***I.J. Education and Management Engineering*, 2018, 1, 48-58**
Published Online January 2018 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijeme.2018.01.06

Available online at http://www.mecs-press.net/ijeme

Keyphrase Extraction of News Web Pages

Chandrakala Arya[a*], Sanjay k. Dwivedi[b]

[a] *Research Scholar, Department of Computer Science, B.B. Ambedkar University, Lucknow-226025,India*
[b]*Professor, Department of Computer Science, B. B. Ambedkar Unversity, Lucknow-226025, India*

Received: 28 Februayr 2017; Accepted: 11 September 2017; Published: 08 January 2018

Abstract

Keyphrase extraction from news web pages is an important task for news documents retrieval and summarization. Keyphrases are like index terms that enclose the important information about document content. Keyphrases actually offer concise and precise description of document content. Key phrases are considered as a single word or a combination of more than one word that represent the important concepts in a text documents. The aim of this paper is to develop and evaluate an automatic keyphrases extraction approach for news web pages. Our approach identifies the candidate keyphrases from documents and chooses those candidate keyphrase having highest weight score. Weight formula combines the feature set that includes TF*IDF, phrase disatnce in documents and lexical chain that is based on WordNet to represent semantic relations between words. The experimental results show that the performance of our approach is better than the contemporary approaches today.

Index Terms: Keyphrase extraction, Lexical chain, Web News, TF*IDF, WordNet.

1. Introduction

Under the growth of worldwide networking through the internet, the news consumption pattern moved from the traditional physical newspapers to online news aggregate system. As thousands of web news is posted on the internet every day, it is difficult to retrieve and summarize the relevant document effectively. So keyphrase extraction technique is used to provide the main contents of a given web page. It is useful in many areas like summarization, automatic indexing, topic search and clustering [7]. Keyphrase extraction is one of the most important tasks in news web pages. Readers make benefit from keyphrase because they can judge more quickly whether the news web page is worth reading.

* Corresponding author.
E-mail address: arya.chandrakala@gmail.com, skd200@yahoo.com

Keyphrases provide a concise description of document content. We treat a document as a set of phrases; any phrase in a new document can be extracted as a keyphrase. Keyphrase can be defined as a phrase of one or more words that denote the main concept of the document. Phraseness and informativeness are the two main features of keyphrase. Phraseness is a fairly dynamic idea which depicts the degree to which a given word sequence is considered to be a phrase. Informativeness denotes how well a phrase catches or outlines the important notions in a set of documents. A set of keyphrases related to a document gives high- level description of a document content that helps readers in searching for relevant information.

Keyphrase extraction in a news web page has been a challenging research topic in recent years because news changes very rapidly. Only a small number of news websites have author given keyphrases and manually allocating keyphrases for each web news document is very effortful. Thus it is absolutely necessary to automatically extract keyphrases. Automatic keyphrase extraction benefits users for the large document collection. Keyphrases of a document should be semantically related with the other words of the document. Therefore, in this paper, we proposed a Keyphrase Extraction approach, which uses lexical chain of semantically related words that are interconnected by semantic relations. The number of words and the number of semantic relations among the words can be different for each lexical chain. WordNet is used for the construction of lexical chain.

The organization of the paper is follows as. In section 2, previous studies on keyphrase extraction are discussed first. Section 3 describes the dataset used in the experiment. In section 4, we describe our proposed approach for the news web page keyphrase extraction. Experimental results and evaluation are discussed in section 5. Finally, some concluding remarks and future scope is discussed in section 6.

