J Inf Process Syst, Vol.9, No.4, pp.602-620, December 2013
http://dx.doi.org/10.3745/JIPS.2013.9.4.602
pISSN 1976-913X
eISSN 2092-805X

Automatic Single Document Text Summarization Using Key Concepts in Documents

Kamal Sarkar*

Abstract—Many previous research studies on extractive text summarization consider a subset of words in a document as keywords and use a sentence ranking function that ranks sentences based on their similarities with the list of extracted keywords. But the use of key concepts in automatic text summarization task has received less attention in literature on summarization. The proposed work uses key concepts identified from a document for creating a summary of the document. We view single-word or multi-word keyphrases of a document as the important concepts that a document elaborates on. Our work is based on the hypothesis that an extract is an elaboration of the important concepts to some permissible extent and it is controlled by the given summary length restriction. In other words, our method of text summarization chooses a subset of sentences from a document that maximizes the important concepts in the final summary. To allow diverse information in the summary, for each important concept, we select one sentence that is the best possible elaboration of the concept. Accordingly, the most important concept will contribute first to the summary, then to the second best concept, and so on. To prove the effectiveness of our proposed summarization method, we have compared it to some state-of-the art summarization systems and the results show that the proposed method outperforms the existing systems to which it is compared.

Keywords—Automatic Text Summarization, Key Concepts, Keyphrase Extraction

1. INTRODUCTION

Automatic text summarization can be used for managing the huge amount of information on the web. The abstracts or summaries help the reader to get a quick overview of an entire document or a document set and thus it reduces reading time.

The automatic text summarization process can be useful for a variety of applications such as document indexing, question-answering systems, aiding retrieval systems, and document classification.

A machine-generated summary is also free from bias and the use of automatic or semi-automatic summarization by commercial abstract services may allow them to scale the number of published texts that they can evaluate.

Depending upon the number of documents accepted as input by a summarization process, automatic text summarization can be categorized as single document summarization and multi-document summarization. When only one document is the input, it is called a single document text summarization and when the input is a set of related text documents, it is called a multi-

Manuscript received January 16, 2013; first revision June 11, 2013; accepted August 25, 2013.
Corresponding Author: Kamal Sarkar (jukamal2001@yahoo.com)
* Dept. of Computer Science and Engineering, Jadavur University, Kolkata, India (jukamal2001@yahoo.com)

document summarization.

Text summarization is also categorized based on the type of users the summary is intended for. User focused (query focused) summaries are tailored to the requirements of a particular user or group of users and generic summaries are aimed at a broad readership community [1]

A summary is produced in the form of an abstract or an extract. An extract is a summary consisting of a number of important textual units selected from the input(s). An abstract is a summary that represents the subject matter of an input(s) with the text units, which are generated by reformulating the salient units selected from an input(s). An abstract may contain some materials, which are not present in the input document(s).

Based on information contained in the summary, it is called an informative summary and an indicative summary. The indicative summary gives an indication about an article's purpose and it provides the user with an approach for selecting the article for in-depth reading; whereas, an informative summary covers all salient information in the document to some degree of detail. For example, the abstract of a research article is more informative than its headline.

The earliest works on text summarization use sentence extraction as a primary component of a text summarization system where the sentence extraction component uses sentence position, word importance, cue phrases, title information, and sentence length as features for ranking sentences. The centroid based summarization system [2] uses the centroid of a document set (the set may contain a single document for single document summarization task) as a feature where the centroid is a pseudo document consisting of a set of terms whose TF*IDF values (TF: Term Frequency, IDF: Inverse Document Frequency) are greater than the predefined threshold. This method computes the similarity of a sentence to the centroid and considers this similarity value to be the score for the sentence. The work presented in [2] also combines the centroid feature with some of the features such as sentence position, cue phrases, title information, and sentence length etc. for improving the performance of the extraction-based summarization. Although the centroid based summarization system computes sentence scores based on the similarities of the sentences to the centroid, which is represented as a collection of single-word keywords, it ignores the uses of multi-word keyphrases for text summarization. Moreover, it does not use keyphrases for maximizing diverse, yet relevant, information in the summary. Thus, our proposed summarization approach is different from the centroid based summarization approach.

Many research studies on keyphrase extraction [3-5] have marked text summarization, automatic indexing, etc. as the application areas of keyphrase extraction. The work presented in [6] considers keyphrases as key concepts and incorporates key concepts in search results. Unlike the above-mentioned works that only concentrated on the keyphrase extraction task, our proposed work investigates the uses of keyphrases (key concepts) extracted by a keyphrase extraction system in improving the summarization performance. Our proposed summarization approach uses keyphrases that are treated as key concepts for adjusting the important factors that maximize diverse, yet relevant, information in the summary and thus it improves the summarization performance.

The main contributions of our work are as listed below.

(1) Novel uses of key concepts in summary generation by maximizing diverse, yet relevant, information in the summary. Since redundancy is a critical issue for the text summarization task, we have proposed an efficient method that exploits keyphrases for non-redundant (diverse) sentence selection in the final summary. We also use a position-based sentence

filtering technique to eliminate the less important sentences.

(2) We propose a two-phase approach to summary sentence selection. In this two-phase method, Phase 1 uses positional information and document keyphrases in an effective manner for summary sentence selection. Phase 2, which is activated when Phase 1 cannot produce a summary of the desired length, combines the positional information and TFIDF information for summary sentence selection. The criteria for Phase 1 are tougher than those for Phase 2. If the input document is relatively longer, it should have many candidate sentences participating in the competition for selection and so tough a competitive environment in the first phase will help to choose a few sentences from many candidate summary sentences. On the other hand, for a shorter document it may happen that the summary of the desired length may not be produced in the first phase since the criteria are tough. Then, Phase 2, which considers relatively relaxed sentence-selection criteria, is activated to generate the summary of the desired length.

In Section 2, we present the previous works related to our proposed method. The proposed summarization method is presented in Section 3. In Section 4, we describe the summary evaluation method and experimental results.

2. RELATED WORK

The earliest works on text summarization are sentence extraction-based. A sentence extraction based text summarization method has two important steps. The first step involves ranking the sentences based on their scores, which are computed by combining few or all of the features such as term frequency (the number of times a word occurs in a document), positional information (the sentences that occur early in a document are assigned higher scores than those that occur relatively late in the document), and cue phrases (such as “more importantly” or “precisely”) [7-11].

