American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN : 2320-0936
Volume-6, Issue-1, pp-226-239
www.ajer.org
Research Paper Open Access

A Comprehensive Survey on Extractive Text Summarization Techniques

Aysa Siddika Asa[1], Sumya Akter[2], Md. Palash Uddin[3], Md. Delowar Hossain[4], Shikhor Kumer Roy[5], Masud Ibn Afjal[6]

[3,4,6](Department of Computer Science and Engineering, Hajee Mohammad Danesh Science and Technology University (HSTU), Dinajpur-5200, Bangladesh)

[1,2,5](B.Sc. in Computer Science and Engineering, Hajee Mohammad Danesh Science and Technology University (HSTU), Dinajpur-5200, Bangladesh)

ABSTRACT: *Automated data collection tools and matured database technology lead to tremendous amounts of data stored in database, data warehouses and other data repositories. With the increasing amount of online information, it becomes extremely difficult to find relevant information to users. Information retrieval system usually returns a large amount of documents listed in the order of estimated relevance. It is not possible for users to read each document in order to find the useful one. Automatic text summarization system, one of the special data mining applications, helps in this task by providing a quick summary of the information contained in the document(s). Some efficient work has been done for text summarization on various languages. But among them there are a few works on Bengali language. It has thus motivated us to do develop or modify a new or existing summarization technique for Bengali document(s) and to provide us an opportunity to make some contribution in natural language processing. To do the same, we have surveyed and compared some techniques on extractive text summarization on various languages in this paper. The summarizations have done for single or multiple documents in different languages. Finally, a comparative nomenclature on the discussed single or multi-document summarization techniques has been conducted.*

Keywords: *big data, data mining, extractive summarization, text mining, text summarization*

I. INTRODUCTION

Nowadays, there exist a lot of amount of data and this rapid growth of data is required to process, store, and manage. Sometimes, it is difficult to find the exact information from large amount of stored data or big data. Today, in the era of big data, textual data is rapidly growing and is available in many different languages. Big data has the potential to be mined for information and data mining is essential to find out the proper information what we need [1]. Search engines such as Google, AltaVista, Yahoo, etc., have been developed to retrieve specific information from this huge amount of data. But most of the time, the outcome of search engine is unable to provide expected result as the quantity of information is increasing enormously day by day and also the findings are abundant [2]. Knowledge discovery (e.g. text mining) from large volumes of data has seen sustained research in recent years. As a field of data mining, text summarization is one of the most popular research areas to extract main theme from large volume of data. This process reduces the problem of information overload because only a summary needs to be read instead of reading the entire document. This can comprehensively help the user to make out ideal documents within a short time by providing scraps of information [3].

1.1 Data Mining

Data mining is a very growing application field for the researchers. It is the process of extracting some meaningful information from chunks of meaningless data whereas text mining is about looking for pattern in text. The information overload problem leads to wastage of time for browsing all the retrieval information and there may have a chance to miss out relevant information [4]. The roots of data mining are traced back along with three family lines: classical statistics, artificial intelligence, and machine learning [5], [6]. Typical data mining tasks include document classification, document clustering, building ontology, sentiment analysis, document summarization, information extraction etc.Data mining utilizes descriptive and predictive approaches in order to discover hidden information. Data mining satisfies its main goal by identifying valid, potentially useful, and easily understandable correlations and patterns present in existing data. This goal of data mining can be satisfied by modeling it as either predictive or descriptive nature [7]. Predictive approaches include classification, regression or prediction, time series analysis etc. whereas descriptive approaches include clustering, summarization, association rules, sequence discovery etc.

1.2 Data Mining Algorithms and Applications

Data mining uses different techniques such as statistical, mathematical, artificial intelligence and machine learning as the computing techniques [7]. The techniques and algorithms for data mining are Naive Bayes decision theory, support vector machine (SVM), decision tree etc. for classification or logistic regression; multiple regression, SVM etc. for regression; minimum description length for attribute importance;one-class SVM for anomaly detection; orthogonal partitioning clustering, expectation maximization(EM) algorithm, K-means algorithm, enhanced K-means etc. for clustering;Apriori for Association;singular vector decomposition (SVD), principal components analysis (PCA), non-negative matrix factorization etc. for feature selection and extractionand so on [8].

As the importance of data analysis continues to grow, the companies are finding more and more applications for data mining and business intelligence. There are a number of commercial data mining systems available today and yet there are many challenges in this field. The applications include financial data analysis, retail industry, telecommunication industry, biological data analysis and other scientific applications such as data warehouses and data preprocessing, graph-based mining, visualization and domain specific knowledge etc. [9].

1.3Comparative Statement of Data Mining

The following table presents the comparative statement of various data mining trends from past to the future

Table 1: Data mining trends comparative statement[10], [27]

	Algorithms/ Techniques employed	**Data formats**	**Computing Resources**	**Prime areas of applications**
Past	statistical, machine learning techniques	numerical data and structured data stored in traditional databases	evolution of 4G programming language and various related techniques	business
Current	statistical, machine learning, artificial intelligence, pattern reorganization techniques	heterogeneous data formats includes structured, semi structured and unstructured data	high speed networks, high end storage devices and parallel, distributed computing etc.	business, web, medical diagnosis etc.
Future	soft computing techniques like fuzzy logic, neural networks and genetic programming	complex data objects i.e., high dimensional, high speed data streams, sequence, noise in the time series graph, multi instance objects, multi represented objects, temporal data	multi-agent technologies and cloud computing	business, web, medical diagnosis, scientific and research analysis fields (e.g., remote sensing), social networking etc.

1.4Text Summarization

Text Summarization aims to generate concise and compressed form of original documents. With text mining, the information to be extracted is clearly and explicitly stated in the text. Text mining summarizes salient features from a large body of text, which is a subfield of text summarization [12]. Summarization can be classified into two main categories i.e. extractive and abstractive summarization. Both techniques are used for summarizing text either for single document or for multi-documents. Extractive summarization involves assigning saliency measure to some units (e.g. sentences, paragraphs) of the documents and extracting those with highest scorestoinclude in the summary. Abstractive summarization usually needs information fusion, sentence compression and reformulation. It is complex because it requires deeper analysis of source documents and concept-to-text generation [13].

1.4.1 Abstractive Summarization Techniques

Abstractive summarization methods consist of understanding the original text and re-telling it in fewer words. It uses linguistic methods to examine and interpret the text and then to find the new concepts and expressions to best describe it by generating a new shorter text that conveys the most important information from the original text document [14]. Abstractive summarization is classified into two categories structured based (Rule based method, tree based method, ontology method etc.) and semantic based (Multimodal semantic model, information item based method, semantic graph based method etc.) methods [15].