2. Related Work

In the previous works authors have suggested that document keyphrase can be useful in many areas as information retrieval and summarization. Chien [1] developed a keyphrase extraction system for Chinese and other Asian languages. Witten I. H. et al. [2] describe KEA algorithm, based on Naïve Bayes classifier automatically extracts keyphrases from text. This algorithm recognizes candidate keyphrases using lexical methods and computes feature values for each candidate by using machine learning algorithm and analyze which candidates are noble keyphrases. Martinez J. L. et al. [3] focus on AKE (Automatic Keyword Extraction), it is a keyword extraction system which is used to extract news articles keywords. KIP (Keyphrase identification program) [4] uses sample human keyphrsaes and then learns to identifiy additional news keyphrases. KIP mines noun phrases from documents and score will be allocated to each noun phrases. Depending on the weights the words that have higher score than the threshold will be selected as keyphrases. Wang J. et al. [5] proposed in their paper Neural Network based keyphrase extraction method. Lui Y. J. [11] presents a domain independent keyphrase extraction algorithm, which distinguish keyphrases from non – keyphrases by using statistical and computational linguistics techniques combination, a new attribute set and a new machine learning method; and shown that it perform well than other keyphrase extraction methods. Li z. F. et al. [12] proposed an approach based on lexical chain by using Reget's thesaurus and improve the KEA keyphrase extraction. Duwairi R. et al. [13] presents a framework for keyphrase extraction based on the KEA system. It relies on supervised learning particularly Naïve Bayes algorithm. Xu, S. et al. [14] introduce several novel word features by extracting inlink, outlink, category and infobox information from Wikipedia article set. Luo, Z. et al. [15] propose a method to integrate the comment posts for keyphrase extraction from web news documents. Boudin, F. [16] present and compare five centrality measures for graph based keyphrase extraction and used three datasets of different language and domain. Their results outperform the other centrality measure on short documents. Xie, F. et al. [17] proposes an approach which acquires semantic features within phrases from a single document. Their result demonstrates better performance than TFIDF and KEA. Gao, Y. et al. [18] propose a method to extract hot keyphrases from news report; their method consists a two-step process of keyphrase extraction based on TF*PDF. In their method each step uses position- weighted TF*PDF schema. Li, Z. et al. [19] propose a method based on the lexical chain to improve KEA keyphrase extraction, their experiments result shows improvements compare with KEA and Nguyen and Kan's method. Hsu, H. M. et al. [20] propose subject- keyphrase concept to extract subject-keyphrases from a documents. They use definition-use chain for subject-keyphrase extraction algorithm. Wang, C. et al [21] propose a system for automatic online news topic keyphrase extraction, their system perform effectively with 70.61% precision and 67.94% recalls.

3. Description of The Dataset

The online news articles have been chosen from the 'The Hindu' news website. All these selected news is world news posted from 20 April 2016 to 30 April 2016. Our dataset contains 150 web news documents. The key purpose, we select 'The Hindu' news website for the experiment is that every news web page has author assigned keywords. We take the author assigned keywords as gold standard keyphrase. We choose some keyphrase manually for each document. Most of the keyphrases consists of one or more than one words.

Keyphrases having more than three words are less in number in our dataset. Average number of manually assigned keyphrases per document is 15. Here it is interesting to note that all author allotted keyphrase for a document may not occur in the title of the document. Total number of noun phrases in our dataset is 2250. The total number of author assigned keyphrases for all the documents in our dataset is 479.

4. Proposed Method

In the proposed method firstly in the document words are segmented, stemmed and stop words are removed. After that candidate phrases from the document are identified. Weight of each candidate phrase is computed by the features TF*IDF, phrase distance, and building lexical chain. According to the weight, a high scorer candidate phrases is selected as a keyphrases. The process of keyphrase extraction is shown in Fig1.

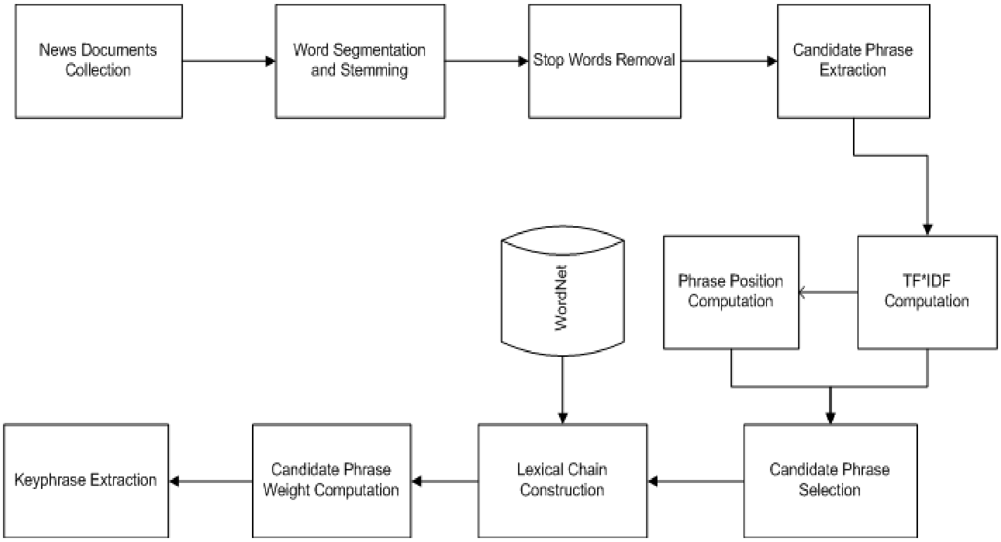

News Documents Collection
Word Segmentation and Stemming
Stop Words Removal
Candidate Phrase Extraction
WordNet
Phrase Position Computation
TF*IDF Computation
Keyphrase Extraction
Candidate Phrase Weight Computation
Lexical Chain Construction
Candidate Phrase Selection