In the second step, K top ranked sentences are selected to form an extract where the users specify K, based on the summary compression ratio.

The very first work on automatic text summarization presented in [8] computes salient sentences based on word frequency (the number of times a word occurs in a document) and phrase frequency.

A simple method of sentence extraction based summarization, which is presented in [7], uses headline information, the first and last sentences of the document, and/or each paragraph.

MEAD [2] is a text summarization system that computes the score of a sentence based on the features such as: similarity of the sentence to the centroid (the centroid is basically a collection of words whose weight is greater than a predefined threshold), the similarity to the first sentence of the document, the position of the sentence in the document, and sentence length.

A machine learning based text summarization approach has been proposed in [12]. Given a training corpus of text documents and their summaries, they developed a summarizer that uses a learning algorithm to classify the sentences of a document as summary worthy sentences and summary unworthy sentences. They applied the machine learning algorithm called, *bagging*, to this learning task, where a C4.5 decision tree has been chosen as the base learner.

A sentence extraction based summarization method that exploits the semantic links between sentences in the text has been presented in [13]. The feature used in this work may be considered to be a cohesion feature. *Text cohesion* refers to the relationship (semantic links) between words, word senses, or referring expressions, which determine how tightly connected the text is. In this approach, text is represented by a graph in which each node represents a paragraph in a document and the edges are labeled with the similarity score between two paragraphs. The paragraph that is connected to many other paragraphs with a similarity above a predefined threshold is considered to be the *bushy node*. The paragraph representing the "bushy" node is considered to be a salient one.

An EM algorithm based method for automatic text summarization has been presented in [14]. In this approach, the EM algorithm has been used to form groups of similar sentences in order to obtain a representative sentence from each cluster to create the summary.

The *Lexical chaining* method for computing the salience of a sentence has been presented in [15]. *Lexical cohesion* [16] links the different parts of the text through semantically related terms, co-references, ellipsis, and conjunctions. Lexical cohesion also involves relations such as reiteration, synonymy, and hypernymy. The concept of the lexical chain was introduced in [17], which defines a lexical chain as a sequence of related terms that spans a topical unit of text. A Lexical chain is basically a lexical cohesion that occurs between two terms and among sequences of related words. Barzilay and Elhadad [15] used WordNet to compute the lexical chains.

A genetic algorithm based text summarization approach has been presented in [18]. In this approach, the i^{th} sentence in the document has been considered to be the gene of a chromosome where the value of a gene can be either 0 (indicating that the i^{th} sentence is not included in the summary) or 1 (indicating that the i^{th} sentence is included in the summary). The fitness function is defined as a measure that uses positional and term frequency information.

The work in [19] applied Hidden Markov Models (HMMs) to a sentence extraction task. This work considers that the probability of the inclusion of a sentence in an extract depends on whether the previous sentence had been included as well.

A maximum entropy (log-linear) model for identifying summary sentences in a document is presented in [20]. The features considered in this work are word pairs, sentence length, sentence position, and discourse features (e.g., whether the sentence follows the "Introduction," etc.).

Compared to creating an extract, automatic abstract generation is harder in the sense that the latter requires a deeper linguistic knowledge, such as semantic properties, of the text. Abstract generation requires the semantic representation of text units (sentences or paragraphs) in a text, the reformulation of two or more text units, and rendering the new representation in a natural language. Abstractive approaches have used template based information extraction, information fusion, and compression [21-23].

Headline (title) generation can be viewed as the generation of a very short summary (usually less than 10 words), which gives an indication about the information content of a document. A headline summary is a kind of indicative summary. The approach presented in [24] uses some statistical methods to generate headline-like abstracts.

The HMM (Hidden Markov Model) based headline generation has been described in [25].

Dorr et al. [26] developed the Hedge Trimmer, which uses a parse-and-trim based approach to generate headlines. In this approach, the first sentence of a document is parsed using a parser. Then, the parsed sentence is compressed to form a headline by eliminating unimportant and low content units using a set of linguistically motivated rules.

TOPIARY [27], which is a headline generation system, combines the compressed version of the lead sentence and a set of topic descriptors (key words) that have been generated from the corpus to form a headline. Here the sentence is also compressed using a rule-based approach that is similar to the approach in [26] and the compressed sentence is augmented with the topic descriptors to form a headline.

Many approaches to abstract generation have placed less emphasis on the issue that a true abstract may contain some information that is not contained in the document.

The Document Understanding Conference (DUC) is an international forum that discusses the area of automatic summarization. From time to time, the DUC carries out extensive evaluations on the several defined tasks.

The National Institute of Standards and Technology (NIST), in the USA, organized the first Document Understanding Conference (DUC) in 2001 to provide researchers with a common platform to evaluate their summarization systems. Subsequently, the DUC has become a yearly event. Every year, the DUC defines different summarization tasks, plans for the creation of reference data (documents and summaries for training and testing), and evaluates the summarization systems that are participating in the various tasks. Over the course of its first six years, the DUC has examined automatic single and multi-document summarization of newspaper/wire articles, with both generic tasks and various focused tasks. Nowadays, the Text Analysis Conference (TAC)[1] provides a forum for the assessment of different information access technologies, including text summarization.

The DUC and TAC undoubtedly play an important role in the progress of summarization research. In a series of DUC conferences held from 2001 onwards, the various tasks on single document summarization, generic multi-document summarization, query-focused summarization, update summarization, and opinion mining have been defined and evaluated. Out of these DUC conferences, the single document summarization tasks were only considered in 2001 and 2002. We have used the DUC 2001 data set for training our proposed summarization system and we tested our system on the DUC 2002 data set and compared it to the performances of the single document summarizers participating in DUC 2002.

Wan et al. [28] presents an approach to single document text summarization that uses neighborhood knowledge that has been constructed with a small number of nearest neighbor documents close to the specified document for improving document summarization under the assumption that the neighbor documents could provide additional knowledge and more clues. This approach has been tested on the DUC 2002 data set. We have also compared our proposed approach to this approach. The details of the approach proposed by Wan et al. are presented in Subsection 4.4.

We have also compared our approach to a single document text summarization approach presented in [29], which uses anaphora resolution and Latent Semantic Analysis (LSA) to develop an LSA-based summarizer, which achieves a significantly better performance than a system that does not use anaphoric information. The approach presented in [29] has also been tested on the DUC 2002 data set.