1.4.2 Extractive Summarization Techniques

Extractive summarizers find out the most relevant sentences in the document. It also avoids the redundant data. It is easier than abstractive summarizer to bring out the summary.The common methods for extractive are Term Frequency/Inverse Document Frequency (TF/IDF) method, cluster based method, graph theoretic approach, machine

learning approach, LSA Latent Semantic Analysis (LSA) method, artificial neural networks, fuzzy logic, query based, concept-obtained text summarization, using regression for estimating feature weights, multilingual, topic-driven summarization, Maximal Marginal Relevance (MMR), centroid-based summarization etc. A general procedure for extractive methods involves three steps to be performed which are discussed below [11], [15], [16].

Step 1: First step creates a representation of the document. Some preprocessing such as tokenization, stop word removal, noise removal, stemming, sentence splitting, frequency computation etc. is applied here.

Step 2: In this step, sentence scoring are performed. In general, three approaches are followed:

- Word scoring–assigning scores to the most important words
- Sentence scoring–verifying sentences features such as its position in the document, similarity to the title etc.
- Graph scoring–analyzing the relationship between sentences

The general methods for calculating the score of any word are word frequency, TF/IDF, upper case, proper noun, word co-occurrence, lexical similarity, etc.The common phenomena used for scoring any sentences are Cue-phrases (‘‘in summary’’, ‘‘in conclusion’’, ‘‘our investigation’’, ‘‘the paper describes’’ and emphasizes such as ‘‘the best’’, ‘‘the most important’’, ‘‘according to the study’’, ‘‘significantly’’, ‘‘important’’, ‘‘in particular’’, ‘‘hardly’’, ‘‘impossible’), sentence inclusion of numerical data, sentence length, sentence centrality, sentence resemblance to the title, etc.Also the popular graph scoring methods are text rank, bushy path of the node, aggregate similarity etc.

Step 3: In this step, high score sentences using a specific sorting order for extracting the contents are selected and then the final summary is generated if it is a single document summarization. For multi-document summarization, the process needs to extend. Each document produces one summary and then any clustering algorithm is applied to cluster the relevant sentences of each summary to generate the final summary.

II. REVIEWED ARTICLES

The previous works on single document or multi-document summarization are trying in different directions to show the best result. Till now various generic multi-document extraction-based summarization techniques are already presented. Most of them are on English rather than on other languages like Bengali. In this section, we discussed some previous works on extractive text summarization.

2.1 Paper I

J. Zhang *et. al.*presented a cue-based hub-authority approach for multi-document text summarization [17]. It involves thefollowing procedure:

a) Detecting sub-topics by sentence clustering using KNN
b) Extracting the feature words (or phrase) of different sub-topics using TF*IDF
c) Detecting vertex
i. Hub vertex (All feature words, Cue phrase)
ii. Authority vertex (All sentences are regarded)
d) If the sentence contains the words in the Hub, there is an edge between the Hub word & the Authority sentence
e) The initial weight of each vertex considers both the content & the Cues such as Cue phrase & first sentence
f) Final summarization finds to order the sub-topics using Markov Models

Sub-topic detection:

- Initialize the set of sub topics by partitioning all the sentences from the documents into clusters. Firstly create *K* clusters via KNN
- Measure sentence similarity by the cosine metric using words contained in the sentences as features
- Extract the feature words using TF*IDF

Ranking sentences by computing Hubs & Authority:

If the sentences contain the words in Hub, there is a directed edge pointing from Hub word to Authority sentence. Let, consider a non-negative Hub weight x (h_i) & a non-negative Authority weight y (a_j)

The weight of each type is normalized so their squares sum to 1:

$$\sum_{v \in Hub} x^2(h_v) = 1 \qquad (1)$$

$$\sum_{v \in Authority} y^2(a_v) = 1 \qquad (2)$$

Given weights $\{x(h_i)\}, \{y(a_j)\}$,the *I* operation updates the *x*-weights as follows:

$$x(h_i) \leftarrow \sum_{aj:(hi,aj) \in E} y(a_j) \qquad (3)$$

The *O* operation updates the *y*-weights as follows:

$$y(a_j) \leftarrow \sum_{aj:(hi,aj) \in E} x(h_{ii}) \qquad (4)$$

Here, the Hub vertex h_i points to many sentences with large *y*-values. And the Authority vertexa_j is pointed by many words with large *x*-values.

For computing the hubs & authorities weights:
Step i. Let *z*=initial Hub weight vector. If Hub word is Cue word, then *z*=2; *z*=1, otherwise.
Step ii. Let *w*=initial Authority weight vector. If Authority sentence is the first sentence, then *w*=2; *w*=1, otherwise.
Step iii. $x_0 = z \& y_0 = w$
Step iv: For $i = 1,2,3, \ldots \ldots \ldots k$

- Apply the *I* operation, obtaining new *x*-weight, $x_i^{'}$
- Apply the *O* operation, obtaining new *y*-weight $y_i^{'}$
- Normalize $x_i^{'} \& y_i^{'}$ to obtain $x_i \& y_i$
- END

Step v. Return $x_{k\&} y_k$
The weight in authority vector y_k can be regarded as the ranking score of the different sentences to be included in the summarization.

Summarization generation:

Selected sentences based upon sub-topic are organized. Here, each sub-topic means all the sentences within one cluster.Suppose, there are *m* different sub-topics i.e., $T = \{T_1, T_2, \ldots\ldots\ldots T_i \ldots\ldots\ldots T_m\}$
Given a document, $d = \{S_1, S_2, \ldots \ldots S_j \ldots \ldots S_n\}$ with *n* sentences.
$T_i = \{S_1^{'}, S_2^{'} \ldots \ldots S_j^{'} \ldots \ldots \ldots \ldots S_l^{'}\}$ any clusters.

So, $I = d \cap T_i = \{S_1^{''} S_2^{''} \ldots \ldots S_j^{''} \ldots \ldots \ldots \ldots S_l^{''}\}$

Topic order of multi-document is Markov Model:

$$TopicOrder(D) = ST_1T_2 \ldots\ldots\ldots T_nE$$

Here,*D*=TopicOrder of *D* document, *S*=the starting state and *E*=the ending state.The state transition possibility is calculated as follows:

$$P(T_j|T_i) = \frac{P(T_i|T_j)}{P(T_i)} \qquad (5)$$

$$Argmax\ P(ST_{k1}T_{k2} \ldots\ldots\ldots T_{kn}E) = P(S)P(T_{k1}|S)P(T_{k2}|T_{k1}) \ldots\ldots P(E|T_{km})$$

Here, $P(ST_{k1}T_{k2} \ldots\ldots\ldots T_{kn}E)$=any paths from starting state to the ending state and *P(S)*=1. Then, the sentence with the maximum ranking score within each topic is selected as thesummarization.