Fig.1. Keyphrase Extraction Process

The steps of the proposed method are as follows:

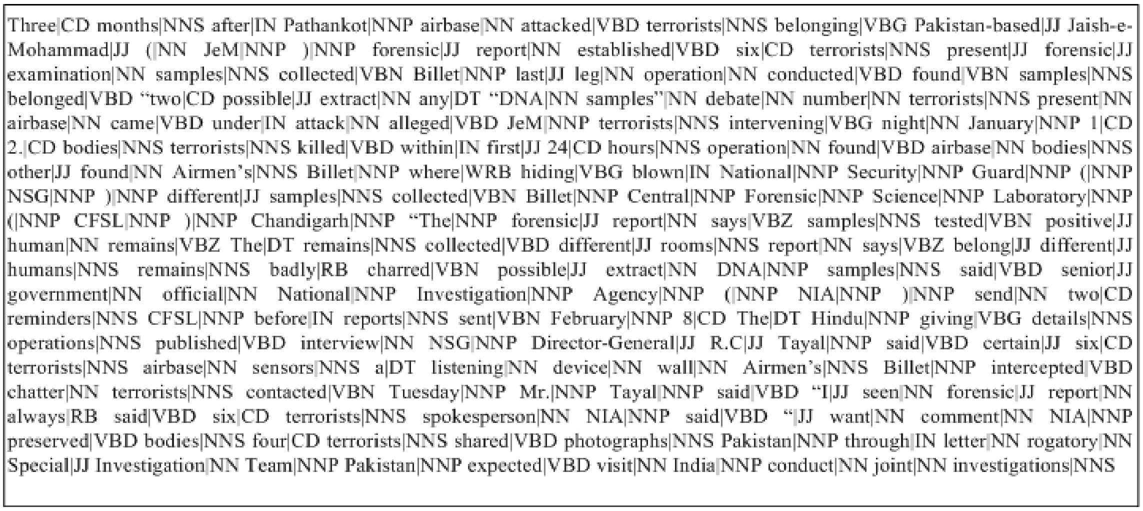
1. Words are segmented and stemmed and stop words are removed.
2. Identify the candidate phrase from each document.
3. Compute the TF*IDF and phrase distance of each candidate word

4. Select the top n candidate phrase according to the value of TF*IDF and phrase distance.
5. Build the lexical chains of each top n candidate phrase.
6. Compute the weight of each candidate phrase.
7. Select the top m candidate words as the keyphrase according to their weights. Select those candidate words as keyphrases which have higher weights.

4.1. Identification of Candidate Phrase

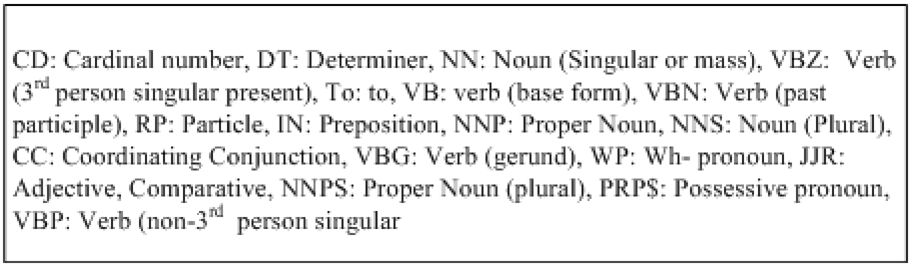
Keyphrases are extracted from candidate phrases. The noun phrases in the document are treated as the candidate keyphrase [6]. In order to recognize the noun phrases documents have been tagged by stanford Part-Of- Speech (POS) tagger [27]. We used Stanford POS tagger to extract the lexical information about the terms in a document. Fig. 2 shows the lexical tag assigned by the tagger for a document. According to this figure, JJ, DT, NN, NNS, VBZ, NNP, PRP$, VBN, IN, CD, etc are lexical tags assigned by the POS tagger.

Fig 3 shows the meaning of these tags. Candidate keyphrase extracted from Fig 2. Are: terrorists, central forensic science laboratory, DNA sample, government officials, National Investigation agency, investigation team, spokesperson, photographs, sensors.