1 http://www.nist.gov/tac

3. THE PROPOSED SUMMARIZATION METHOD

Our proposed summarization method is extraction based. It has several major steps: (1) preprocessing, (2) keyphrase extraction, and (3) summary generation.

3.1 Preprocessing

The preprocessing step includes stop-word removal and breaking the input document into a collection of sentences.

3.2 Keyphrase extraction

The main focus of the work presented in this paper is on the use of keyphrases in the text summarization task. To keep the summarization method as simple as possible, the keyphrase extraction component of the proposed summarization system uses a simple keyphrase extraction method, which is a variant of the method presented in [30]. We use a simple method for keyphrase extraction. It has the following two major steps: identifying the candidate keyphrases and ranking the candidate keyphrases for extracting the keyphrases. We consider that a candidate keyphrase is a sequence of words containing no punctuations and stop words. A list of common verbs is also added to the stop word list because we have observed that the keyphrases rarely contain common verbs. The process of candidate keyphrase extraction also has two steps, which are as follows:

Step 1: Extracting candidate keyphrases by considering punctuations and stop words as the phrase boundary and forming a candidate phrase list with the extracted phrases.

Step 2: Further breaking the phrases selected in Step 1 into smaller phrases using the following rules:

i If a phrase is L-word long, all n-grams (n varies from 1 to L-1) are generated and added to the candidate phrase list

ii If a phrase is longer than 5 words, it is discarded

Sample Sentence:
This study was one of the first to investigate potential risk factors for anxiety (i.e., behavioral inhibition, parental negative affect, parenting stress) in early childhood.

Initial list of candidate keyphrases (after Step 1)
study, investigate potential risk factors, anxiety, behavioral inhibition, parental negative affect, parenting stress, childhood

The list of candidate phrases (after Step 2)
study, investigate, potential, risk, factors, investigate potential, potential risk, risk factors, investigate potential risk, potential risk factors, anxiety, behavioral inhibition, behavioral, inhibition, parental negative affect, parental, negative, affect, parental negative, negative affect, parenting stress, parenting, stress, childhood

Fig. 1. A sample sentence and the candidate keyphrases identified from this sentence

Fig. 1 shows a sample sentence and the candidate keyphrases identified from the sentence. Some candidate phrases, which have been generated using the above-mentioned method, may not be meaningful to human readers. For example, in Fig. 1, the candidate phrase "investigate potential" is less meaningful. After computing phrase frequency, such candidate keyphrases are filtered out. For this purpose, we apply two conditions. The first condition states that a candidate keyphrase should occur at least twice in a document.

The second condition is related to the first appearance of the phrase in the document. Previous works [3] have suggested that keyphrases appear sooner in an article. The works in [30] define a threshold value as: when a phrase appears for the first time, after a certain number of words, it is filtered out and ignored.

In our experiments, the value for word-cutoff has been set to 400, which is suggested in [30] for obtaining the best results on keyphrase extraction. So, the phrases that appear for the first time after the 400-word cutoff are not considered as candidate keyphrases.

3.2.1 Assigning scores to candidate keyphrases

In general, a document has a lower number of author assigned keyphrases. To select a small subset of candidates as the keyphrases requires assigning weights to the candidates and ranking them based on these weights. The weight of a candidate keyphrase is computed using the two important features of Phrase Frequency (PF) and Inverse Document Frequency (IDF) as follows:

$$SCORE_{\text{pf*idf}} = \begin{cases} \text{PF*IDF}, & \text{if plength=1} \\ \text{PF*log(N)}, & \text{if plength>1} \end{cases} \quad (1)$$

Where:

plength= length of a phrase in terms of words

PF = phrase frequency, which is counted as the number of times a phrase occurs in a document

IDF= log(N/DF), where N is the total number of documents in the corpus (a collection of documents in a domain under consideration) and DF is the number of documents in which a phrase occurs at least once. Equation (1) shows that for multi-word phrases, the phrase score is computed using the PF * log (N), which is basically PF * IDF with the DF set to 1. This is due to the fact that multi-word phrases do not occur as frequently as single-word phrases do within a relatively small collection of documents.

After ranking the candidate keyphrases in the decreasing order of their scores, the top *m* candidates are selected as the keyphrases. Our proposed summarization system starts with all the ranked candidate keyphrases and chooses keyphrases from the top of the list to use them in the summarization process. In fact, how many keyphrases should contribute to the final summary depends on the summary length and the nature of the input document. The detailed method has been presented in the summary generation step, which is discussed in the next subsection.

3.3 Summary generation

We extracted the document keyphrases as the first abstraction level of a document. The keyphrases of a document can be viewed as the important concepts that a document elaborates on. We hypothesized that an extract is an elaboration of the important concepts that is to some permissible extent controlled by the given summary length. In other words, for summary genera-

tion, we chose a subset of sentences from a document that maximizes the important concepts in the final summary. While selecting sentences in the summary with the objective of maximizing important concepts, care must be taken to allow the summary to contain diverse information because space should not be exhausted with the discussion of only a single important concept when a document may contain many important concepts. So, to allow for these to be diverse information in the summary, for each important concept, we selected one sentence that is the best possible elaboration of the concept. The most important concept will first contribute to the summary, then to the second best concept and so on. When a sentence is selected in a summary, the selected sentence may cover one or more key concepts. So, if the key concepts, which are contained in the previously selected sentences, are found in the global list of key concepts for a document, they are removed from the global list.

To implement the above-mentioned sentence selection process, all of the sentences containing a key concept are grouped and the sentences in a group are ranked based on scores computed using TF*IDF (TF: Term Frequency and IDF: Inverse Document Frequency) and position features. Finally, the sentence with the highest score in the group is selected into the summary. For computing the score of a sentence based on these two features, the observed features are translated into two different scores, which indicate the summary worthiness of a sentence. Here we consider that two different properties (features) provide two weak pieces of evidence for the summary worthiness of a sentence in a document and when they are combined together, it gives stronger evidence to summary worthiness of the sentence in terms of a unique score that facilitates sentence ranking. Thus, the score of a sentence is computed using the following equations:

$$score_{pos} = \frac{1}{\sqrt{i}} \ , where\ i\ is\ the\ sentence\ position \tag{2}$$

$$Score_{tfidf} = \sum_{w \in s} TF(w) * IDF(w) \tag{3}$$

$$score(s) = score_{pos} + \left(score_{tfidf} \Big/ \max\{score_{tfidf}\} \right) \tag{4}$$

Where:

The position of a sentence s in a document is indicated by i,

TF (w) (term frequency) is the number of times a word w occurs in a document,

IDF (w) (Inverse Document Frequency) is computed on a corpus using the formula:

IDF (w)=log(N/df(w)), where *N* = number of documents in the corpus and *df(w)* (document frequency) indicates the number of documents in which a word w occurs at least once,

$Score_{pos}$ is the score of a sentence S due to its position in a document,

$Score_{\text{tfidf}}$ is the score of a sentence S based on TF*IDF feature,

$score(s)$ is the overall score of the sentence s,

$\max(score_{tfidf}\}$ is the maximum obtained sentence-score based on the TF*IDF feature.