2.2 Paper II

Y.Ouyang*et. al.* [18] presented an integrated multi-document summarization approach based on hierarchical representation. In this paper, query relevancy and topic specificity are used for filtering process. Also it calculatespoint wise mutual information (PMI) for identifying the sub-sumption between words and high PMI is regarded as related. Then hierarchical tree is constructed.
Hierarchical tree construction algorithm:
Step i. Preprocessing includes tokenization, stop word removal, stemming and giving frequency in each word
Step ii. Sort the identified key words by their frequency in the document set in decreasing order such as $T = \{t_1, t_2, t_3, \ldots\ldots\ldots t_n\}$
Step iii. For each word t_i, *i* from 1 to *n* find the most relevant word t_j from all the words before t_i in *T* as $Ti = \{t_1, t_2, t_3, \ldots\ldots\ldots t_{i-1}\}$. Here, the relevancy of two words is calculated as:

$$\text{PMI}(t_i, t_j) = \log \frac{freq(t_i, t_j) * N}{freq(t_i) freq(t_j)} \qquad (6)$$

Here,*N*= Total number of tokens in the document set, $freq(t_i) = freq(t_j)$ =frequency of $t_i and t_j$, word, $freq(t_i, t_j)$=The co-occurrence of t_i and t_j in same sentences and if the coverage rate of word t_i by word t_j is

$$\text{P}(t_i|t_j) = \frac{freq(t_i,t_j)}{freq(t_i)} \geq 0.2 \qquad (7),$$

$t_i$is regarded as being subsumed by t_j.
Step iv: If all sub-sumption relations are found, the tree is constructed by connecting the related words from the first word t_1.

Word significance estimation:

For estimating the word significance, a bottom-up algorithm is used which propagates the significance of the child nodes in tree backward to their parents. Now, the word scoring theme algorithm is:
Step i. Set the initial score of each word in *T* as its log frequency.

i.e. $Score(t_i) = \log freq_i(t_i)$

Step ii: For t_i from *n* to 1 propagate an importance score to its parent node par (t_i) [if exists] according to their relevance i.e.,

$$Score(par(t_i)) = Score(par(t_i)) + \log freq_i(t_i, par(t_i))$$

Here, $freq_i(t_i, par(t_i))$ is the frequency of t_i and parents oft_i.In this paper, another algorithm is used for the summary generation.

Sentence Selection Algorithm:

i.For the words in the hierarchical tree set, the initial states of the top *n* words are treated as activated and the states of the other words are treated as inactivated.For all the sentences in the document set, the sentence with the largest score according to the activated word set is selected. The score of a sentence *S* is defined as:

$$\text{Score}(S) = \frac{1}{|S|}\sum Score(t_i) \qquad (8)$$

Here,t_i = word belongs to *S* and states of $t_i$should be activated and $|S|$= number of words in *S*.

ii. For the selected sentence,S_k, the sub-sumption relation with the existing sentences in the current summary are calculated and the most related sentence S_l is selected.S_k is then inserted to the position behind S_l.

i. For each word,$t_i$belongs to the selected sentence S_k, its state is set to inactivated state. For each word,t_j which is subsumed by t_i, its state to set toactivated state.

ii. Repeat the process until the length limit of the summary is exceeded.

2.3 Paper III

X. Li, J. Zhang and M. Xing[19] proposed an automatic summarization technique for Chinese text based on sub topic partition and sentence features where sentence weight is calculated by LexRank algorithm combining with the score of its own features such as its length, position, cue words and structure. So, automatic summarization proposed based on maximum spanning tree and sentence features. First, partition the text document into certain number of sub topics based on maximum spanning tree. Then, the weights of the sentences in each subtopic based on LexRank algorithm combining with sentence features are computed. As a result, redundancy is reduced and summarization is extracted from each sub topic.

Sub topic partition:

It involves three steps-

- Compute similarity between any two sentences and construct fuzzy similarity matrix
- Generate a maximum spanning tree
- Partition the maximum spanning tree and get the corresponding sub topics

Fuzzy similarity matrix:

Each sentence is considered as an *m*-dimensional vector and the word weight is computed by TF*IDFalgorithm:

$$W(wi, S) = tfwi, sXidfwi \qquad (9)$$

$$idfwi = \log(\frac{N}{ni} + 1) \qquad (10)$$

Here,$tfwi$ = number of occurrence of the word wi in the sentence S,$idfwi$= inverse document frequency, *N*= total number of the sentences in the text andni= number of sentences in which word wi occurs.Then, cosine similarity between two corresponding vectors is computed and it constructs a fuzzy similarity matrix where each value is the similarity between correspondingsentence pair.

Maximum Spanning Tree generation:

To generate maximum spanning tree, they uses Kruskal's algorithm after getting the fuzzy similarity matrix.

Maximum Spanning Tree Partition:

The maximum spanning tree T is partitioned into number of sub trees according to the similarity between sentences by depth-first method.

Sentence Weight Computation:

Each sub topic is considered as a graph, vertices represent the sentences and edges define the similarity relation between pairs of sentences. The weight of each sentence is computed by LexRank score until the current score of every sentence is equal or very close to the last score:

$$p(u) = d/N(1-d)\sum\nolimits_{v\in adj[u]} \frac{w(u,v)}{\sum_{z\in adj[v]} w(z,v)} p(v) \qquad (11)$$

Here,*N*= total number of vertices in the graph, $W(u,v)$ = edge weight between vertices *u* and *v*, $Adj[u]$= set of vertices that are adjacent to the vertices u and*d*= damping factor which is typically chosen in the interval $[0.1, 0.2]$.