Three|CD months|NNS after|IN Pathankot|NNP airbase|NN attacked|VBD terrorists|NNS belonging|VBG Pakistan-based|JJ Jaish-e-Mohammad|JJ (|NN JeM|NNP)|NNP forensic|JJ report|NN established|VBD six|CD terrorists|NNS present|JJ forensic|JJ examination|NN samples|NNS collected|VBN Billet|NNP last|JJ leg|NN operation|NN conducted|VBD found|VBN samples|NNS belonged|VBD "two|CD possible|JJ extract|NN any|DT "DNA|NN samples"|NN debate|NN number|NN terrorists|NNS present|NN airbase|NN came|VBD under|IN attack|NN alleged|VBD JeM|NNP terrorists|NNS intervening|VBG night|NN January|NNP 1|CD 2.|CD bodies|NNS terrorists|NNS killed|VBD within|IN first|JJ 24|CD hours|NNS operation|NN found|VBD airbase|NN bodies|NNS other|JJ found|NN Airmen's|NNS Billet|NNP where|WRB hiding|VBG blown|IN National|NNP Security|NNP Guard|NNP (|NNP NSG|NNP)|NNP different|JJ samples|NNS collected|VBN Billet|NNP Central|NNP Forensic|NNP Science|NNP Laboratory|NNP (|NNP CFSL|NNP)|NNP Chandigarh|NNP "The|NNP forensic|JJ report|NN says|VBZ samples|NNS tested|VBN positive|JJ human|NN remains|VBZ The|DT remains|NNS collected|VBD different|JJ rooms|NNS report|NN says|VBZ belong|JJ different|JJ humans|NNS remains|NNS badly|RB charred|VBN possible|JJ extract|NN DNA|NNP samples|NNS said|VBD senior|JJ government|NN official|NN National|NNP Investigation|NNP Agency|NNP (|NNP NIA|NNP)|NNP send|NN two|CD reminders|NNS CFSL|NNP before|IN reports|NNS sent|VBN February|NNP 8|CD The|DT Hindu|NNP giving|VBG details|NNS operations|NNS published|VBD interview|NN NSG|NNP Director-General|JJ R.C|JJ Tayal|NNP said|VBD certain|JJ six|CD terrorists|NNS airbase|NN sensors|NNS a|DT listening|NN device|NN wall|NN Airmen's|NNS Billet|NNP intercepted|VBD chatter|NN terrorists|NNS contacted|VBN Tuesday|NNP Mr.|NNP Tayal|NNP said|VBD "I|JJ seen|NN forensic|JJ report|NN always|RB said|VBD six|CD terrorists|NNS spokesperson|NN NIA|NNP said|VBD "|JJ want|NN comment|NN NIA|NNP preserved|VBD bodies|NNS four|CD terrorists|NNS shared|VBD photographs|NNS Pakistan|NNP through|IN letter|NN rogatory|NN Special|JJ Investigation|NN Team|NNP Pakistan|NNP expected|VBD visit|NN India|NNP conduct|NN joint|NN investigations|NNS

Fig.2. POS Tagged Document



CD: Cardinal number, DT: Determiner, NN: Noun (Singular or mass), VBZ: Verb (3rd person singular present), To: to, VB: verb (base form), VBN: Verb (past participle), RP: Particle, IN: Preposition, NNP: Proper Noun, NNS: Noun (Plural), CC: Coordinating Conjunction, VBG: Verb (gerund), WP: Wh- pronoun, JJR: Adjective, Comparative, NNPS: Proper Noun (plural), PRP$: Possessive pronoun, VBP: Verb (non-3rd person singular

Fig.3. Meanings of the Tags

4.2. TF*IDF of Candidate Phrase

After identifying candidate phrase, the collection of candidate phrases identified in the web news documents may be huge in number. From a vast collection a small number of phrases may be selected as the keyphrases. In this paper we select 15 keyphrases from a single document. TF*IDF of each candidate phrase is used to rank the phrases. TF*IDF measure the phrase frequency in a document compared to its rarity in general use.

We compute the TF*IDF of each word by the given eq. (1)

$$TF * IDF = \frac{t_f}{t_n} * \log(\frac{N}{n_i}) \tag{1}$$

Where tf is the frequency of term t in a document, tn is the total number of terms in a documents, N is the total number of documents and ni is the number of documents in the dataset that contains term t.

4.3. Phrase Distance

The distance attribute is the position where a phrase first appears in the document. The candidate keyphrases that appears early in a document should be given higher score. Like previous approach [7], Distance of a phrase from the start of a document is measured as the number of words that precede its first appears divided by the number of words in the documents. The distance of a phrase in the document is calculated as in eq. (2)

$$PhraseDis\tan ce = \frac{n_j}{n} \tag{2}$$

Where nj is the number of words that predate its first appearance, and number of words in the document are denoted by n.

