Computer Science and Information Technology 4(1): 33-43, 2016 http://www.hrpub.org
DOI: 10.13189/csit.2016.040106

Performance Evaluation of an Improved Model for Keyphrase Extraction in Documents

Awoyelu I.O.*, Abimbola R.O., Olaniran A.T., Amoo A.O, Mabude C.N.

Department of Computer Science and Engineering, Obafemi Awolowo University, Nigeria

Abstract Keyphrases are one of the most important parts of a document that give an insight on how a specific document is related. Keyphrase extraction systems are becoming increasingly vital in extracting quality keyphrases. They extract quality phrases that describe a document at hand. Existing keyphrase extraction systems, that employed the unsupervised approach, extract non-domain-specific keyphrases, thereby producing generic keyphrases. An improved model for domain-specific keyphrase extraction in journal articles is therefore proposed in this study. It is a framework that employs document structure, term frequency and inverse document frequency, noun phrase identifier and domain knowledge for keyphrase extraction. Data used in this research include nouns and stop words in English Language. Author-assigned keyphrases were extracted from the *International Journal of Data Mining and Knowledge Processing (IJDKP)* between the year 2011 and year 2014 for building the domain knowledge and testing the system. It was implemented using Java programming language and MySQL query language. Evaluation was carried out using precision, recall and f-measure as performance metrics. The results obtained show that the proposed system yielded an average precision, recall and f-measure of 27%, 53% and 35% respectively compared to the existing model – MAUI - which yielded average precision, recall and f-measure of 23%, 45% and 35% respectively. This shows that the proposed model outperformed the existing model by 5%.

Keywords Keyphrase, Document Extraction, Phrase Frequency, Noun Tagger, Journal Articles

1. Introduction

Phrases are two or more words that do not contain the subject-verb pair necessary to form a clause [1]. They can be very short or quite long. A phrase does not necessary contain a subject and verb. For example, considering a sentence, "he is laughing at the joker"; "At the joker" is a phrase. Certain phrases have specific names based on the type of word that begins or governs the word group such as noun phrase, verb phrase, prepositional phrase, adverbial phrase, adjectival phrase, infinitive phrase, participle phrase and gerund phrase. Some of these phrases (especially the noun phrases) found in documents are concatenated to form a keyphrase.

Keyphrases can be defined as a shortlist of phrases (typically five to fifteen noun phrases) that captures the main topics discussed in a given document [2]. These keyphrases are extracted from the documents like keywords because they play important roles in identifying the context of documents or articles. However, keyphrases and keywords extraction are distinct in the fact that extracted keyphrases contain one or more strings of words while extracted keywords contain only a word. They both represent the documents at hand but keyphrases produce more comprehensive document representation. Keyphrase extraction is an aspect of text mining that deals with the selection of important phrases in a document that relate to what the reader of such document has in mind. Keyphrases in documents help in providing important information on what constitutes a document at hand.

1.1. Keyphrase Extraction Methods

Most of the keyphrase extraction methods have been based on the supervised machine learning approach i.e. the use of training data, machine learning schemes like naives Bayes, conditional random field and support vector machine. One of the main advantages of a supervised machine learning approach is that it gives more accurate outputs when the documents to be extracted are in the same context as the training data. For example, a document on "engine performance" would give accurate keyphrases when tested with a document on "car mechanics". One of the drawbacks of the supervised machine learning approach is that it performs less accurately when it is not trained within the context of the documents; for example, using the trained documents as "crop production" while the document for extraction is on "databases". The unsupervised approach

does not support training of data; the approach involves the use of statistics, linguistic and heuristic approaches. One of the main advantages of this approach is that it gives better output irrespective of the document context i.e. it doesn't make use of training document rather it makes use of statistical or linguistic features such as the term frequency and inverse document frequency (TF*IDF), lexical analysis, syntactic analysis, word frequency. [3] Used the TF*IDF to formulate a weighting scheme for medical documents. Other examples of statistical methods include N-gram, word frequency, word co-occurrences and PAT-tree (Patricia Tree - a suffix tree or position tree) [4]. [5] Used a statistical approach to generate keywords for headline. The disadvantage of this approach is that in some professional texts, such as Computer Science, the most important keyword may appear just once in the journal article. The use of statistical models may carelessly discard important phrases. As manual extraction of keyphrases is a stressful task, several approaches to keyphrase extraction have been proposed. Only a few of them are freely available, which makes it difficult for researchers to replicate previous results or use keyphrase in some other application such as information retrieval or question answering systems [6]. Most of the researches, till date, have employed the supervised machine learning approach within the fields of term categorization and text mining.

1.2. Related Works

Existing works on keyphrase extraction have been based mostly on supervised approaches. Reference [7] tried to improve on the work of [8], which is concerned with automatic extraction of keywords from abstracts using a supervised machine learning algorithm - Naive Bayes. The main focus of this work is that by adding linguistic knowledge to the representation (such as syntactic features), rather than relying only on statistics (such as term frequency and n-grams), a better result is obtained when compared with keywords previously assigned by professional indexers. In other words, extracting noun phrase chunks gives a better precision than *n*-grams, and by adding the part of speech tags assigned to the term as a feature, an improvement is made independent of the term selection approach. One of the weakness of this work is that there is currently no relation between the different parts of speech tag feature values. For example, a singular noun has no closer relationship with a plural noun than to an adjective.

