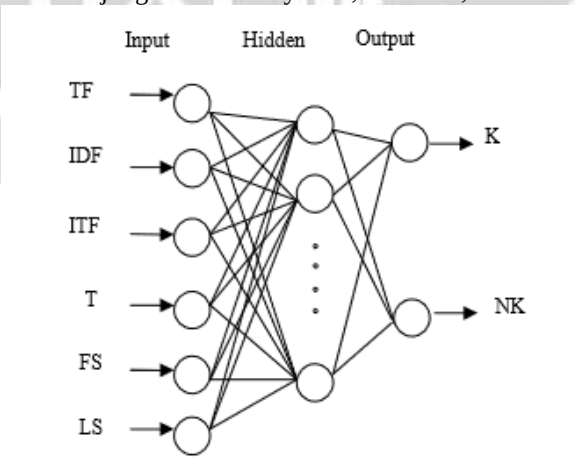

Fig. 4 Architecture of Back propagation to judge keywords

The neural based approach to judge keywords is calculated with equations based on TF (Term Frequency) and IDF (Inverse Document Frequency).

$W_i = TF (\log_2 N - \log_2 IDF + 1)$ ---------- (1)

$W_i = TF^m / (IDF + ITF + 1)$ ---------- (2)

The equation (1) is used to calculate the weight of each word in the document. Then the equation (2) is used to develop two modules that are text categorization and text summarization. The precision for judging keywords in documents with this approach is maximized, when the threshold value is given as maximum. [12]

iv. Word Co-Occurrence

Word co-occurrence takes into account the relation between words. Its aim is to find similarity between words or similarities of meaning among word patterns. The sentences in the document (after removing stop words) are considered as a set of words; it includes title of a document, section title and a caption.

The word frequency is determined by counting the frequent words occurred in a document. The frequencies of the co-occurred words can be represented in N×N matrix format. Co-occurrence distribution [13] shows the importance of terms in a document and the co-occurrence biases are derived from semantic or lexical or from other relations.

Clustering methods are also used for this approach to cluster the frequent words. The words are clustered using similarity distribution of co-occurrence with other words. Co-occurrence words are counted from these clusters and then the expected probability is calculated. [14]Then the statistical value of X^2 is used to measure the degree of biases of distribution, which is calculated using following the formula:

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

In this freq(w,g) denotes frequency of co-occurrence of word w and g. $(freq(w,g)-n_w p_g)$ denotes the difference between predictable frequencies. $n_w p_g$ represents the expected frequency of co-occurrence, in which n_w represents the total number of words in the sentence where w appears and p_g denotes the sum of the total number of words where g appears is proportional to the total number of words in the document. [15]

TABLE I
Comparison of Techniques for Keyword Extraction

Author	**Technique**	**Precision**
Gonenc Ercan, Ilyas Cicekli	Lexical chain	Lowest
Marina Litvak, Mark Last	Graph-based approach	Moderate
Yutaka Matsuo, Mitsuru Ishizuka	Word Co-occurrence	High
Taeho Jo	Neural based approach	Highest

IV. AN ANALYTICAL MODEL

A RAKE i.e. Rapid Keyword Extraction Algorithm is an algorithm for efficient keyword extraction from documents.

RAKE being one of the computing techniques, finds keywords by first parsing a documents' text into a set of candidate keywords. RAKE uses stop words (common words for e.g. 'a', 'an', 'the', etc.) and phrase delimiters (e.g. ',') to partition the document text into candidate keywords, which are sequences of content words as they occur in the text. Then finally the score is calculated and the most appropriate keyword is used to define the subject of document. [16]

The steps followed by RAKE algorithm are as follows:

1. It starts keyword extraction by parsing the text of given document into a set of candidate keywords.
2. It first splits the text in an array of words using the word delimiters, then splits the array into sequences of contiguous words at stop word and phrase delimiters.
3. Next it calculates a 'score' which gives the importance of a keyword. The score for a candidate keyword is defined as the scores of individual member words. Several metrics for calculating word scores are evaluated, based on the degree and frequency of word vertices in the co-occurrence graph:
   (1) Word frequency ($ffffffff$ (ww)),
   (2) Word degree ($ddffdd$ (ww)), and
   (3) Ratio of degree to frequency ($ddffdd$ (ww)/$ffffffff$ (ww)).
4. After the word scores for individual words are calculated, the candidate word scores are found. The score for each candidate keyword is computed as the sum of its member word scores.

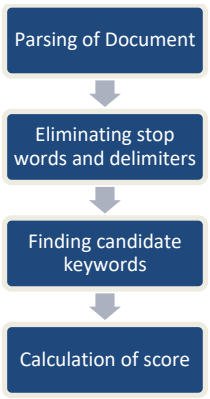

Fig. 5 RAKE algorithm flowchart

where m is the number of hops in the route, TE = TE_{node} is the transmission energy between the nodes. The route having minimum total transmission energy i.e. min (TTE_R) will be selected as energy efficient route.