Sentence Weight:

Compute the feature score of a sentence Si as:

$$FeaScore(Si) = \alpha Wp(Si) + \beta Wc(Si) + \gamma Ws(Si) \qquad (12)$$

Where, α, β, γ= tune parameters and $\alpha + \beta + \gamma = 1$, $Wp(Si)$ = weight of Si based on position.

$$Wp(Si) = \begin{cases} 1, if Si is the first sentence of a paragraph \\ 0.5, if Si is the second or last sentence \\ 0, \quad otherwise. \end{cases}$$

$Wc(Si)$ = the weight of *Si* based on cue words then we get

$$Wc(Si) = \begin{cases} 1, if there is a cue word in Si \\ 0, \quad otherwise \end{cases}$$

$Ws(Si) =$ The weight of Si based on sentence structure then

$$Ws(Si) = \begin{cases} 1, if Si contains person, time, space \\ 0, otherwise \end{cases}$$

After getting the LexScore and feature score of each sub topic, then the weight of any sentence is calculated by the following formula:

$$W(Si) = \begin{cases} 0, if length(Si) \leq 6 \ Chinese characters \\ LexScore(Si) + \delta * FeaScore(Si) \end{cases}$$

Here,δ= tune parameter and it assigns the average of the similarity matrix to each sub topic.Then, certain number of sentences is extracted as summarization by compression ratio required.

2.4 Paper IV

P. Hu, T. He and H. Wang[20] proposed multi-view sentence ranking for query biased summarization. This proposed approach first constructs two base rankers to rank all the sentences in a document set from two independent but complementary views (i.e. query-dependent view and query-independent view), and then aggregates them into a consensus one.To select the most significant content from the document set with high biased information let *q* be the given query and $R1(si) \rightarrow \acute{R}$ and $R2(si) \rightarrow \acute{R}$ be two base ranker constructed based on two view *V1* and *V2* where *V1*= query dependent view and *V2*= query independent view.

i.Sentence ranking from query dependent view:

Score of each sentences *Si* is measured by the cosine relevance by the following formula:

$$Rel(S_i|q) = \frac{\overrightarrow{Si}.\vec{q}}{|\overrightarrow{Si}|.|\overrightarrow{q}|} \qquad (12)$$

$$R1(Si) = \frac{Rel(Si|q)}{\sum_{S_j \in S} Rel(Si|q)} (13)$$

Si and *q* are calculated by *tf*isf*. Frequency of the terms in those sentences is *t*. Then we get,

$$isf = 1 + \log(N/Nt) (14)$$

Here, *N* = total number of sentences in the document set and *Nt* = number of sentences in where term*t* occurs.

ii. Sentence Ranking from Query-Independent View**:**

To rank the sentences from query independent view the whole sentences consider as an undirected graph. And from this affinity graph, the cosine similarity is calculated and then the sentences query independent view is ranked using Markov chain model.

2.5 Paper V

K. Sarkar[21] applied an approach for sentence clustering based summarization for multi-documents text in which sentences are clustered using a similarity histogram based sentence-clustering algorithm to identify multiple sub-topics (themes) from the input set of related documents and it selects the representative sentences from the appropriate clusters to form the summary. Processes include preprocessing, sentence clustering, cluster ordering, representative sentence selection, summary generation.Preprocessing includes removal of stop word i.e., preposition, article, other low content words, punctuation marks except dots at the sentence boundary. Sentence clustering ensures the coherency of the clusters and minimizes inter-cluster distance. Here, similarity histogram based clustering method is used. Cosine similarity is used for measuring the similarity and each sentence is considered as a vector and similarities are calculated using the following formula:

$$Sim(Si, Sj) = (2 * |Si \cap Sj|)/(|Si| + |Sj|) \qquad (15)$$

Here, *Si*, and *Sj* are two sentences from the input document.$|Si \cap Sj|$= number of matching words between sentences. $|Si|=|Sj|$=number of words in that sentence.

The main concept for the similarity histogram-based clustering method is to keep each cluster as coherent as possible and a degree of coherency in a cluster at any time is monitored with a cluster similarity histogram.A perfect cluster histogram contains all pair wise similarities of maximum value. The right most bin represents all similarity. A loose cluster contains all pair wise similarities of minimum and the similarities would tend to be counted in the lower bins. To prevent redundancy, each sentence must keep coherent in each cluster. If the inclusion of this sentence is going to degrade thedistribution of thesimilarities in the clusters very much, it should not be added, otherwise it is added. The ratio of the count of similarities above a certain similarity threshold is calculated to the total count of similarities.

The higher this ratio, the more coherent the cluster is:

$$HR = \frac{\sum_{i=T}^{n_b} h_i}{\sum_{j=1}^{n_b} h_j} \qquad (16)$$

Here, $T=|S_T * n_b|$, T= bin number corresponding to the similarity threshold, S_T= the similarity threshold, n_b= the number of bins in a histogram andh_i =the count of sentence similarities in bin *i*.Being unsupervised the sentence clustering algorithm doesn't have any prior knowledge, so it is not perfect way to order them in their size and the following problems are found when it is done:

- Several top clusters are of equal size
- Clusters consist of a number of less informative short sentences, which increase only the size, but not the contents.

So it needs to order the clusters based on the cluster importance, which is computed by the sum of the weights of the content words of a cluster.

$$weightofaclusterC, W(C) = \sum_{w \in C} \log(1 + count(w)) \quad (17)$$

Here,*count (w)* = the count of the word *w* in the input collection and the *count (w)* is greater than a threshold. The cluster weight represents the information richness of the cluster.The log-normalized value of the total count of a word in the set of input documents has been taken as the weight of a word. Before computing the counts of the words in the input collection, all stop word includes into the summary. The selection of sentences is continued until a predefined summary size is reached.The importance of a sentence is calculated by the following formula:

$$Score(S) = \sum_{w \in S} weight(W) \quad (18)$$

$$weight(W) = \alpha 1 \log(1 + CF)\, \alpha 2 \log(1 + CF) \quad (19)$$

Where,*Score(S)* =the importance of the sentence *S*, *Weight (w)* = importance of the word *w*, is computed by taking weighted average of the local and global importance of the word*w*, *α1=α2=0.5*. After ranking sentences in the cluster based on its scores, the sentence with highest score is selected as the representative sentence.

2.6 Paper VI

A. Kogilavani *et. al.*[3] presented a multi-document summarization using clustering and feature specificsentence extraction. The following processes are used for summarizationas shown in Fig. 1:

Preprocessing:
Stop words like "a", "the", "of" are removed and words are converted into their stems using enhanced Porter Stemmer algorithm.