Reference [9] explored several unsupervised approaches to automatic keyword extraction using meeting transcripts in the term frequency and inverse document frequency weighting framework. This work incorporated part of speech tagging, word clustering and sentence salience score. The work also evaluated a graph-based approach that measures the importance of a word based on its connections with other sentences or words. The system performance was evaluated in different ways, including comparison to human annotated keywords using *F*-measure and a weighted score relative to the oracle system performance, as well as novel alternative human evaluation. The results showed that the unsupervised TF*IDF approach performs reasonably well and the additional information from the part of speech and sentence score improved the keyword extraction. However, the graph method was less effective for the study. The results of the graph method could have been improved if an investigation was made to different weighting algorithms. A better way was needed to decide the number of keywords to generate instead of using a fixed number. Furthermore, since there are multiple speakers in the meeting domain, there should have been a way of incorporating the information of the speakers in various approaches. Reference [10] presented an approach to multi-document summarization using automatic keyphrase extraction. The approach proposed in this work has two parts. The first part is the automatic extraction of keyphrases from a certain document and the second part is the automatic generation of a multi document summary based on the extracted keyphrases. The supervised approach employed in this work made use of the conditional random field algorithm.

Reference [4] presented a supervised approach on extracting keywords using conditional random field algorithm from Chinese document. The study postulated that large portion of documents do not have keywords assigned while manual assignment of high quality keywords is expensive, time consuming and error prone. The experimental results show that the conditional random field model outperforms other machine learning methods such as support vector machine, multiple linear regression model in the task of keywords extraction. The conditional random field in this work took full advantage of all the features of the document. The system precision and recall would have improved if the study employed the use of semantic relations between keywords. Reference [11] presented a detailed supervised approach of automatic extraction of keyphrases from scientific articles using Conditional random fields (CRF). The system was trained using 144 scientific articles and tested on 100 scientific articles. The work opined that a combination of eleven features would enhance the extraction of quality phrases. The structure of a document was taken into consideration in extracting a keyphrase. The system was evaluated with precision, recall, f-measure and benchmarked with combined keywords of both author-assigned and reader-assigned keywords. The unsupervised approach would have extracted more quality keyphrases if there was a proper cleaning of the input documents and identification of more appropriate features. Reference [12] compared several statistic-based language independent methodologies to automatically extract keywords. The work rank words, multi-words and word prefixes (with fixed length: 5 characters). The system was tested on Portuguese, English and Czech languages. The result was evaluated using precision.

Reference [13] presented a neural network based approach

to keyphrase extraction from scientific articles. The study carried out two experiments, the first one was to check the quality of the system while the other experiment was a comparison with KEA [8] by using precision and recall as performance metrics. The work made use of five features such as TF*IDF, position of a phrase's first appearance, phrase length, word length in a phrase and the links of a phrase to other phrases. The results presented showed that the methodology performs better than some of the state of the art keyphrase extraction approaches. The system would produce a better result if the structure of the system is taken into consideration. GenEx (GenitorExtractor) is a system designed by [2]. It contains two components: the genitor algorithm and the extractor algorithm. The extractor takes a document as input and produces a list of keyphrases. It makes use of 12 parameters and the genitor algorithm is used to adjust the parameters of the extractor algorithm. The limitation of the system is that the training documents took a longer time. A keyphrase extraction system was designed by [8]. It was a machine learning scheme that made use Naive Bayes learning model that trains document faster than GenEx. It considers all document phrases as potential keyphrases, with the aim to differentiate between keyphrases and non-keyphrases. In Kea, selecting Keyphrases is carried out almost the same way as in GenEx, except that stemming was done with the iterated Lovins stemmer rather than truncation. Kea also allowed stop words to be in between candidate phrases, but not at the beginning or the end of such phrase. They also made use of three candidate phrases as used by GenEx. They made use of two features in considering a keyphrase which are the TF*IDF and the distance into the document of the phrase first appearance. When the boosting factor has been calculated, then the individual weight of each phrase or term was calculated. Reference [14] proposed a *Maui* system that improved on KEA [8]. The *Maui* framework employed the use of semantic knowledge retrieved from Wikipedia. A classification model was built to differentiate between a phrase and non-phrase. The idea behind *Maui* was basically automatic tagging on the web and keyphrase extraction employing statistical methods. The study claimed that the success of keyphrase extraction system is dependent on the quality of features used. The features used include TF*IDF, position of first occurrence, keyphraseness, phrase length, node degree, wikipedia-based keyphraseness, spread, semantic relatedness and inverse wikipedia linkage. One of the major challenges of *Maui* is that too much priority was given to the information got from the documents, for example, the journal publisher name took about 50% of the extracted keyphrases. The KPminer (keyphrase miner) system was built to extract keyphrases from documents. It was built for effectively extracting keyphrase automatically in English and Arabic [15]. The system made use of the state of the art TF*IDF for calculating the weight of individual phrases. It is noted that the frequency of phrases were less than that of single terms in documents. The idea was to introduce a boosting factor to normalize this issue.

2. Materials and Methods

This paper addresses an automatic assignment of keyphrases to Computer Science documents, specifically, articles in the area of Data Mining. The proposed method is an unsupervised approach. The method does not use any set of training data but has a stored set of important keyphrases in Computer Science field. The work identifies attributes with keyphrases, make some analysis on them, use different methods to calculate their weights and compare the results with the domain. The proposed keyphrase extraction model is depicted in Figure 1. The proposed model is an improved unsupervised keyphrase extraction model, where the noun phrase tagger is combined with the domain knowledge to extract quality keyphrases. The combination of the two units helps to improve the resultant keyphrases.