One of the problems with TF*IDF based scoring is that sometimes longer sentences get a higher score merely due to fact that they contain more words. So, the words whose TF*IDF value is less than the predefined threshold (we set this threshold value to 2.5 for our experiments) are removed from a sentence as noise, and the score of the sentence ($Score_{\text{tfidf}}$) is computed with the TF*IDF values of the remaining words.

The algorithm for sentence selection and summary generation works in two phases, which are as follows: (1) Phase 1 considers the first n sentences from a document as the candidate summary sentences, ignores the rest, clusters the candidate summary sentences into groups where one group is formed per key concept (that is, the sentences containing a particular key concept are put into a group), and chooses the best sentence from each group as a summary sentence; and (2) Phase 2 is activated if the desired summary length is not reached in Phase 1. Phase 2 revises all the sentences from the document for selection into the summary, ranks sentences using Equation 4, chooses sentences one by one from the ranked list of sentences, and adds a chosen sentence to the summary if it was not already selected into the summary during Phase 1.

Phase 1 of the summarization algorithm presented here is tighter than Phase 2, which is only used whenever Phase 1 is unable to produce a summary of the desired length. Through observation and manual verification of the reference summaries, it has been found that most summary sentences are selected from the sentences that appear early in the document. So, Phase 1 of the algorithm makes the sentence selection criteria tougher and it restricts the candidate summary sentences to a lesser number by setting the threshold of n on the positions of the sentences. Whereas, Phase 2 is relatively relaxed and is used to fill the space in the summary that remains unfilled during Phase 1. Phase 2 might be omitted by relaxing the threshold value for sentence position, but through experimentations, it has been found that setting higher values to n degrades the summarization performance.

In order to select an optimal position threshold value, an experiment has been set up. In this experiment, a DUC 2001 dataset was used for tuning the threshold values. In the devised experiment on adjusting the threshold value for sentence position, the summarizer was run on the data set with a position threshold value, where the value ranged from 1 to approximately the half of the average document size. Each time the experiment was run, the position threshold value is incremented by 1 and the summaries generated by the summarizer are evaluated using the ROUGE-1 average F-score [31]. After all of the summaries for the input documents are generated, the average ROUGE-1 score for each threshold value was calculated. Since the whole idea of having a threshold value is to maximize the summarization performance, a score was obtained by normalizing these values by dividing each average value by the maximum obtained average. So, the value of one represents the highest summarization performance obtainable by the proposed summarizer.

Figure 2 shows the effect on summarization performance when the threshold value *(n)* for sentence position is varied.

The graph shown in Fig. 2 suggests that the maximum improvement in the summarization performance on a set of documents at hand is obtained whenever the threshold is set to 5.

The detailed algorithm for creating a summary using the key concepts (keyphrases) and sentence importance is presented in Fig. 3. The detailed discussion on the algorithm is given below.

The input to the algorithm are a list of sentences with their scores computed using Equation 4; a list of keyphrases (key concepts), which are ranked in decreasing order of their weights, and that are computed using Equation 1; and a predefined threshold on sentence position. The output

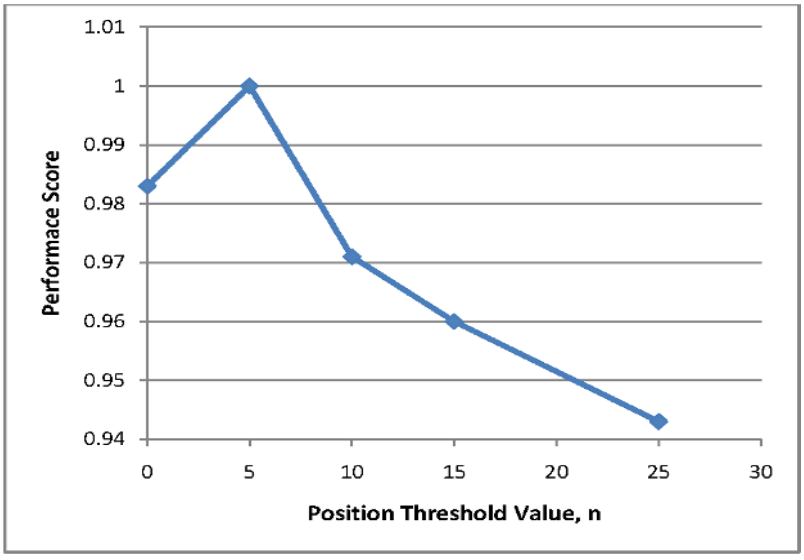

Fig. 2. ffect on summarization performance when the threshold value for sentence position is varied

of the algorithm is a summary.

Step 1 of the algorithm initializes the variables where the output variable SUMMARY is initially set to "empty" and it is gradually updated based on sentence selection criteria.

The variable *KPcoverage* monitors the list of concepts covered by all the sentences previously selected in *SUMMARY*. The algorithm consults the information maintained in *KPcoverage*, while a sentence is considered for being selected into the summary. The pointer p points to the first member (the topmost keyphrase) in the ranked list of keyphrases, K. The algorithm works in two phases.

The first phase stops when the summary of desired length L is achieved. Otherwise sentence selection proceeds in the following way: the topmost keyphrase is selected from the ranked list of keyphrases and if the selected keyphrase is not present in the previously selected sentences, the sentences containing the keyphrase are collected from the document to make a pool, and then after removing from the pool the sentences with position greater than a predefined threshold, the most important sentence from the pool is selected and included into the *SUMMARY*, and then the *KPcoverage* is updated with the keyphrases that are present in the currently selected sentence.