Documents representation and clustering:
Each term in the document can be represented using a weighting scheme called TSF-ISF.

$$Term - weight(ti) = \frac{TSF(t_i).ISF(t_i)}{n} \quad (20)$$

Where, $i = 1,2,3 \ldots\ldots m$.

$$TSF(t_i) = \sum_{t_i \epsilon \{\{t\} \cup synonym\{t\}\}} \alpha.TF(t_i) \quad (21)$$

Here, *α= 1* for the term and *α= 0.5* for synonym of the term. *TF* is calculated as:

$$TF(t_i) = \frac{n_j}{\sum_k n_k} \quad (22)$$

Where,n_j is the number of occurrences of the term *j* in document collection and the denominator is the number of occurrencesof all terms in the document collection. *ISF* is calculated as:

$$ISF(t_i) = \log \frac{N}{n_i} \quad (23)$$

Where, n_i is number of sentences that contain term *i*. After calculating $TSF - ISF$, term-document matrix is constructed.
Document Clustering Algorithms:
Input: Term-document matrix
Output: Clusters with related documents.
Steps:

i. The first document is assigned to the first cluster and that cluster centroid is calculated by adding *TSF-ISF* values of all the terms in the document
ii. The remaining documents are clustered as:

Similarity between each cluster centroid and one of the remaining documents are calculated using cosine similarity measure. If the similarity value is greater than the given threshold range for any cluster, then the document is placed in that cluster and the centroid of that cluster is updated by taking the mean value of *TSF-ISF* values of all the terms in the cluster. If not, the document is placed in a new cluster and *TSF-ISF* values of the terms in the document is added and the result is assigned as new centroid of that cluster.

iii. Repeat step (ii) until all the documents are clustered.

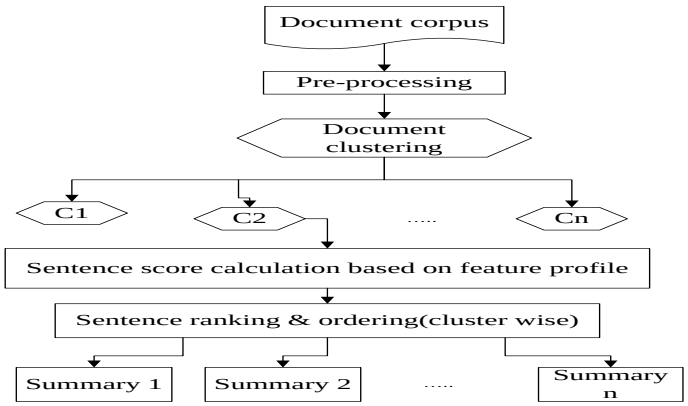

Figure1. Clustering and feature sentence extraction based multi-document summarization

Sentence Score Calculation Based on Feature Profile:

- Term Feature: $T\_F(Si,k) = \sum term\_weight(t).f(t,Si,k)$(24)
- Position Feature:$P_{F(Si,k)} = \frac{position(S_{i,k})}{3}$ (25)
- Sentence Length Feature: $P_L(Si,k) = \frac{N*length(S_{i,k})}{length(d_k)}$ (26)
- Sentence Centrality Feature:

$$C_F(Si,k) = \frac{words(S_{i,k}) \cap words(others)}{words(S_{i,k}) \cup words(others)}$$

- Sentence with Proper Noun Feature:$PN_{F(Si,k)} = \frac{PN_{COUNT}(S_{i,k})}{length(S_{i,k})}$ (28)
- Sentence with Numerical Data Feature:$ND_{F(Si,k)} = \frac{ND_{COUNT}(S_{i,k})}{length(S_{i,k})}$ (29)

Now,$Sentence_{score(Si,k)} =$

$$twT_{F(Si,k)} + tpP_{F(Si,k)} + tlP_L(Si,k) + tcC_F(Si,k) + tpnPN_{F(Si,k)} + tndND_{F(Si,k)}(30)$$

$Where, tw = 0.3, tp = 0.2, tl = 0.2, tc = 0.1, tpn = 0.1, tnd = 0.1$

A term t is a key word if $Term_{weight(t)} \geq 2 * \sum_{t \in D} Term\_weight(t)/n_tD$

Here ntD number of term in document cluster D.Then sentences are ranked according to their score values in descending order. After reordering all the sentences in each cluster, the summary is generated by extracting highly ranked sentences one at a time till the required summary length is met.

2.7 Paper VII

T. J. Siddiqui$et.al.$[22]proposed another method of multi-document summarization using sentence clustering for English language. Here the following processes are applied for summarization:

a) Pre-processing: Noise removal (Removing header and Footer), tokenization, stemming, frequency computation, sentence splitting are performed.

b) Feature extraction: Document feature: $DF = w1 + w2 + w3 + \cdots \ldots \ldots + wn$

Here, DF=Document feature of a sentence and $w1, w2, , w3 \ldots \ldots wn$ =Weight of the words contains the sentence.

Location feature: It gives high weight to the top and bottom sentences and low weight to the middle sentences.

Sentence reference index:If a sentence contains a pronoun, then the weight of the preceding sentence is increased.

Concept similarity feature:Number of sign sets of query words matching with words in the sentence.

c) Single document summary generation:

$$SW = v * DF + w * LF + x * SRI + y * CS(31)$$

Here, $v = 0.5\ w = 0.2\ x = 0.2\ y = 0.1$ now the sentence weights are normalized as below:

$$NormalizedWeight = \frac{\text{Weight of each sentence present in a document}}{\text{Maximum weight of any sentence present in the document}} \quad (32)$$

Sentences are ranked using normalized weight. Top K sentences are extracted to generate the summary for a single document.

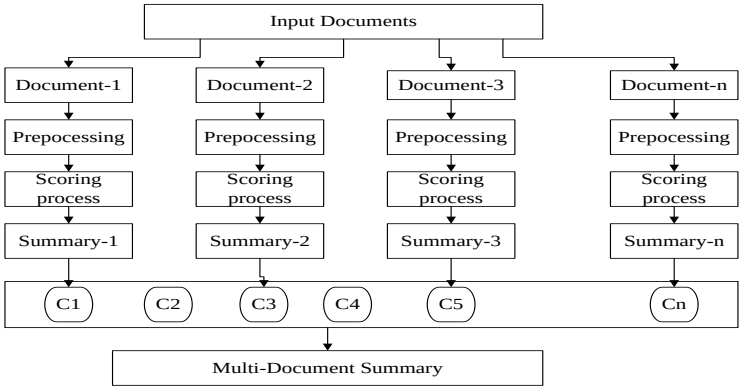

Figure2: A Multi document text Summarization using sentence clustering

d) Multi-document summary generation:

Sentences appearing in single document summaries are clustered, Top scoring sentences are extracted from each cluster and the sentences are arranged according to their position in the original document to generate the final multi-document summary as shown in Fig. 2.