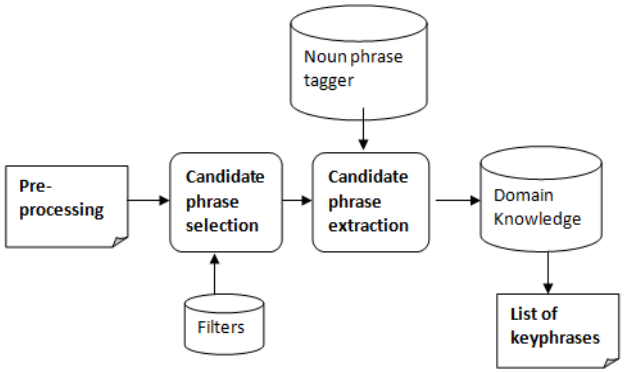

Figure 1. Proposed Model

In the pre-processing phase, as shown in Figure 1, the file to be used is converted to text files (*.txt*) or portable document format (*.pdf*) files because the system accepts text files or portable document format files as input. The document is analyzed by counting the total number of words in it. The words are divided into three sections. The first section contains the introduction which constitutes 32.4 percent of the total words for this study. The second section contains the body of the document which constitutes 45.3 percent. The third part contains the concluding part of the document which constitutes the remaining 22.3 percent of the document for this study. These sections are depicted in Table 1. In the table, 20 journals from the *International Journal of Data Mining and Knowledge Management Process (IJDKP)* for the year 2011 and 2012 were used to determine the structure of a journal article. The journals used were structured into introduction, related work, proposed model, experiments, results and analysis and conclusion. This study assumes that the introduction and related work constitute the introduction section. The proposed model and experiments constitute the body section. The results and analysis with the conclusion constitute the concluding section.

Table 1. Section Information for the Journal Article Structure

JOURNAL (words)	INTRODUCTION (words)	BODY (words)	CONCLUSION (words)
A(4248)	1210	1832	1206
B(3327)	1051	1102	1174
C(2727)	1106	585	1036
D(4600)	1194	2216	1190
E(6652)	1925	4003	724
F(4379)	1769	1295	1315
G(3849)	1476	1606	767
H(4060)	1081	2017	962
I(3860)	1673	1268	919
J(3195)	759	2045	391
K(5509)	2428	2547	534
L(3397)	1119	1448	830
M(6192)	1673	4000	519
N(4151)	1187	1927	1037
O(6899)	1540	3661	1698
P(3636)	1045	1930	653
Q(2578)	1141	983	454
R(8409)	4193	2170	2046
S(5218)	1429	1876	1913
T(5738)	754	3948	1036
TOTAL(91616)	29753	41459	20404
AVERAGE (4580.80)	1487.65	2072.95	1020.20

The candidate phrase selection, as shown in Figure 1, allows all necessary processing of the document three sections before selecting a phrase; punctuations are also removed in the phase. A stop word list is added, which includes a list of stop words in English language. All possible combination of words are formed, phrases that contain more than three words are discarded; this is because most keyphrase are combination of two words. The remaining phrases, combinations of one word and two words are checked in the filter list. The filter list contains the stem notation of words and a list of possible stop words. The porter's stemmer [16] is used for this study. After the words have been reduced to their individual stems, the stemmed word and the original form of the word are both saved, in case such words end up becoming a keyphrase. This is because the study suggests that after the removal of stop words and punctuation, all the remaining combinations of words are possible phrases. The output from this phase are possible phrases i.e. a combination of both relevant and irrelevant phrases. For the candidate phrase extraction, the tf*idf of a phrase is calculated separately for the three groups i.e. different scores for the three groups. The scores are combined before the noun phrase tags each word individually. The reason for the separation of the documents is to calculate the inverse document frequency. The system calculates the weight of each phrase by calculating its phrase frequency. The noun tagger tags a particular phrase containing a noun giving it a score. Research findings have noted that most possible keyphrases are mostly nouns, so any phrase containing a noun is tagged and stored. The tagged phrases are compared with the domain knowledge, which contains a list of keyphrases in Computer Science. Any phrase found in the domain is scored, noted and has a definite keyphrase while phrases that passed the noun tagger test but failed to be in the domain is added to the domain list. The final selected keyphrase would be a combination of keyphrase found in the domain list and keyphrase that passed the noun phrase tagger test. This is because a keyphrase could be a very good score from the noun phrase tagger test and not be in the domain list. The proposed system would improve over time because any learned keyphrases are added to the domain. The basic activities of the proposed model are as shown in Figure 2. The domain list contains a list of phrases in Computer Science. The study stored 500 phrases in the area of Data Mining and the phrases were collected from Computer Science journals from the web site - *http://airccse.org/journal/ijdkp/papers*.

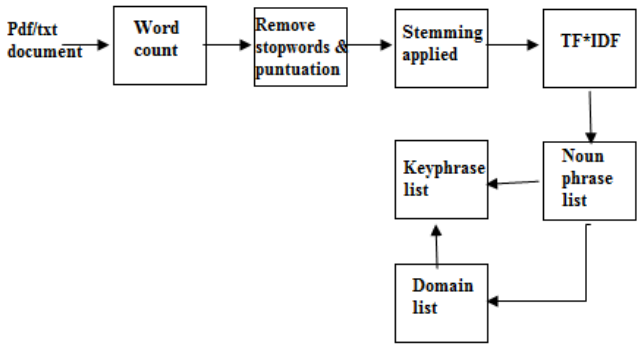

Figure 2. Basic Activities in the Proposed Model

The techniques used in the proposed keyphrase extraction model are explained as follows:

Stop word list: This contains a list of possible stop words and punctuations in English language. Stop words include articles such as a, an; pronouns such as he, she, they; prepositions such as under, on, below; conjunctions and interjections such as 'but'. Stop words totaling 187 words were used in this study. The stop words were extracted from the website: *http://www.ranks.nl/stopwords*.

Stemming method: The stemmer used for this study is the porter stem. This is because the porter stem is one of the most popular stemming methods, and most keyphrase that have proven to be successful have employed the use of the porter stemmer. It is based on the idea that the suffixes in the English Language (approximately 1200) are mostly made up of smaller and simpler suffixes. It is one of the fastest stemming algorithm, for example, it stems "believes to believ" and believable to believ.