Phase 2 of the algorithm is activated if the summary of the desired length is not produced in Phase 1. Such a situation may arise when the number of keyphrases in list K is relatively small. For example, a shorter document may contain a fewer number of phrases that satisfy the criteria (discussed in Subsection 3.2) for being selected as candidate keyphrases. In Phase 2, all of the sentences in the document are considered and sorted in decreasing order of their scores. The sentences from the sorted list are then selected and included in the summary one by one, if the sentence under consideration is not already present in the summary. Phase 2 stops sentence selection when the summary of the desired length of L is produced.

Algorithm 1: Creating a Summary Using Key Concepts and Sentence Importance

Input:
S: a list of sentences with their scores computed using Equation 4.
K: a list of keyphrases (key concepts) ranked in decreasing order of their weights, which are computed using Equation 1.
T_{pos}: the predefined threshold on sentence position.

Output: *SUMMARY*: a summary of length L (where L is specified by the user as input).

Steps

1: Initialization

SUMMARY = empty, *KPcoverage* = empty, *p* = 1

2: Summary Generation

2.1 Phase 1

While *p* ⟨=|*K*| and |*SUMMARY*| ⟨= *L*

- Choose the *p-th* member from the ranked phrase list *K* and assign it to the temporary variable *KP*, that is, *KP* = *K*[*p*].
- If the keyphrase stored in *KP* is not present in the list *KPcoverage*, pool all the sentences containing the keyphrase *KP*.
- Discard from the pool the sentences whose positional value is ⟩ T_{pos}.
- Choose the most important sentence *s'* from the pool and add s' to the SUMMARY, that is, SUMMARY = SUMMARY ∪ *s'*.
- Extract the candidate kephrases *k'* from *s'* and update the list *KPcoverage as: KPcoverage* = *KPcoverage* ∪ *k'*.
- p= p+1 (advance pointer to point to the next member in *K*).

end while

2.2 Phase 2

If |*SUMMARY*| ⟨ L

Arrange the sentences in *S* in the decreasing order of their scores and form a list named *RankedSentenceList*.

i = *1*
While |*SUMMARY*| ⟨= *L*

- Choose a sentence *s'* from the RankedSentenceList, i.e., *s'* = *RankedSentenceList*[*i*].
- If *s'* does not belong to SUMMARY, then SUMMARY = SUMMARY ∪ *s'*.
- i= i + 1.

end while
end if

Fig. 3. Algorithm 1: Creating a Summary Using Key Concepts and Sentence Importance

4. EVALUATION, EXPERIMENTS AND RESULTS

4.1 Evaluation setup

The generic single document summarization task (Task 1) was addressed at the DUC 2001 (Document Understanding Conference 2001) and at the DUC 2002, with a target summary length of 100 words or less. The baseline (called the *lead* baseline) both years, with summarizer code 1, was the same. It *took the first n words of the input document*. Out of the DUC conferences organized till present, only the DUC 2001 and the DUC 2002 considered single document summarization tasks. As such, the data for these two years is only available for experimentation on single document summarization.

For tuning the parameters of the proposed summarization system, we used the DUC 2001 dataset that contains 309 English news articles, which were categorized into 30 clusters. Each cluster contains a set of documents related to a topic. We tested the proposed system on Task 1 of the DUC 2002 with a dataset containing 567 English news articles, which were categorized into 59 news clusters, and where each cluster contained a set of news articles related to a topic.

For these two datasets, the news articles were collected from TREC-9 for the single document summarization task. Table 1 shows a brief description of the two datasets used in training and testing our proposed system.

For summary evaluation, we have used the commonly used automatic evaluation tool called, the ROUGE package, which was developed by Lin [31]. ROUGE is based on the n-gram overlap between a system generated summary and a set of reference summaries [32]. It measures a summary quality by counting overlapping units, such as the word *n*-gram, word sequences, and word pairs between the candidate summary and the reference summaries.

The ROUGE-N recall score is computed as follows:

$$ROUGE-N = \frac{\sum_{S \in (\text{Reference Summaries})} \sum_{gram_n \in S} \text{Count}_{\text{match}}(gram_n)}{\sum_{S \in (\text{Reference Summaries})} \sum_{gram_n \in S} \text{Count}(gram_n)}$$

where ROUGE-N is an *n*-gram recall between a system generated summary and a set of reference summaries. *n* stands for the length of the *n*-gram, $gram_n$ and $Count_{match}(gram_n)$ are the maximum number of *n*-grams co-occurring in a system generated summary and a set of reference summaries.

The older versions of the ROUGE package, such as Versions 1.1, 1.2, 1.2.1, and 1.4.2, only used a recall based score for summary evaluation. Whereas, the newer version of the ROUGE

Table 1. A brief description of datasets

	DUC 2001	DUC 2002
Task	Task 1	Task 1
Total number of documents	309	567
Number of clusters	30	59
Data sources	TREC-9	TREC-9
Summary length	100 words	100 words

package--ROUGE 1.5.5--evaluates summaries based on three metrics such as ROUGE-N precision, ROUGE-N recall, and the ROUGE-N F-Score, where N can be 1, 2, 3, 4 etc. Thus, the ROUGE toolkit reports separate scores for 1, 2, 3, and 4-grams, and also for the skip bigram. We have used ROUGE Version 1.5.5 for our system evaluation.

Among the various ROUGE scores, the unigram-based ROUGE score (ROUGE-1) has been shown to most agree with human judgment [32]. We have showed three of the ROUGE metrics in the experimental results: ROUGE-1 (unigram-based), ROUGE-2 (bigram-based), and ROUGE-SU4 (matching bigrams with a skip distance of up to 4 words [31]).

4.2 System performance comparison

Our proposed approach is trained on the DUC 2001 dataset and was tested on the generic single document summarization task of the DUC 2002. The summary length was set to 100 words. The proposed approach was compared with the LEAD baseline method, which considers the first 100 words of a document as a summary.

Previous research works [33] concluded that, for a generic single document summarization task, the *lead* baseline was quite strong and that it was difficult to outperform this baseline due to the journalistic convention of putting the most important part of an article in the initial paragraphs. But the fact that the most human summarizers perform significantly better than the baseline shows that the task is meaningful and that better-than-baseline performance is possible [33].

The results shown in Table 2 were obtained after testing our proposed approach on DUC 2002 data and by evaluating the system generated summaries using ROUGE 1.5.5 with the settings that apply stemming (stop-words were not removed) while computing the ROUGE score. The results shown in Table 3 were obtained after evaluating the system generated summaries using ROUGE 1.5.5 with the setting that applies both the stemmer and stop-word remover while computing ROUGE scores. These tables also show the comparisons of our proposed summarization approach and the DUC 2002 baseline. In these tables, we have also shown within square brackets the 95% confidence interval reported by the ROUGE toolkit for the corresponding ROUGE scores.