4.4. Construction of Lexical Chain

Firstly Morris and Hirst [8] give the concept of lexical chain. According to them lexical cohesion is an arrangement of related words that give the continuity of lexical meaning. Lexical cohesion occurs as a result of semantic relation between words. One of the main advantages of lexical cohesion is that it is an easily recognizable relation that enables the computation of lexical chain. Lexical chains visualize the semantically related words or phrases in the text. These words or phrases are called the lexical items and each item gives a specific meaning to a lexical chain. In this paper we use WordNet for creating lexical chains. With the help of path between concepts, lexical chain can be found. In general two concepts can have many possible lexical chains. For creating lexical chains we ignore numbers, units, currencies, times/periods, names, places and referring items [10]. Suo, Hong-guang et al. [22] use HowNet to determine the relationship between words and build vocabulary chain. For the construction of lexical chain we used synonym, hypernym/hyponym, coordinate term and meronym, Silber, H. Gregory [23] and Ercan, Gonenc [9] also used the same relations except coordinate term. In order to rank lexical chains, high scoring chains must be picked as the important concept from the original document. We use Barzilay and Elhadad [24] idea of strong chain.

Fig 4 shows the different set of lexical chains chooses from the tagged document in Fig 2.

Lexical chains usually depend on semantic relations that can be acquired from WordNet. Hypernym/ Hyponym, Synonym/ Repetition, Meronym/ Holonym, Antonym, and Sibling relations are used to build lexical chains.

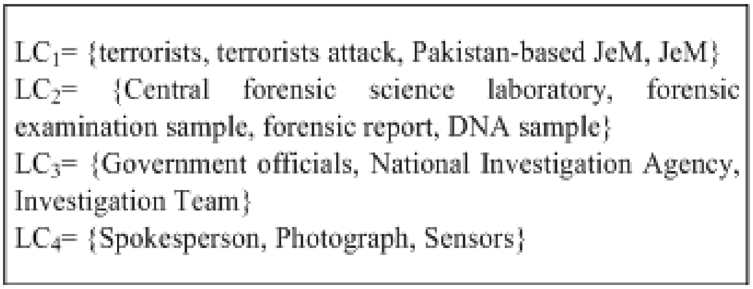
Fig 5 shows the lexical graph of LC1 in detail.

Weights of every relation between word senses are given allegedly [9]. Table 1, shows the allocated weights for the relation. Subsequent to scoring each lexical chain of the word, we select the chain with a maximum score as the lexical chain.

Table 1. Weight of Lexical Chain Relation

Relation	Explanation of Relation	Weight
Synonym/ reiteration	Same meaning	10
Coordinate Term	Sibling	8
Hypernym/hyponym	General/specific	7
Meronym	Is a part of	4

According to these assigned weights, the score of lexical chain LC1 is equal to 43 (=5*7 + 8) since there are five Hypernym/Hyponym relations and one Coordinate term.



LC$_1$= {terrorists, terrorists attack, Pakistan-based JeM, JeM}

LC$_2$= {Central forensic science laboratory, forensic examination sample, forensic report, DNA sample}

LC$_3$= {Government officials, National Investigation Agency, Investigation Team}

LC$_4$= {Spokesperson, Photograph, Sensors}

Fig.4. Set of Lexical Chain

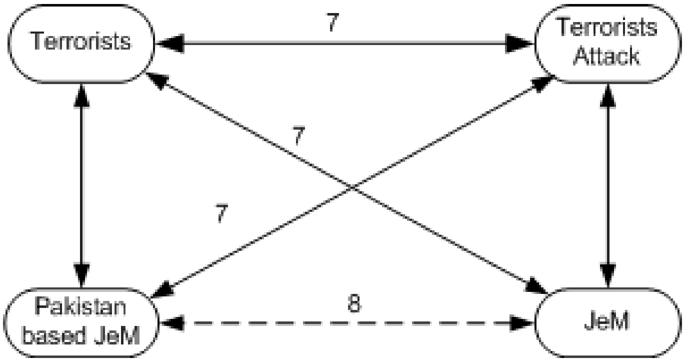


Fig.5. Lexical graph

4.5. Weight of Candidate Phrase

Weight of a candidate phrase can be obtained by the combination of the features: TFIDF, phrase distance, and lexical chain shown in eq. (3).

$$\text{weight} = a * TF * IDF + b * \text{phrasedistance} + c * \text{Lexical chain} \tag{3}$$

Where TF*IDF is the value of the candidate phrase, phrase distance is the distance into the document of candidate phrase first appearance, and lexical chain is the length of the chain that contains candidate phrase and a, b, c are the parameters that can be adjusted. The value of these parameters in our experiment has been set to 1.