V. CONCLUSION

This paper represents various approaches available for keyword extraction from the documents. Keywords are used to define or represent the main subject of a document. They serve as a dense summary for a document, lead to improved information retrieval, or be the entrance to a document collection. The approaches elucidated above shows different ways to extract the efficient keywords from documents. Neural based method is used in most of the approaches to identify the frequency of the words. It provides a better precision value when compared to other approaches stated above. In future neural-based approach or some other approaches can be used for extracting the keywords from documents.

REFERENCES

[1] MATSUO, Y., and M. ISHIZUKA. "KEYWORD EXTRACTION FROM A SINGLE DOCUMENT USING WORD CO-OCCURRENCE STATISTICAL INFORMATION". International Journal on Artificial Intelligence Tools 13.01 (2004): 157-169. Web.
[2] D. V. Paul and J. D. Pawar, "A Binomial Heap Extractor for Automatic Keyword Extraction," 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE), Ernakulam, 2016, pp. 113-121.
[3] doi: 10.1109/SAPIENCE.2016.7684154.
[4] S. Aciar, D. Zhang, S. Simoff and J. Debenham, "Informed Recommender: Basing Recommendations on Consumer Product Reviews," in IEEE Intelligent Systems, vol. 22, no. 3, pp. 39-47, May-June 2007, doi: 10.1109/MIS.2007.55.
[5] Y. Z. Wei, L. Moreau and N. R. Jennings, "Learning users' interests by quality classification in market-based recommender systems," in IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 12, pp. 1678-1688, Dec 2005 doi: 10.1109/TKDE.2005.200.
[6] Liu, F., Pennell, D., Liu, F., & Liu, Y. (2009, May). Unsupervised approaches for automatic keyword extraction using meeting transcripts. In Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics (pp. 620-628). Association for Computational Linguistics.
[7] (Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., & Nevill-Manning, C. G. (1999, August). KEA: Practical automatic keyphrase extraction. In Proceedings of the fourth ACM conference on Digital libraries (pp. 254-255). ACM.
[8] Beliga, S., Meštrović, A., & Martinčić-Ipšić, S. (2015). An overview of graph-based keyword extraction methods and approaches. Journal of information and organizational sciences, 39(1), 1-20.
[9] Nagarajan, R., Dr. S. Anu H. Nair, Dr. P Aruna, and N. Puviarasan. "Keyword Extraction using Graph Based Approach." International Journal of Advanced Research in Computer Science and Software Engineering 6.10 (2016): n. pp 25-29. Web.
[10] Kim, Y., Kim, M., Cattle, A., Otmakhova, J., Park, S., & Shin, H. (2013). Applying Graph-based Keyword Extraction to Document Retrieval. In IJCNLP (pp. 864-868).
[11] Azcarraga, A., Liu, M. D., & Setiono, R. (2012, June). Keyword extraction using backpropagation neural networks and rule extraction. In Neural Networks (IJCNN), The 2012 International Joint Conference on (pp. 1-7). IEEE.
[12] Jo, T., Lee, M., & Gatton, T. M. (2006, November). Keyword extraction from documents using a neural network model. In Hybrid Information Technology, 2006. ICHIT'06. International Conference on (Vol. 2, pp. 194-197). IEEE.
[13] Sangeetha, J., & Jothilakshmi, S. (2015, January). A novel spoken document retrieval system using Auto Associative Neural Network based keyword spotting. In Intelligent Systems and Control (ISCO), 2015 IEEE 9th International Conference on (pp. 1-6). IEEE
[14] Christian Wartena, Brusee, Slakhorst. 2010 “Keyword Extraction using Word Co-occurrence”, published in Database and Expert System Applications, p. 54 - 58.
[15] Sarkar, K., Nasipuri, M., & Ghose, S. (2010). A new approach to keyphrase extraction using neural networks. arXiv preprint arXiv: 1004.3274.
[16] C. Wartena, R. Brussee and W. Slakhorst, "Keyword Extraction Using Word Co-occurrence," 2010 Workshops on Database and Expert Systems Applications, Bilbao, 2010, pp54-58 doi 10.1109/DEXA.2010.3
[17] [16] V. Bhatia and V. Hasija, "Targeted advertising using behavioural data and social data mining," 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN), Vienna, 2016, pp. 937-942.
[18] doi: 10.1109/ICUFN.2016.7536934

BIOGRAPHY

Rasika Arun Londhe is currently a third year student of Computer Engineering in D. Y. Patil College of Engineering, Akurdi, Pune.
Ms. Vidya Nikam is currently a professor in Computer Engineering department in D. Y. Patil College of Engineering, Akurdi, Pune.