Table 2. ROUGE scores for the proposed system and the baseline system on the DUC 2002 data where the summaries are stemmed but where the stop-words are not removed. 95% confidence intervals are shown in brackets.

Systems	ROUGE-1 F-Score	ROUGE-2 F-Score	ROUGE-SU4 F-Score
Proposed system	**0.4855** [0.4783 - 0.4925]	**0.2304** [0.2222 - 0.2393]	**0.2471** [0.2399 - 0.2547]
DUC baseline	0.4751 [0.4679 - 0.4824]	0.2240 [0.2160 - 0.2321]	0.2408 [0.2338 - 0.2480]

Table 3. ROUGE scores for the proposed system and the baseline system on the DUC 2002 data where the summaries are stemmed and where the stop-words are removed. 95% confidence intervals are shown in brackets

Systems	ROUGE-1 F-Score	ROUGE-2 F-Score	ROUGE-SU4 F-Score
Proposed system	**0.4308** [0.4220 - 0.4395]	**0.2176** [0.2095 - 0.2263]	**0.2174** [0.2104 - 0.2249]
DUC baseline	0.4182 [0.4094 - 0.4271]	0.2138 [0.2055 - 0.2222]	0.2117 [0.2047 - 0.2189]

As can be seen from Tables 2 and 3, our proposed summarization approach outperforms the DUC baseline. This shows that the use of key concepts in single document summarization improves summarization performance.

4.3 Comparison with DUC systems

We compared our proposed method with five typical participating systems on the DUC 2002. The typical systems are the systems with high ROUGE scores. They were chosen from the participating systems on the single document summarization task of the DUC 2002. Table 4 gives a brief description of the five systems.

To compare our proposed method with the others mentioned above, we used the summaries released by DUC officials on the web. Table 5 shows the comparisons of our proposed summarization approach and the top five systems that participated in the DUC 2002.

In the table, the top five systems that participated in the DUC 2002 are indicated by Sys 28, Sys 21, Sys 29, Sys 27, and Sys 31 where 28, 21, 29, 27, and 31 are the system codes that were assigned to them during their evaluations at the DUC 2002.

Table 5 shows the results that were obtained after evaluating the systems using ROUGE 1.5.5 with a setting that applies a stemmer (but no stop-word removal) while computing the ROUGE scores. Though Table 6 also shows the comparisons of our proposed summarization system and the five top systems participating in DUC 2002, the ROUGE scores shown in this table have been computed with a different setting that applies a stemmer along with a stop-word remover on the summaries. Whereas, the ROUGE scores shown in Table 5 only use stemming without removing stop-words from the summaries.

As can be seen in Tables 5 and 6, our proposed summarization approach performs well and the performance of the approach is comparable to and sometimes better than a number of systems that were reported to be performing the best on the DUC 2002 dataset. Compared to the best system (System Code 28 shown in Tables 5-6) that uses the Hidden Markov Model (HMM) and the LRM (logistic regression model) for sentence extraction, our proposed method is also relatively simple and easily implementable.

Table 4. A brief description of top five systems participating on the DUC 2002

System ID	System Code	Group	System Description
Ccsnsa.v2	28	CCS-NSA	Supervised sentence extraction with the Hidden Markov Model (HMM) and the LRM (Logistic Regression Model)
Wpdv-xtr.v1	21	Catholic Univ. Nijmegen	Supervised sentence classification with WPDV (Weighted Probability Distribution Voting)
ULeth 131m	31	Univ. of Lethbridge	Unsupervised sentence extraction + text segmentation
Kul.2002	29	Catholic Univ. Leuven	Unsupervised sentence extraction + topic segmentation
Ntt.duc02	27	NTT	Supervised sentence classification with SVM (Support Vector Machines)

Table 5. System comparison results on the DUC 2002 data with a 95% confidence interval. For summary evaluation, summaries are stemmed, but the stop-words were not removed

Systems	ROUGE-1 F-score	ROUGE-2 F-score	ROUGE-SU4 F-score
Our Proposed System	**0.4855** [0.4783 - 0.4925]	**0.2304** [0.2222 - 0.2393]	**0.2471** [0.2399-0.2547]
Sys 28	0.4830 [0.4757-0.4898]	0.2297 [0.2217-0.2382]	0.2461 [0.2392 - 0.2533]
Sys 21	0.4757 [0.4688 - 0.4829]	0.2218 [0.2140-0.2290]	0.2390 [0.2322-0.2458]
DUC baseline	0.4751 [0.4679-0.4824]	0.2240 [0.2161-0.2321]	0.2408 [0.2338-0.2480]
Sys 29	0.4685 [0.4616 - 0.4758]	0.2147 [0.2071-0.2230]	0.2332 [0.2264 - 0.2403]
Sys 27	0.4651 [0.4576- 0.4728]	0.2155 [0.2076 - 0.2242]	0.2331 [0.2264- 0.2407]
Sys 31	0.4599 [0.4528- 0.4664]	0.2018 [0.1932 - 0.2100]	0.2239 [0.2165- 0.2310]

Table 6. System comparison results on the DUC 2002 data with a 95% confidence interval. For summary evaluation, the summaries are stemmed and the stop-words were removed.

Systems	ROUGE-1 F-score	ROUGE-2 F-score	ROUGE-SU4 F-score
Our Proposed System	**0.4308** [0.4220-0.4395]	**0.2176** [0.2095-0.2263]	**0.2174** [0.2104-0.2249]
Sys 28	0.4281 [0.4193-0.4368]	0.2177 [0.2094-0.2263]	0.2172 [0.2101-0.2244]
DUC baseline	0.4182 [0.4094-0.4271]	0.2138 [0.2055-0.2222	0.2117 [0.2047-0.2189]
Sys 21	0.4155 [0.4064-0.4246]	0.2103 [0.2022-0.2184]	0.2085 [0.2014-0.2156]
Sys29	0.4077 [0.3990-0.4163]	0.2044 [0.1964-0.2121]	0.2033 [0.1964-0.2106]
Sys 27	0.4069 [0.3972-0.4165]	0.2031 [0.1952-0.2117]	0.2034 [0.1966-0.2110]
Sys31	0.3922 [0.3828-0.4012]	0.1890 [0.1801-0.1971]	0.1912 [0.1834-0.1986]

4.4 Comparison with the post-DUC systems

We also compared our proposed summarization approach to some selected approaches, which were tested on Task 1 of the DUC 2002 and whose results were published after the conference. Since we have compared our proposed approach to the approaches that used the older version of the ROUGE package for summary evaluation, we only considered the recall scores obtained by our summarization system for comparisons with other existing approaches.