Noun tagger: The noun tagger tags a word or phrase that is a noun or contains a noun in it. The noun tagger for this work is the controlled database of nouns in English language. The noun list was extracted from the web site

-http://www.talkenglish.com/vocalbulary/Top-1500-Nouns.aspx. This tags a particular word to be a noun or not, i.e. if a word contains a noun, a particular score is added to the previous score to boost its tf*idf score.

The proposed system generates keyphrases for journal articles using equation 1.

$$Kw = Pf * Idf + Nps * DL \quad (1)$$

where Kw represents the keyphrase weight or score;

Pf represents phrase frequency which is the number of times a phrase appear in a particular section given as *p/P*;

p is the number of times a phrase appears in a document;

P is the total number of phrases in the document;

Idf represents the inverse document frequency which is the number of sections a phrase appears given as log (*B/b*);

B is the total number of sections and b is the number of sections a phrase appears. B is set to (4) if a particular phrase appears in the three sections so as not to get a zero weight.

Nps represent the noun weight which checks if a particular phrase is totally a noun i.e. all words are nouns or partial nouns i.e. not all part of the phrase is a noun. It is given as *n/N*, where *n* is 1 for a total noun phrase and 0.5 for a partial noun phrase. *N* is the total number of nouns in the database and for this study, *N* is 1000 nouns. A score of 1 is assigned to a phrase *Nps* that does not contain a noun as part of it.

DL represents the domain score which scores a phrase found in the domain list and it is given as log (*D/d)*, where *D* is number of words in the domain at the instance of extraction and *d* is set to 1 if a word is found in the domain. A score of 1 is set for the phrase DL that is not domain specific i.e. not in the domain list.

The top 10 are ranked according to their *Kw* in the descending order.

Techniques used in the Proposed Model

The proposed system is an utility that demonstrates an innovative and convenient way of extracting keyphrases without complexities. The system works in a way that allows it to extract keyphrase from documents as regards the structure of the document. The system accepts .txt and .pdf documents and the words in the document can also be copied to the interface. The proposed keyphrase extraction model uses some already known techniques such as stopword list, stemming and noun tagger which are discussed one after the other as follows.

Stop word list: This contains a list of possible stop-words and punctuations in English language. Stop words include articles such as "a", "an" e.t.c; pronouns such as "he", "she", "they"; preposition such as "under", "on", "below" e.t.c; conjunction and interjection such as "but", "and" e.t.c. Stopwords of 187 words were used in this study. The stop-words were extracted from *http://www.ranks.nl/stopwords*.

Stemming methods: The stemmer used for this study was the porter stemmer. This is because the porter stem is one of the most popular stemming methods and most keyphrases that have proven to be successful have employed the use of the porter stemmer. It is based on the idea that the suffixes in the English Language (approximately 1200) are mostly made up of smaller and simpler suffixes. It is one of the fastest and least aggressive stemming algorithm, for example, it stems "believes to believ" and believable to believ.

Noun tagger: The noun tagger tags a word or phrase that is a noun or contains a noun in it. The noun tagger for this research was the controlled database of nouns in English language. The nouns were extracted from *http://www.talkenglish.com/vocalbulary/Top-1500-Nouns.aspx*. This will tag a particular word to be a noun or not i.e. if a word is a noun, a particular score is added to the previous score to boost its product of term frequency and inverse document frequency (tf*idf) score. The noun list holds the most frequent 1000 (one thousand) nouns used in English language, the noun list is a static list i.e. it does not grow over time.

Domain Knowledge: The domain knowledge contains a total of 564 (five hundred and sixty-four) phrases that were extracted from Computer Science journal articles in the area of data mining from *http://airccse.org/journal/ijdkp/papers* between the year 2011 and 2013. The phrases extracted contained one, two and three words as Author-assigned keyphrases in each document. The phrases extracted were first checked for redundancy after which all subset of the phrases were formed, for example an Author-assigned keyphrase like "Information retrieval Technology" was broken down to "Information", "retrieval" , "Technology", "Information retrieval", "retrieval technology". The phrase was limited to two words because the model is limited to phrase extraction that contains two words in them. The domain knowledge is a non-static list that contains new phrases that are learnt during extraction.

Term frequency-inverse document frequency: This method is used for calculating how important a word is to a document in a collection. It is basically a weighing factor used in information retrieval. The product of the term frequency and inverse document frequency (tf*idf) value increases according to the frequency of a particular word or phrase. For this study the term frequency multiplied by the inverse document frequency has been replaced by the phrase frequency multiplied by the inverse document frequency. This is so because we are dealing with keyphrases (two or more terms) not keywords (single term).

System Design, Implementation and Evaluation

The proposed system was designed using Unified Modeling Language (UML). The class diagram is as shown in Figure 3.

The system was implemented using Java programming language as the front-end and MySQL (version 5.0.51b) as the backend.

In order to evaluate the performance of the proposed system, it was tested using journal articles from the webpage *-http://airccse.org/journal/ijdkp/papers*. A total of twenty (20) journal articles for the year 2014 and year 2015 were considered. The journal articles contain author assigned

keyphrases. The twenty journals were used to test the system. In this study, keyphrases extracted by the system were compared to those assigned by the author. The results were then compared to those produced by MAUI. The reason MAUI was used is because the system is accessible at *(http://maui-indexer.appspot.com/)* and hence can be tested on the same dataset and in the same computing environment as the presented system. In the two systems, a match is considered to have been found if the stem of the extracted keyphrase matches the stem of any of the author assigned keywords where stemming is carried out by the porter stemmer. Three evaluation metrics were used: precision, recall and F-measure. In addition, the system comparison with MAUI was possible because both systems extracted 10 keyphrases per document.