Suppose we have to find the weight of a candidate phrase “Terrorist”, which we selected from our dataset. The weight computation as discussed in eq. (3) comprise of three components, the value of each component has been obtained as follows.

Firstly we calculate the TF*IDF of the phrase “Terrorist”. The position of “Terrorist” phrase in the document of our dataset is 9 and the number of words in the whole document is 367, then the value of TF*IDF is calculated as in eq. (1)

$$TF * IDF = \frac{t_f}{t_n} * \log(\frac{N}{n_i})$$

TF*IDF= 0.0174832

Secondly we find the value of Phrase distance of the “Terrorist” phrase as calculated in eq. (2)

$$PhraseDis\tan ce = \frac{n_j}{n}$$

Where the value of nj is 6 in the document and n is 367. Therefore the value of phrase distance is

Phrase distance = 0.0163

In the next step, we construct and then calculate the value of lexical chain. We choose lexical chain LC1 from fig. 4, because the value of LC1 is higher than other lexical chains. The value of lexical chain LC1 is 43, calculated in previous section 4.4. Finally the weight of the candidate phrase is calculated as:

W= 1*0.0174832+ 1* 0.245 + 1* 43
W= 43.0174832 ~ 43.01

Like the “terrorist” phrase, all the candidate phrases of the dataset are calculated. We select the top fifteen higher weight scorer candidate phrases as keyphrases of a document.

5. Experiment Result and Evaluation

Experiments were carried out to evaluate the overall performance of our approach. For evaluating the automatically generated keyphrases, we first take the two standard information retrieval metrics precision and recall. The precision; measures the proportion of number of extracted key phrases that are also author tagged key phrases to the total number of extracted keyphrases. The second one ‘recall’ measures the proportion of the extracted key phrases that are also author tagged key phrases to the number of author tagged keyphrases. These metrics show how well generated phrases match a set of relevant phrases.

$$\Pr ecision = \frac{N_{e \cap t}}{N_e} \tag{4}$$

$$\text{Re}\,call = \frac{N_{e \cap t}}{N_t} \tag{5}$$

Where Ne is the number of keyphrases extracted, Nt the number of keyphrases tagged by author. Ne∩t is the number of extracted keyphrases that are also keyphrases tagged by author.

Table 2 shows the keyphrases assigned by the author of the news article which is the document number 2 in our dataset.

Table 2. Author Assigned Keyphrases for News Article Number 2 in the Dataset

Document No.	Author Key
2	Pathankot attack
2	forensic attack
2	Terrorism
2	Special Investigation Team
2	Joint investigation

From the document 2, our proposed approach extracted the top 5 keyphrases as shown in Table3.

Table 3. Top 5 Keyphrases Extracted By our Proposed Approach

Document No.	Author Key
2	Pathankot attack
2	forensic science laboratory
2	terrorism
2	Special Investigation Team
2	National Investigation agency

Table 2 and Table 3 show that out of 5 keyphrases extracted by our approach, 3 keyphrases matched with the author assigned keyphrases.

In order to compare our approach with state- of- the-art keyphrase extraction systems we have selected KEA [2] and KESR [25]. Most existing systems identify candidate phrases by the method applied in KEA and KESR.

KEA is comparatively simple and useful in automatic keyphrase extraction. The KEA identifies candidate keyphrase using lexical methods and calculates the feature value of each candidate phrase, and then predicts the good keyphrase from candidate by using machine learning algorithm. The basic model of KEA involves two stages. Firstly build a model for recognizing keyphrases by using training documents where the author keyphrases are known. Secondly, use the model create on first stage, choose the keyphrases from a new document. The overall performance of KEA show that on average KEA can match between one and two of the five keyphrases chosen by the average author in the collection.

NFAS system considers all phrases except stop words in the web news pages. In this system Key-phrase Extraction based on Semantic Relations (KESR) algorithm is used for keyphrase extraction. The goal of KESR is to extract those words that have a low frequency but provide a major impact to the text subject. The basic model of KESR algorithm involves two attributes: TFIDF, and word similarity and lexical chain. Word similarity is computed through HowNet. Extracted keyphrases compared with the phrases in the news title and phrases in the core hints provided by the author. By comparing their results with TF*IDF and KELC (Key-phrase extraction based on lexical chains) [22], KESR outperforms the other two in both the cases, when the title kept and when the title removed and core hints kept.

In this paper, we compare the overall performance of our keyphrase extraction method with the existing keyphrase extraction methods. In the experiment, the number of keyphrases to be extracted was set to 5, 10, and 15 respectively. Table 4, shows that the approach presented in this paper seems to be better than other approaches in terms of precision and recall.