The *Manual* (readme file) available with ROUGE Package, version 1.5.5 suggests that only the recall scores should be considered when comparing any system's performance with the previous DUC results. The main reason for using a recall score in this context is that the previous DUC results were obtained through summary evaluation by using older versions of the ROUGE pack-

age that only gives the recall score for summary evaluation. Precision has been incorporated into the later version of the ROUGE package, because the recall score by itself is not sufficient in the situation where there is a possibility of improving recall by sacrificing precision.

The first approach to which our proposed summarization approach is compared was proposed by Wan et al. and published very recently in [28]. The approach proposed by Wan et al. [28] exploits the neighborhood knowledge, which has been constructed with a small number of nearest/neighbor documents that are close to the specified document, for improving document summarization. It does so under the assumption that the neighbor documents could provide additional knowledge and more clues. Here the specified document is actually expanded to a small document set by adding a few neighbor documents close to the document. Then the graph-based ranking algorithm is applied to the expanded document set to make use of both the local information in the specified document and the global information in the neighbor documents. This approach has been tested on the DUC 2002 data set and the older version of ROUGE toolkit has been used by Wan et al. for evaluating the system generated summaries. The authors have experimentally shown that the neighborhood knowledge is useful for document summarization. Table 7 compares our proposed summarization approach with the summarization approach presented in [28] in terms of ROUGE-1, ROUGE-2, and ROUGE-W recall scores. The ROUGE scores for the DUC 2002 data set, reported in [28] by the authors of the article as the evaluation results for their proposed summarization approach, has been shown in Row 2 of Table 7. Row 1 of Table 7 shows the results that were obtained after we evaluated our proposed summarization approach using ROUGE Version 1.5.5, which is run with similar command-line options used for system evaluation by Wan et al. That is, the summaries are stemmed but the stop-words are not removed during the system evaluation. The results presented in Table 7 show that our proposed summarization approach performs better than the approach proposed in [28] by Wan et al. in terms of ROUGE-1, ROUGE-2, and ROUGE-W recall scores.

Steinberger et al. (2006) in [29] proposed the second approach to which our proposed approach has been compared. Their approach uses an anaphora resolution for improving text summarization performance. The authors developed an LSA (Latent Semantic Analysis)-based summarizer that uses an approach to single document text summarization, which incorporates anaphoric information in Latent Semantic Analysis (LSA).

Steinberger et al. (2006)[29] evaluated summaries using an older version of the ROUGE package with the settings: ROUGEeval-1.4.2.pl -c 95 -m -n 2 -s -2 4 –a. According to this setting, summaries are stemmed and stop words are removed to conduct summary evaluation. For evaluating the summaries generated by our proposed summarization approach, we used ROUGE version 1.5.5 and ran it with similar settings. That is, we used: ROUGE-1.5.5.pl -c 95 -m -n 2 -s -2 4 –a. We considered ROUGE-1, ROUGE-2, and ROUGE-L recall scores for comparisons of our proposed summarization approach with the approach proposed in [29] by Steinberger et al.

Table 7. Comparisons of our proposed approach and the approach proposed by Wan et al. (2010) in [28]. The summaries are stemmed but the stop-words were not removed for summary evaluation

	ROUGE-1 recall	ROUGE-2 recall	ROUGE-w recall
Our proposed summarization approach evaluated using ROUGE 1.5.5	**0.4879**	**0.2316**	**0.1711**
Approach proposed in [28] by Wan et al. (2010)	0.47162	0.20114	0.16314

Table 8. Comparisons of our proposed summarization approach and the approach proposed by Steinberger et al.(2006)[29]. The summaries are stemmed and the stop words were removed for summary evaluation.

	ROUGE-1 recall	ROUGE-2 recall	ROUGE-L recall
Our proposed summarization approach evaluated using ROUGE 1.5.5	**0.4333**	**0.2191**	**0.3967**
The summarization approach by Steinberger et al. (2006) [29]	0.4228	0.2074	0.3928

(2006). The ROUGE scores for the DUC 2002 data set, reported in [29] by the authors of the article as the evaluation results for their proposed summarization approach, has been shown in Row 2 of Table 8. As can be seen in Table 8, our proposed summarization approach performs better than the approach proposed in [29] by Steinberger et al.

5. CONCLUSION

This paper discusses a single document text summarization method that uses key concepts extracted from a document for summary generation. The comparisons of our proposed summarization approach to some state-of-the art summarization approaches suggest that our proposed summarization approach performs well and that the performance of our approach is comparable to and sometimes better than a number of systems that have been reported to perform best on the DUC 2002 data set. Our approach is also simple and easily implementable compared to some state-of-the art summarization approaches.

Since our proposed summarization approach depends on key concepts in documents, we may expect that an improvement to the keyphrase extraction method will enhance summarization performance. However, at the same time, we should try to avoid using a sophisticated keyphrase extraction method because it may degrade the speed of the summarization system.

REFERENCES

[1] I. Mani, "Automatic summarization," Vol. 3, Amsterdam/Philadelphia: John Benjamins Publishing Company, 2001.

[2] D. R. Radev, H. Jing, M. Sty and D. Tam, "Centroid-based summarization of multiple documents," Journal of Information Processing and Management, Elsevier, Volume 40, no. 6, 2004, pp. 919-938.

[3] P. D. Turney, "Learning algorithm for keyphrase extraction," Journal of Information Retrieval, vol. 2, no. 4, 2000, pp. 303-36.

[4] I. H. Witten, G.W. Paynter, E. Frank, C. Gutwin and C. G. Nevill-Manning, "KEA: Practical Automatic Keyphrase Extraction," Proceedings of Digital Libraries'99: The Fourth ACM Conference on Digital Libraries, ACM Press, Berkeley, CA, 1999, pp. 254 – 255.