Precision, in keyphrase extraction, can be defined as the proportion of the extracted keyphrases that match the keyphrases assigned by a document's author. It is denoted as:

$$Precision = A/(A + C) * 100\%$$

where A is the number of relevant keyphrases extracted, and C is the number of irrelevant keyphrases retrieved as compared with the author assigned keyphrases.

Recall, in keyphrase extraction, can be defined as the proportion of the keyphrases assigned by a document's author that are extracted by the system. It is denoted by:

$$Recall = A/(A + B) * 100\%$$

where A is the number of relevant keyphrases retrieved, and

B is the number of relevant keyphrases not retrieved as compared with the author assigned keyphrases.

F-measure, in information retrieval, is the harmonic mean of the precision and recall. It is given by:

$$F - measure = 2 * \frac{P * R}{P + R}$$

where P is Precision and R is Recall.

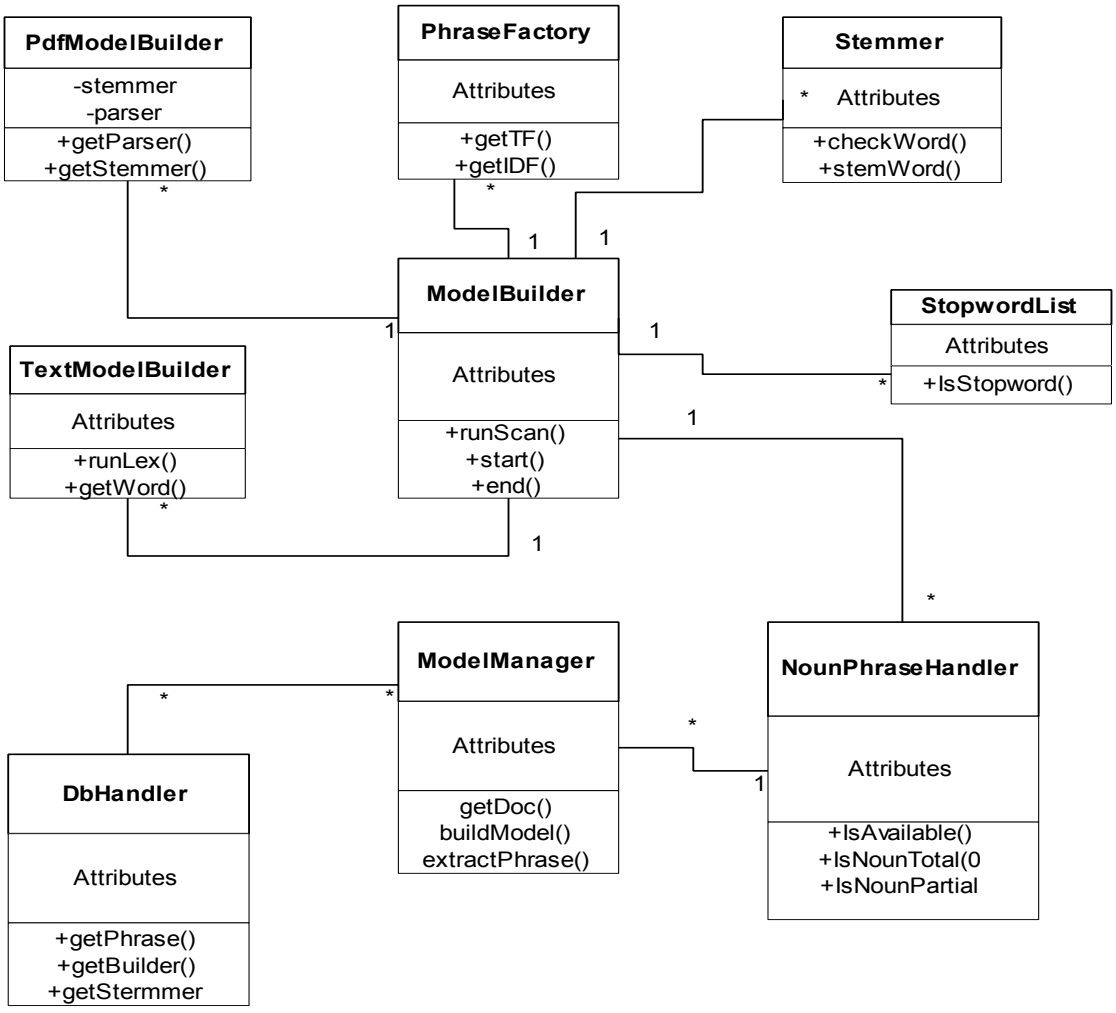

Figure 3. Class Diagram for the Proposed System

3. Results and Discussion

Evaluation was carried out by calculating how many of the keyphrases generated by the developed system match with the author assigned ones; and then comparing these to those extracted by *MAUI*. Sample results of top ten extracted keyphrases by the proposed system and MAUI from three different documents are as shown in Table 2 with the user interface as shown in Figure 4. The evaluation scores are as shown in Table 3. The table depicts the precision, the recall and the f-measure results of the two systems. The line graph for the precision, recall and f-measure are depicted in Figure 5, Figure 6 and Figure 7. Specifically, when extracting ten keyphrases from twenty documents, the proposed system approximately has an average precision of 27% while MAUI approximately has an average precision of 23%, both as compared with the author assigned keywords. The average recall for the proposed system is 53% while MAUI has an average recall of 45%, both as compared with the author assigned keywords. The average F-measure for the proposed system is approximately 35% while MAUI has an average F-measure of 30%. It is worth noting that the developed system consistently outperforms MAUI on the dataset used.