Table 4. Precision and Recall Comparison of Three Approaces

Number of Keyphrases	Average Precision			Average Recall		
	Our Approach	KESR	KEA	Our Approach	KESR	KEA
5	0.34	0.32	0.28	0.25	0.24	0.29
10	0.22	0.20	0.19	0.46	0.36	0.40
15	0.17	0.18	0.15	0.51	0.41	0.48

Fig 6 shows the comparison of the individual performance of three different approaches. Precision is the proportion of the keyphrases extracted that are correct. The experiments indicates that the precision of our approach when extracting 5, 10, and 15 keyphrases is 0.34, 0.22 and 0.17 respectively is greater than KEA and KESR for the same 5, 10 and 15 keyphrases 0.28, 0.19 and 0.15; and 0.32, 0.20, and 0.18 respectively.

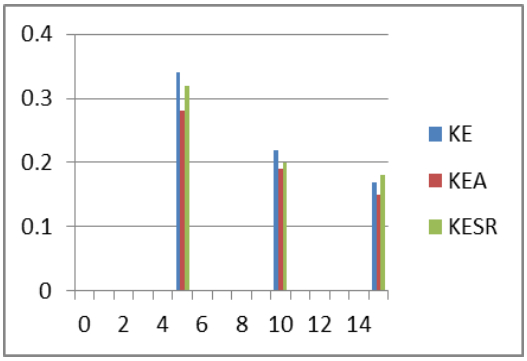


Fig.6. Precision Comparison of Three Algorithms.

Fig 7 shows the recall comparison of three different approaches. Recall is the fraction of relevant instances that are retrieved. Recall of our approach when extracting 5, 10, and 15 keyphrases is 0.25, 0.46 and 0.51 respectively is greater than KEA and KESR for the same 5, 10 and 15 keyphrases 0.29, 0.40 and 0.48; and 0.24, 0.36, and 0.41 respectively.

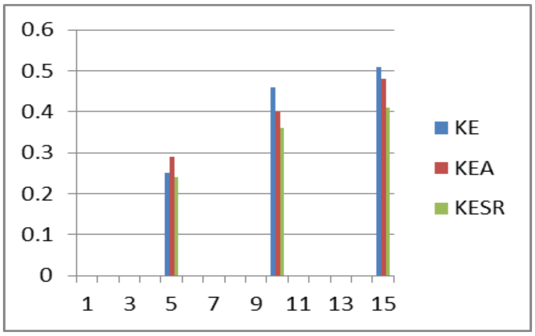


Fig.7. Recall Comparison of Three Algorithms

6. Conclusion and Future Work

In this paper, we presented an effective technique which can extract keyphrases from news web page. In this work, we take noun phrases of the documents as a candidate Phrase, and used POS tagger for this task. While ranking candidate keyphrases, weights of each candidate keyphrase is measured and choose the highest score keyphrases. To determine the weight of the keyphrase we use the TFIDF, phrase distance in the document and lexical chain. The approach is evaluated by the evaluation parameters precision and recall. Experimental results show that this approach is competitive with other known approaches.

In the future, we might want to use more datasets to assess our system. For keyphrase extraction algorithm there is no standard datasets are available. In this paper we compare the extracted keyphrase with the author assigned keyphrases. But there are many other issues in this method. First, author assigned keyphrases are not generally show up in the document to which they belong. So, if a keyphrase is not contained in a given web page, it is never extracted as a keyphrase of the given web page by the automatic keyphrase extraction algorithm. Second authors provide only limited keyphrases which are less than extracted automatic. Therefore we will accomplice more research on searching for a more logical and target approach to assess the automatic extraction results.

References

[1] Chien LF. PAT-tree-based adaptive keyphrase extraction for intelligent Chinese information retrieval. Inf. Process. Manage.1999 Jul 1; 35(4):501-21.

[2] Witten IH, Paynter GW, Frank E, Gutwin C, Nevill-Manning CG. KEA: Practical automatic keyphrase extraction. InProceedings of the fourth ACM conference on Digital libraries 1999 Aug 1; 254-255.

[3] Mart ínez-Fern ández JL, Garc ía-Serrano A, Mart ínez P, Villena J. Automatic keyword extraction for news finder. InInternational Workshop on Adaptive Multimedia Retrieval 2003 Sep 15; 99-119.

[4] Wu YF, Li Q, Bot RS, Chen X. KIP: a keyphrase identification program with learning functions. In Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on 2004 Apr 5; 2: 450-454.

[5] Wang J, Peng H, Hu JS. Automatic keyphrases extraction from document using neural network. Advances in Machine Learning and Cybernetics. 2006; 633-41.