[5] K.Sarkar, M.Nasipuri and S.Ghose, "Machine Learning Based Keyphrase Extraction: Comparing Decision Trees, Naïve Bayes and Artificial Neural Networks," Int J Inf Process Syst, vol. 8, no. 4, 2012, pp.693-712.

[6] Y. B. Wu and Q. Li, "Document keyphrases as subject metadata: incorporating document key concepts in search results," Journal of Information Retrieval, vol. 11, no. 3, 2008, pp. 229-249.

[7] P. Baxendale, "Man-made index for technical literature-An experiment," IBM Journal of Research and Development, vol. 2, no. 4, 1958, pp. 354 – 361.

[8] H. P. Luhn, "The automatic creation of literature abstracts," IBM Journal of Research Development, vol. 2, no. 2, 1958, pp.159–165.

[9] H. P. Edmundson, "New methods in automatic extracting," Journal of the Association for Computing Machinery, vol. 16, no. 2, 1969, pp.264–285.

[10] K. Sarkar, "An approach to summarizing Bengali news documents," Proceedings of the International Conference on Advances in Computing, Communications and Informatics, ACM, 2012, pp. 857-862.

[11] K. Sarkar, "Bengali text summarization by sentence extraction," Proceedings of International Conference on Business and Information Management, NIT Durgapur, 2012, pp. 233-245.

[12] K. Sarkar, M. Nasipuri, S. Ghose, "Using Machine Learning for Medical Document Summarization," International Journal of Database Theory and Application, Vol. 4, no. 1, pp. 31- 48, 2010.

[13] G. Salton, A. Singhal, M. Mitra and C. Buckley, "Automatic text structuring and summary," Journal of Information Processing and Management, vol. 33, no. 2, 1997, pp. 193-207.

[14] Y. Ledeneva, R. G. Hernández, R. M. Soto, R. C. Reyes and A. Gelbukh, "EM clustering algorithm for automatic text summarization," In Advances in Artificial Intelligence, Springer Berlin Heidelberg, 2011, pp. 305-315.

[15] R. Barzilay and M. Elhadad, "Using Lexical Chains for Text Summarization," Proceedings of the Workshop on Intelligent Scalable Text Summarization, Madrid, Spain, 1997, pp. 10--17.

[16] M. A. K Halliday and R. Hasan, "Cohesion in English," English Language Series, Longman, London, 1976.

[17] J. Morris and G. Hirst, "Lexical cohesion computed by thesaural relations as an indicator of the structure of text," Computational Linguistics, vol. 17, no. 1, 1991, pp. 21-43.

[18] R. A. García-Hernández and Y. Ledeneva, "Single Extractive Text Summarization Based on a Genetic Algorithm," In Pattern Recognition, Springer Berlin Heidelberg, 2013, pp. 374-383.

[19] J. M. Conroy and D. P. O'Leary, "Text summarization via hidden Markov models and pivoted QR matrix decomposition," Tech. Rep., University of Maryland, College Park, 2001.

[20] M. Osborne, "Using maximum entropy for sentence extraction," Proceedings of the ACL-02, Proceedings of Workshop on Automatic Summarization, (Philadelphia, Pennsylvania), Annual Meeting of the ACL, Association for Computational Linguistics, Morristown, vol. 4, 2002.

[21] C. D. Paice and P.A. Jones, "The identification of important concepts in highly structured technical papers," Proceedings of the 16th International Conference on Research and Development in Information Retrieval (SIGIR '93), 1993, pp. 69-78.

[22] H. Jing and K. McKeown, "The decomposition of human-written summary sentences," Proceedings of SIGIR '99: 22nd International Conference on Research and Development in Information Retrieval, University of California, Berkeley, August, 1999, pp. 129–136.

[23] H. Jing, "Using hidden Markov modeling to decompose human-written summaries," Computational Linguistics, vol. 28, no. 4, 2002, pp. 527–543.

[24] M. Banko, V. Mittal and M. Witbrock, "Headline generation based on statistical Translation," Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000), Hong Kong, 2000, pp. 318–325.

[25] D. Zajic, B. Dorr and R. Schwartz, "Automatic Headline Generation for Newspaper Stories," Workshop on Automatic Summarization, Philadelphia, PA, 2002, pp. 78-85.

[26] B. J. Dorr, D. Zajic and R. Schwartz, "Hedge trimmer: A parse-and-trim approach to headline generation," Proceedings of the HLT/NAACL 2003 Text Summarization Workshop and Document Understanding Conference (DUC 2003), Edmonton, Alberta, 2003, pp. 1-8.

[27] D. Zajic, B. J. Dorr and R. Schwartz, "BBN/UMD at DUC-2004: Topiary," Proceedings of the North American Chapter of the Association for Computational Linguistics, Workshop on Document Understanding, Boston, MA, 2004, pp. 112-119.

[28] X. Wan and J. Xiao, "Exploiting Neighborhood Knowledge for Single Document Summarization and Keyphrase Extraction," ACM Transactions on Information Systems, vol. 28, no. 2, Article 8, 2010, pp. 8:1-8:34.

[29] J. Steinberger, M. Poesio, M.A. Kabadjov and K. Ježek, "Two uses of anaphora resolution in summarization," Information Processing & Management, vol. 43, no. 6, 2007, pp. 1663-1680.

[30] S. Elbeltagy and A. Rafea, "Kp-miner: A keyphrase extraction system for English and Arabic documents," Information Systems, vol. 34, no. 1, 2009, pp. 132–144.

[31] C. Y. Lin, "ROUGE: A package for automatic evaluation of summaries," Proceedings of the Workshop on Text Summarization Branches Out, July 25–26, Barcelona, Spain, 2004.

[32] C.-Y. Lin and E. Hovy, "Automatic evaluation of summaries using n-gram co-occurrence," Proceedings of 2003 Language Technology Conference (HLT-NAACL 2003), Edmonton, Canada, 2003.

[33] A. Nenkova, "Automatic Text Summarization of Newswire: Lessons Learned from the Document Understanding Conference," AAAI, 2005.

Kamal Sarkar

He received his B.E degree in Computer Science and Engineering from the Faculty of Engineering, Jadavpur University in 1996. He received the M.E degree and Ph.D. (Engg) in Computer Science and Engineering from the same University in 1999 and 2011 respectively. In 2001, he joined as a lecturer in the Department of Computer Science & Engineering, Jadavpur University, Kolkata, where he is currently an associate professor. His research interest includes text summarization, natural language processing, machine learning, web mining, knowledge discovery from text data.