Table 2. Top Ten Extracted Keyphrases by the Proposed System and MAUI from Three Different Documents

1.) Document's Title: Enhancing The Labelling Technique Of Suffix Tree Clustering Algorithm	
Author assigned keyphrase: Information retrieval, clustering, search results clustering, suffix tree clustering, cluster labeling	
Proposed System = (2 matches)	**MAUI = (2 matches)**
documents clusters	cluster
clustering uses	data mining
cluster labels	journal of data
clustering results	international journal
information retrieval	knowledge management process
Framework	cluster label
root	management process knowledge management
retrieval systems	international journal of data
labels	international journal of data mining
knowledge	system

2.) Document's Title: Mining Frequent Itemsets (MFI) over data streams: Variable window size(VWS) by Context Variation Analysis(CVA) of the streaming Transactions.	
Author assigned keyphrase: Data mining, Data streams, Frequent itemset, Frequent itemset mining, Data stream mining, Variable window, Sliding window	
Proposed System = (5 matches)	**MAUI = (3 matches)**
itemset mining	Frequent itemsets
information	data streams
management process	context variation
variable window	variation analysis
frequent itemset	window size
variations	VWS CVA
window	international journal
retrieval systems	knowledge management
data stream	MFI VWS
data mining	window

3.) Document's Title: A Fuzzy logic based on Sentiment Classification				
Author assigned keyphrase: Sentiment classifcation, Fuzzy logic, Fuzzy rules, Fuzzy c-means algorithm, Text minir				
Proposed System = (1 match)		**MAUI = (2 matches)**		
words		sentiment classification		
topic words		based		
clustering words		transcripts		
knowledge dataset		journal of data		
clustering dataset		using		
method		fuzzy logic		
dataset		knowledge management process		
fuzzy logic		international journal		
example words		knowledge management process		
extraction method		sentiment classifcation		

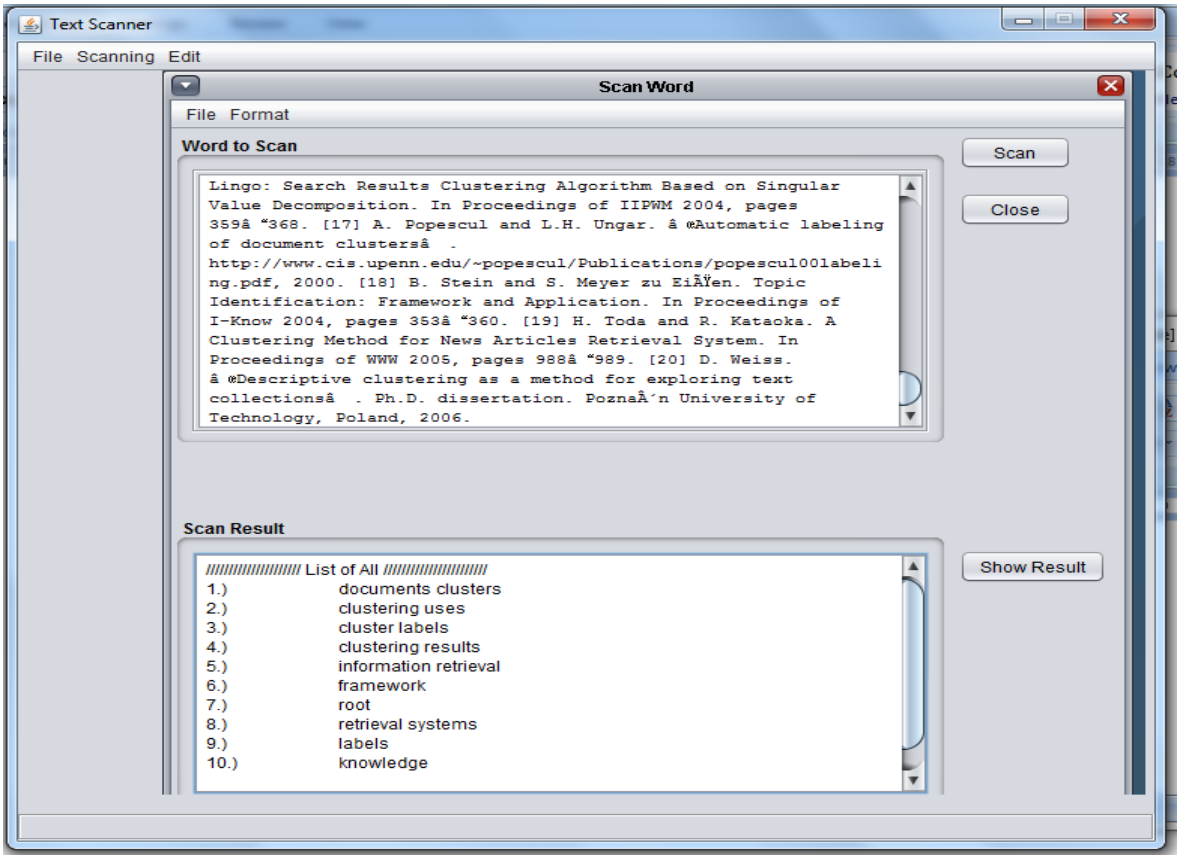

Figure 4. User Interface after Scanned Documents.