Sentence clustering: It usessyntactic and semantic similarity.

- **Syntactic similarity: For example,**

S1= the cat runs faster than a rat, *S2*=the rat runs faster than a cat, Index no. for *S1={1,2,3,4,5,6,7}*, Index no for *S2={1,2,3,4,5,6,7},* Original order Vector,*V0={1,2,3,4,5,6,7},* Original order Vector, *Vr={* 1,7,3,4,5,6,2} So, the semantic similarity will be as follows:

$$Sim0(S1,S2) = \frac{\sum(v0 * vr) - \frac{\sum v0 * \sum vr}{k}}{\sqrt{(\sum v0^2 - \frac{(\sum v0)^2)}{k})(\sum vr^2 - \frac{(\sum vr)^2)}{k})}} \quad (33)$$

Here,*K*=number of words in *S1*and maximum value of syntactic similarity is 1 when the original and relative word is same.

- Semantic similarity:Semantic similarity between words i.e.,creating a graph.
  - Shortest path lengthIf the words are similar, then the shortest path length between them is 0. If the length is less, then words are more similar and if the length is more, than words are less similar.
  - Depth of sub summer

Semantic similarity between words:$S_w(w1,w2) = \frac{f(d)}{f(d)+f(l)}$ (34)

Here, *d*= Depth is the subsume,*l*=shortest path length and *f*=transfer function $f(x) = e^x - 1$

If words are exactly similar, then similarity=*1*; (*l*=0)

If words are dissimilar, then similarity=0; (no common parent)

If both *h* and *l* are non-zero, then the similarity between word *w1*&*w2* is defined as follows:

$$S_w(w1,w2) = \frac{e^{\alpha d} - 1}{e^{\alpha d} + e^{\beta l} - 2} (0 < \alpha, \beta \leq 1) \quad (35)$$

Here, *α, β*= smoothing factors

Information content:$I(W) = -\frac{logp(w)}{\log(N+1)}$ (36)

Probability of words, $P(w) = \frac{n+1}{N+1}$

Here, *n*=Frequency of the word in the corpus and *w*=Total no. of words in the corpus.

Now, the semantic similarity is:$Sim_s(S1,S2) = \frac{\sum_{wi \in s1} max_{wj \in s2}(Sw(wi,wj) * I_{wi})}{\sum_{w_i \in S1} I_{w_i} + \sum_{w_j \in S2} I_{w_j}}$ (37)

Overall similarity between two sentences is

$$Sim_{sen} = Sim_s(S_1,S_2) * \left((1-\gamma) + \gamma * Sim_0(S_1,S_2)\right) + Sim_s(S_2,S_1) * \left((1-\gamma) + \gamma * Sim_0(S_2,S_1)\right) \quad (38)$$

Here,*δ*=Smoothing factor

Thus for multi-document summary, the sentences are clustered using sentence similarity from each cluster and then single sentence is extracted.

2.8 Paper VIII

A. R. Deshpande*et. al.* [23] presented a text summarizer using clustering technique as shown in Fig. 3.

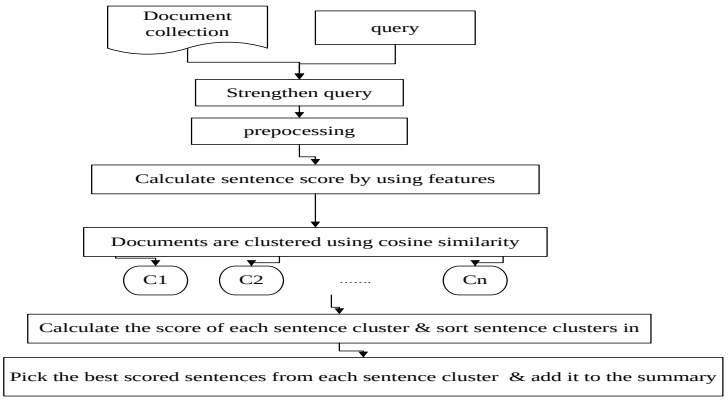

Figure3. Document & Sentence clustering approach to summarization

It is the clustering based approach that groups first the similar documents into clusters and then sentences from every document cluster are clustered into sentence clusters. And best scoring sentences from sentence clusters are selected into the final summary. For finding similarity, cosine similarity is used. It merged sentence and query. Then each word from the merged sentence is taken and checked whether that word appears in sentence and query both. If yes, then the weight $(tf * idf)$ of the word from document is used and placed that value in vector of sentence for the ith location in vector, and term frequency of the term is placed in vector of query.

2.9 Paper IX

A. Agrawal and U. Gupta proposed [24] an extraction based summarization technique using k-means clustering algorithm which is an unsupervised learning technique. Thescore for each sentenceis computed and centroid based clustering is applied on the sentences and extracting important sentences as part of summary.In this paper, tokenization method occurs as follows:

- All contiguous strings of alphabetic characters are part of one token; likewise with numbers
- Tokens are separated by whitespace characters, such as a space or line break, or by punctuation characters
- Punctuation and whitespace may or may not be included in the resulting list of tokens

For computing the score of sentence, first the *TF*IDF* of each individual words in the sentence is calculated.

Then,*K*-means clustering algorithm is applied. The main idea is to define *k* centroids, one for each cluster. These centroids are chosen to place them as much as possible far away from each other. This approach gives a precise summary because the densest cluster which is returned by the *K*-means clustering algorithm that consists of the sentences with highest scores in the entire document. These sentence scores are computed by summing up the$tf - idf$scores of individual terms in the sentence and normalizing it by using the length of the sentence.$tf - idf$.

2.10 Paper X

M. A. Uddin*et. al.* [25] presented a multi-document text summarization for Bengali text, where termfrequency (TF) based technology is used for extracting the most significant contents from a set of Bengali documents. Pre-processing includes tokenization, elimination or removal of punctuation characters, numeric digits, stop word etc.

The total term frequency (TTF) is measured bycounting the total numbers of appearances of a word in all the documents.

$$\sqcap TTF_i = \sum_{j=0}^{n} TTF_{ij}$$

Here, $i = 1, 2, 3 \ldots . kdifferentwordsandj$= 1, 2, 3 ….*n* number of documents.

I. Sentence Scoring (*SC)*

Score of a sentence is determined by summing up *TTF* of each word in that sentence as follows:

$\sqcap k \sqcap jSC_{kj} = \sum_{i=1}^{m} TTF$,if all the sentences have the same length

Sentence score of a long sentence is greater than a short sentence. And also smaller sentence is more meaningful than the larger one. So, SC is found by ordering$TTFi$ in decreasing order and the total words found in the document.