[6] Wu YF, Li Q. Document keyphrases as subject metadata: incorporating document key concepts in search results. Information Retrieval. 2008 Jun 1; 11(3):229-49.

[7] Frank E, Paynter GW, Witten IH, Gutwin C, Nevill-Manning CG. Domain-specific keyphrase extraction. In16th International Joint Conference on Artificial Intelligence (IJCAI 99) 1999; 2: 668-673).

[8] Morris J, Hirst G. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. Computational linguistics. 1991 Mar 1; 17(1):21-48.

[9] Ercan G, Cicekli I. Using lexical chains for keyword extraction. Information Processing & Management. 2007 Nov 30; 43(6):1705-14.

[10] Steffen R. Lexical chain Annotation Guidelines. 2012.

[11] Lui YJ, Brent R, Calinescu A. Extracting significant phrases from text. In Advanced Information Networking and Applications Workshops, 2007, AINAW'07. 21st International Conference on 2007 May 21; 1: 361-366.

[12] Li ZF, Zhao XH, Yi J, He B. Improvement of KEA Based on Lexical Chain. In Advanced Materials Research 2013; 756: 2999-3004.

[13] Duwairi R, Hedaya M. Automatic keyphrase extraction for Arabic news documents based on KEA system. Journal of Intelligent & Fuzzy Systems. 2016 Jan 1; 30(4):2101-10.

[14] Xu S, Yang S, Lau FC. Keyword Extraction and Headline Generation Using Novel Word Features. In AAAI 2010 Jul 5; 1461-1466.

[15] Luo Z, Tang J, Wang T. Improving keyphrase extraction from web news by exploiting comments information. InAsia-Pacific Web Conference 2013; 140-150.

[16] Boudin F. A comparison of centrality measures for graph-based keyphrase extraction. In International Joint Conference on Natural Language Processing (IJCNLP) 2013; 834-838.

[17] Xie F, Wu X, Hu X. Keyphrase extraction based on semantic relatedness. In Cognitive Informatics (ICCI), 2010 9th IEEE International Conference on 2010; 308-312

[18] Gao Y, Liu J, Ma P. The hot keyphrase extraction based on tf* pdf. InTrust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on 2011 Nov 16; 1524-1528

[19] Li Z, He B. Adding Lexical Chain to Keyphrase Extraction. InWeb Information System and Application Conference (WISA), 2014 11th 2014; 254-257.

[20] Hsu HM, Chang RI, Chang YJ, Lin SY, Wang YJ, Ho JM. Subject-Keyphrase Extraction Based on Definition-Use Chain. In Web Intelligence and Intelligent Agent Technology (WI-IAT), 2015 IEEE/WIC/ACM International Conference on 2015 Dec 6; 3: 199-202.

[21] Wang C, Zhang M, Ru L, Ma S. An automatic online news topic keyphrase extraction system. In Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01 2008 Dec 9; 214-219.

[22] Suo H, Liu Y, Cao S. A keyword selection method based on lexical chains. Journal of Chinese Information Processing. 2006; 20(6):25-30.

[23] Silber HG, McCoy KF. Efficiently computed lexical chains as an intermediate representation for automatic text summarization. Computational Linguistics. 2002 Dec; 28(4):487-96.

[24] Barzilay R. *Lexical chains for summarization* (Doctoral dissertation, Ben-Gurion University of the Negev). 1997

[25] Wu X, Wu GQ, Xie F, Zhu Z, Hu XG. News filtering and summarization on the web. IEEE Intelligent Systems. 2010 Sep; 25(5):68-76.

[26] http://nlp.stanford.edu/software/tagger.shtml

Authors' Profiles

Chandrakala Arya is Research Scholar at Department of Computer Science in B.B. Ambedkar University, Lucknow, India. She has received her M.C.A. Degree in the year 2011 from Uttarakhand Technical University. Her research interest includes Information Extraction and Text Summarization. She has published some of the research papers in international conferences.

Prof. Sanjay K. Dwivedi is Professor and Head at Department of Computer Science in B.B. Ambedkar University, Lucknow, India. He has received his Ph.D. Degree from Banasthali Vidyapeeth in area of Web Mining in the year 2006. His research interest includes Web content Mining, Semantic Web, Search Engine performance evaluation, Machine translation, Information Retrieval etc. He has published many of the valuable research papers in various national and international Journals of repute.

How to cite this paper: Chandrakala Arya, Sanjay k. Dwivedi,"Keyphrase Extraction of News Web Pages", International Journal of Education and Management Engineering(IJEME), Vol.8, No.1, pp.48-58, 2018.DOI: 10.5815/ijeme.2018.01.06