Table 3. Performance Evaluation of the Proposed System and MAUI

	Precision	Precision	Recall	Recall	F-measure	F-measure
Document title	Proposed system	MAUI	Proposed System	MAUI	Proposed system	MAUI
d1	0.20	0.20	0.67	0.67	0.31	0.31
d2	**0.40**	0.20	**0.80**	0.40	**0.53**	0.27
d3	0.10	**0.20**	0.33	0.33	0.15	**0.25**
d4	0.10	**0.20**	0.33	**0.67**	0.15	**0.31**
d5	**0.30**	0.10	**0.60**	0.20	**0.40**	0.13
d6	0.30	0.30	0.60	0.60	0.40	0.40
d7	0.30	0.30	0.60	0.60	0.40	0.40
d8	0.30	0.30	0.60	**0.75**	0.40	**0.43**
d9	**0.30**	0.20	**0.60**	0.20	**0.40**	0.20
d10	0.20	**0.30**	0.40	**0.60**	0.27	**0.40**
d11	**0.30**	0.10	**0.75**	0.25	**0.43**	0.14
d12	0.20	**0.30**	0.40	**0.60**	0.27	**0.40**
d13	**0.30**	0.10	0.25	0.25	**0.27**	0.14
d14	**0.50**	0.30	**0.71**	0.43	0.59	**0.35**
d15	0.10	**0.20**	0.20	**0.40**	0.13	**0.27**
d16	**0.20**	0.10	**0.67**	0.33	**0.31**	0.15
d17	0.30	0.30	0.50	0.50	0.38	0.38
d18	0.30	0.30	0.27	0.28	0.29	0.29
d19	0.30	0.30	0.60	0.60	0.40	0.40
d20	**0.40**	0.30	**0.67**	0.33	**0.50**	0.32
Average	**0.27**	0.23	**0.53**	0.45	**0.35**	0.30