Or,$\sqcap k \sqcap jSC_{kj} = \sum (T - L + 1) * TTF$

Here, $k = 1, 2, 3 \ldots . knumberofsentences$,$j = 1,2,3 \ldots \ldots \ldots jnumberofdocuments,\ i = 1,2,3 \ldots \ldots \ldots \ldots mnumber$ of words in each sentence.*T*=total word, and *L*=word position.

II. Primary Summarization

SCs are sorted in decreasing order, *k* sentences are chosen as primary summarized content.To choose two sentences having identical meaning, this method would prefer one which is more descriptive and represents the

document better than the other one.Let consider, two sentences as vector, given two vectors of attributes S_x&S_y, so we get,

i. Cosine similarity measure:Sim_{xy} = Cos Θ

$$= \frac{\sum_{i=1}^{m} S_{xi} * S_{yi}}{\sqrt{\sum_{i=1}^{m} S_{Xi}^2} \sqrt{\sum_{i=1}^{m} S_{yi}^2}} \quad (39)$$

Here, $S_x = S_y$ =TTF of words in two sentences $i.e. -1 < 0 < 1$, indicates that Dissimilarity<Independence<Similarity.

ii. $A.Sim_i = \frac{\sum_{i=1}^{m} Sim_{iy}}{m} = \frac{\sum_{i=1}^{m} TTF}{m}$ (40)

$$A.Sim_i \geq M.Sim$$

Here, $A.Sim$ = Average sentence similarity, $M.Sim$ = Maximum similarity or threshold value ($i.e.95\%$) ∩ $If A.Sim_i \geq 95\%$, then $P(S_j|S_i) = \frac{P(S_j \cap S_i)}{P(S_j)}$

Here $P(S_j|S_i) = ProbabilityofSjbeingsimilartoallothersentenceSihasbeenSelected P(S_j \cap S_i) = Probabilityofhavingsimilaritybetween S_j \& S_i$, $P(S_j) = A.SimofSj$. Let consider, $P(S1) = 0.625/N$, $P(S1 \cap S2) = 0.5/N, so, P(S1|S2) = 0.5/0.625$ as shown in table 2.

Table 2: A simple probability calculation for generating sentence relevancy

	S1	S2	S3	$A.Sim$
S1	1	0.5	0.5	0.625
S2	0.5	1	0.5	0.625
S3	0.5	0.5	1	0.625

Every sentence is considered as a node and $P(S_j|S_i)$ is the weight of the edge connecting to the node S_j&S_i.As a result, we get an undirected graph. The sentence having largest $A.Sim$ value is chosen to be the first sentence. For finding the best relevancy of sentences, the A* search algorithm is applied. Then it will find the final summary.

2.11 Paper XI

Another work proposed by M. I. A. Efat*et.al.* [26] by sentence scoring andranking for Bengali language.

Pre-processing:

It includes tokenization, stop words removal (but, or, and am, is, are, a, an, the etc.) and stemming.

Sentence Scoring & Summarization:

i.Frequency:$STF_K = \sum_{i=1}^{n} WF$

Here, STF_K= Sentence Total Frequency, *WF*=Word Frequency and *n*=number of words in a sentence.

ii. Positional Value:$PV_K = \frac{1}{\sqrt{K}}$ Here, *k*=the actual positional value of a sentence in the document.

iii. Cue words consist of "therefore","hence", "lastly", "finally", "meanwhile", on the other hand etc. and skeleton word of the documentconsists of the words in tittle and headers.

Sentence scoring is done as follows:

$$S_k = (\alpha * STF_k) + (\beta * PV_k) + \gamma + \lambda, 0 \leq \alpha, \beta, \gamma, \lambda \leq 1 \quad (41)$$

Here,α*=0.1,* β*=0,* γ*=0.7*(Cue words co-factor) and λ*=0.4*(Skeleton co-factor of the document)

iv. Summary Making: After ranking the sentences based on S_k, *X* number of top ranked sentences are selected for producing the summary. Comparison between the human summarizer and the machine summarizer can be measured by three equations. For each document, we let *kh* be the length of the human summary, *km* the length of the machine generated summary and the *r* the number of sentences they share in common. The method defined precision *(P)*, Recall *(R)* as metrics, then $P = 100\frac{r}{Kh} R = 100\frac{r}{Km}$ and $F1 = 100\frac{2r}{Kh+Km}$.

2.12 Paper XII

H. Dave and S. Jaswal [28] presented a multi-document summarization system using hybrid summarization techniques. There are two main blocks in proposed model, an extractive summary and an abstractive summary.

Steps for generating extractive summary are as follows-

Linguistic Analysis: It extracts sentences from the input documents, and then performs tokenization [11]. Redundancy Detection:In multiple documents there are chances of repetition of sentences. Hence redundancy detection is important to reduce the unwanted and repetitive sentences or words. This process is carried out using stop word removal and stemming.Sentence Representation:Sentence representation is the process of calculating the frequency of relevant sentences weight.

Term Frequency: Each entry of a sentence vector denotes the term weight which is explained by the equation:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (42)$$

Where$n_{i,j}$ is the number of occurrences of term t_i in sentence s_j and $\sum_k n_{k,j}$ is the sum of number of occurrences of all the terms in sentence,s_j.The last process is the generation of summary. Sentence representation is used to find the score (weight) of word from each sentences. This score is assigned to each sentence and generates the highest rank by:

$$max_n = log_2 \frac{N}{n_i} \quad (43)$$

Steps for generating abstractive summary are as follows-

Word Graph Generation:Word graph generation is phase of finding out important nodes from extractive summary. Here, a set of heuristic rules are applied such as hypernymy, holonymy, and entailment. Domain Ontology: According to Tom Gruber ontologyis a specification of a conceptualization [29]. It provides a vocabulary and set of Synset. This synset consist of Synonym, Antonym etc. Ontology represents the domain which talks about the same topic having same knowledge. Here the domain ontology is defined by domain experts. Next meaningful terms are produced by preprocessing and classifier classifies those terms.

WordNet: WordNet is lexical database of English. It is used to define word meaning and models. It consists of a set of synonyms called as synsets, and is also used as combination of dictionary and thesaurus. The Synsets provide different semantic relationships such as synonymy (similar) and antonym (opposite), hypernym (super concept)/hyponymy (sub concept), meronymy (part-of) and holonymy (has-a). Hyponym shares a type-of relationship with its hypernym. Meronymy is defined as a word that denotes a constituent part or a member of something.