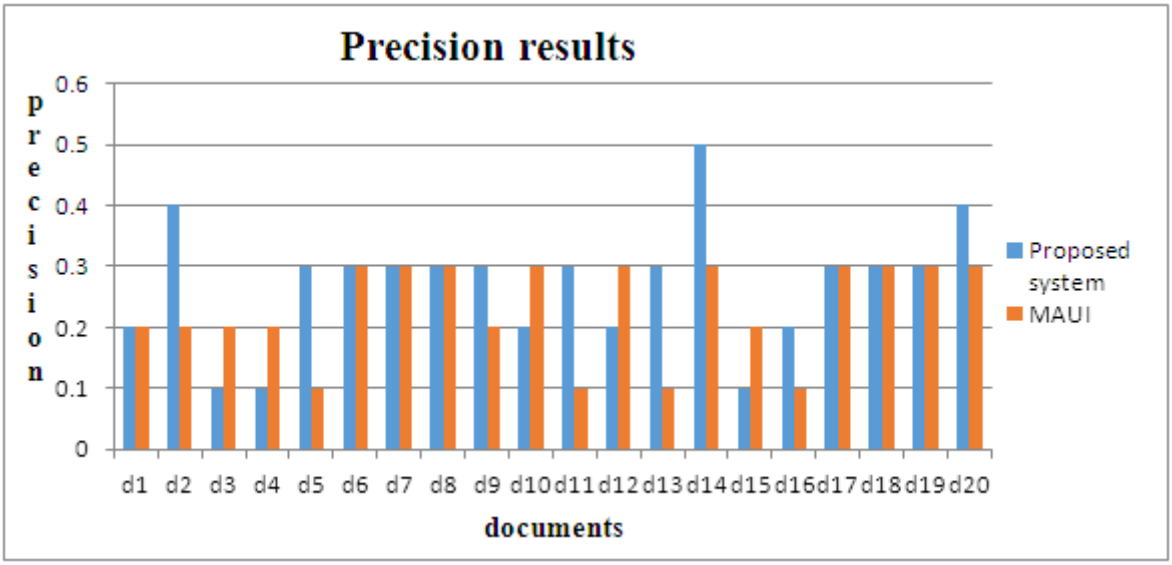

Figure 5. Precision Results for the Proposed System and MAUI

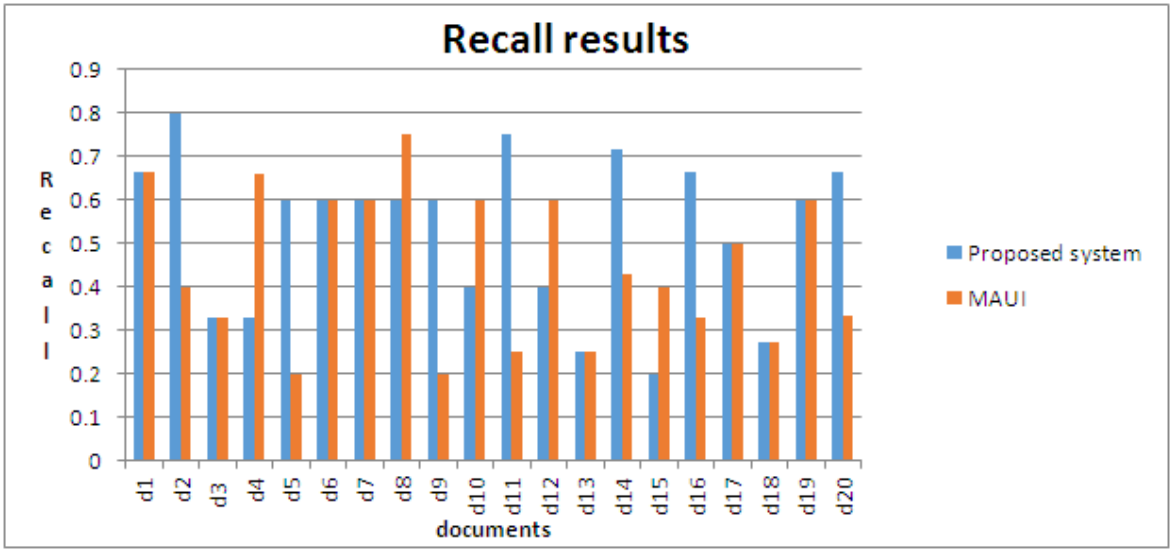

Figure 6. Recall Results for the Proposed System and MAUI

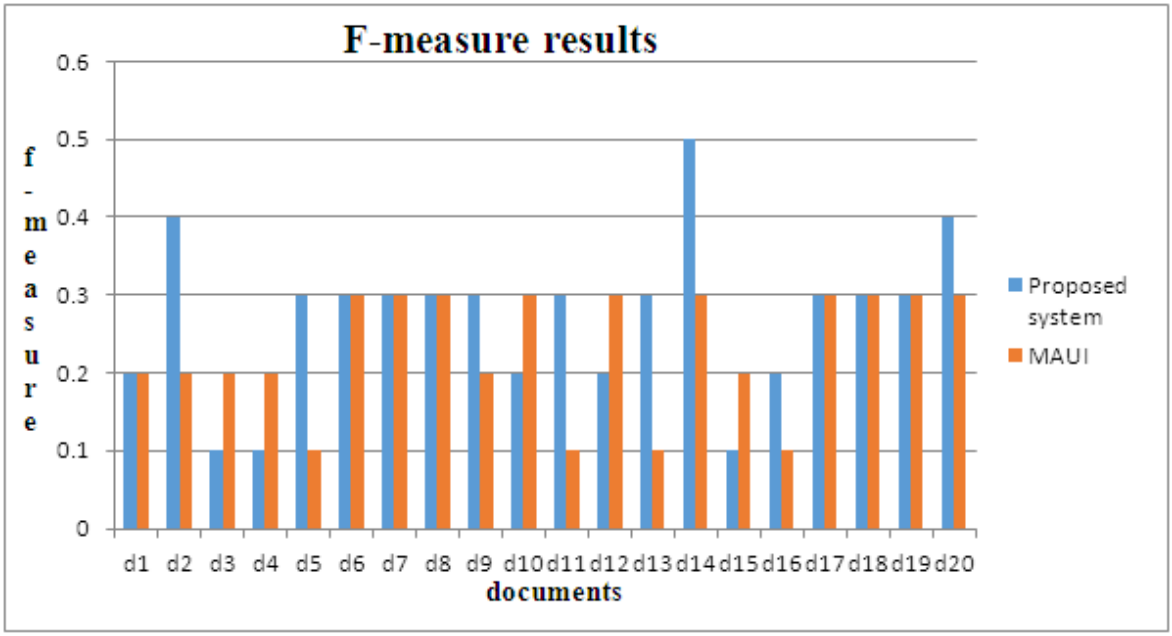

Figure 7. F-measure Result for the Proposed System and MAUI

4. Conclusions

This study has shown that the proposed model is effective for the task of automatic keyphrase extraction of journal articles in the specific area of Data mining in Computer Science. Unlike other extraction systems, it has the advantage of taking the structure of the document into consideration and eradicates the need to train a particular system, due to the domain knowledge that is being built over time. Developers of systems that use keyphrase extraction can also easily make use of the introduced domain knowledge to fit their requirements. In addition, other parts of speech can be integrated to produce more quality keyphrases.

REFERENCES

[1] C.M. Millward and M. Hayes, "A Biography of the English Language". Third edition: Wadsworth Cengage Learning (2012).

[2] D. P. Turney, "Learning Algorithms for Keyphrase Extraction". Information Retrieval. pp. 303- 336, 1999.

[3] K. Sarkar, "A Hybrid Approach to Extract Keyphrases from Medical Document". International Journal of Computer Applications. Vol 63, pp.14-19, 2013.

[4] C. Zhang, H. Wang, Y. Liu, D. Wu, Y. Liao and B. Wang,

"Automatic Keyword Extraction from Documents using Conditional Random Fields". Journal of Computation Information System. Vol 4, pp.1169-1180, 2008.

[5] A. K. Mondal and D. K. Maji, "Improved Algorithms for Keyword Extraction and Headline Generation from Unstructured Text". M.Sc. Thesis. Indian Institute of Technology, Kanpur, 2004.

[6] E. Nicolai, B.S. Pedro, G. Iryna and Z. Torsten, "DKPro Keyphrases: Flexible and Reusable Keyphrase Extraction Experiments". In the Proceedings of 52nd Annual meeting of the Association for Computational Logistics, June 2014. pp 71-76, 2014.

[7] A. Hulth, "Improved Automatic Keyword Extraction Given More Linguistic Knowledge". In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2003), pp. 216–223, 2003.

[8] E. Frank, G. W. Paynter, J. H. Witten, C. Gutwin and C. G. Nevill-manning, "KEA: Practical Automatic Keyphrase Extraction". In Proceeding of the International Joint Conference on Artificial Intelligence (IJCAI "99"). pp. 668-673, Stockholm Sweden, 1999.

[9] F. Liu, D. Pennell, F. Liu and Y. Liu, "Unsupervised Approach for Automatic Keyword Extraction using Meeting Transcripts". Association for Computational Logistics. pp. 620-628, 2009.

[10] P. Bhaskar, K. Nongmeikapam and S. Bandyopadhyay, "Keyphrase Extraction in Scientific Articles: A supervised approach". International Conference on Computational Linguistic. pp. 17-24, 2012.

[11] P. Bhaskar, "Multi-document Summarization using Automatic Keyphrase Extraction". In Proceedings of the Student Research Workshop associated with RANLP 2013, Hissar Bulgaria. pp 22-29, 2013.

[12] L. Teixeira, G. Lopes and R.A. Ribeiro, "Automatic Extraction of Document Topics". International Federation for Information Processing, pp. 101-108, 2011.

[13] K. Sarkar, M. Nasipuri and S. Ghose, "A New Approach to Keyphrase Extraction using Neural Networks". International Journal of Computer Science Issues. Vol 7, pp. 16-25, 2010.

[14] Medelyan, O., Frank, E., and Witten, H.I. (2011). Human-competitive Tagging using Automatic Keyphrase Extraction. International Journal of Computer Science Issues. Vol 7, pp. 25-35.

[15] S. R. El-bethagy and A. Rafea, "Kp-miner: A Keyphrase Extraction System for English and Arabic documents". Elsevier Information systems. 34(1):132–144, 2009.