2.13 Paper XIII

F. E. Gunawan, A. V. Juandi and B. Soewito [30] presented an automatic text summarization using text features and singular value decomposition for popular articles in Indonesia language. The scoring procedure is described as follows:

TF-IDFWeighting: The notation TF_i denotes the frequency of the *i* word appears in the document; meanwhile, the notation IDF_i denotes the inverse of the number of sentences that has the i word.
Singular Value Decomposition (SVD): The above TF-IDF process results in the terms-by-sentences matrix, A € R^{m*n}. The number of rows *m* denotes the number of words and *n* is the number of sentences in the article. The entity $a_{i,j}$ in the matrix denotes the frequency of the *i* word in the j sentence. The matrix *A* is usually a sparse matrix. Subsequently, the matrix is decomposed into three matrices by the singular value decomposition: *A*= USV^T, where *U* is the matrix of the left singular vectors, *S* is the matrix of the singular values, and *V* is the matrix of the right singular vectors. The SVD text summarization procedure can be performed with the steps described in algorithm as follows:

Step i: Decompose the document into sentences
Step ii: k <- 1
Step iii: Establish the terms-by-sentences matrix A
Step iv: A = USV^T
Step v: while (required), do
Step vi: Take the *k*th singular vector in V
Step vii: Put the sentence associated with the singular vector into the summary
Step viii: k <- k + 1
Step viii: End
Word Stemming: This process intends to remove all suffixes from words and produces their roots.
Performance Evaluation: The performance of the current implementation is measured by Precision =$\frac{\#CorrectSentences}{\#CorrectSentences + \#WrongSentences}$, Recall=$\frac{\#CorrectSentences}{\#CorrectSentences + \#MissedSentences}$, F=2$\frac{Precision\ .\ Recall}{Precision + Recall}$

III. RESULT AND COMPARATIVE DISCUSSION

The comparative result of the reviewed papers is illustrated in the following table.

Table 3: Comparative study of proposed technique with existing methods

Paper no.	Language	Document type	Major operations
Paper I[17]	English	Single(Graph based)	Subtopic detection(KNN), word scoring (TF*IDF), ranking sentences(calculating Hub & Authority), ordering subtopic(Markov model)
Paper II[18]	English	Single	Preprocessing,measure relevancy(PMI), Word significance estimation
Paper III [19]	Chinese	Single	Fuzzy similarity matrix(TF*IDF), maximum spanning tree generation, sentence weight computation(LexRank& using feature)
Paper IV [20]	English	Single(Query based)	Sentence ranking from query dependent view(Cosine relevancy), query independent view(cosine similarity, Markov chain), Multi-view ranking fusion
Paper V	English	Multiple	Preprocessing, clustering using cosine similarity,

[21]			cluster ordering
Paper VI [3]	English	Multiple	Preprocessing, clustering using TSF-ISF, sentence scoring(Feature based)
Paper VII [22]	English	Multiple	Preprocessing, Sentence scoring (Feature based), summary, and clustering using syntactic & semantic similarity.
Paper VIII [23]	English	Multiple(Query based)	Preprocessing, Sentence scoring (Feature based), clustering using cosine similarity
Paper IX[24]	English	Single	TF*IDF, sentence scoring, K-means clustering
Paper X[25]	Bengali	Multiple	Preprocessing, sentence scoring(TTF) Cosine Similarity measure, A* algorithm
Paper XI[26]	Bengali	Single	Preprocessing,sentence scoring(Feature based), sentence ranking
Paper XII [28]	Any	Multiple	Extractive summary: Tokenization, linguistic analysis, redundancy detection, sentence representation, TF; Abstractive summary:word graph generation, domain ontology, WordNet
Paper XIII [30]	Indonesia	Multiple	Sentence segmentation, tokenization, stop word removal, stemming, TF-IDF weighting, singular value decomposition

IV. CONCLUSION

In this paper, the state-of-the-art of extractive text summarization techniques for various languages has been described. We can notice that good work has been done for various foreign languages like English, Chinese etc. But summarization system for Bengali languages is still in lack. Hence, it is challenging to propose a summarization technique using different types of features. In future, we are aiming to use more features for extracting Bengali sentences. Also, we will try to compare different machine learning techniques for summarization and to achieve better accurate results. Also, we will try to test the techniques rigorously on large dataset of various domains like news, autobiography,etc.

REFERENCES

[1]. Unnamed, Big Data. [Online]. Available: http://searchcloudcomputing.techtarget.com/definition/big-data-Big-Data

[2]. M. Haque, S. Pervin, and Z. Begum, Literature Review of Automatic Multiple Documents Text Summarization, International Journal of Innovation and Applied Studies, 3(1), 2013, 121-129.

[3]. A. Kogilavani, P. Balasubramani, Clustering and Feature specific sentence extraction based summarization of multi-documents, International Journal of Computer Science & Information Technology (IJCSIT), 2(4), 2010.

[4]. R. M. Chezian,and Ahilandeeswari. G., A Survey on Approaches for Frequent Item Set Mining on Apache Hadoop, International Journal for Trends in Engineering & Technology, 3(3), 2015.

[5]. R. Pradheepa, and K. Pavithra, A Survey on Overview of Data Mining, International Journal of Advanced Research in Computer and Communication Engineering, 5(8), 2016.

[6]. A. Totewar, Data mining: Concepts and Techniques. [Online]. Available http://www.slideshare.net/akannshat/data-mining-15329899, 2016.

[7]. E. Turban, J. E. Aronson, T. Liang, and R. Sharda, Decision Support and Business Intelligent Systems, Pearson Education, 8th Edition.

[8]. Unnamed, Data mining techniques and algorithm. [Online]. Available http://www.oracle.com/technetwork/database/enterprise-edition/odm-techniques- Algorithms-097163.html

[9]. Unnamed, Data Mining-Applications & Trends [Online]. Available: ww.tutorialspoint.com/data_mining/dm_applications_trends.htm

[10]. M. Verma, and D. Mehta, A Comparative study of Techniques in Data Mining, International Journal of Emerging Technology and Advanced Engineering, 4(4).

[11]. A. Nenkova and K. McKeown, Automatic Summarization, Foundations and Trends® in Information Retrieval, 5(2–3), 2011, 103–233.

[12]. A. H. Witten, Text mining. [Online]. Available: http://www.cos.ufrj.br/~jano/LinkedDocuments/_papers/aula13/04-IHW Textmining.pdf

[13]. F. El-Ghannam and T. El-Shishtawy, Multi-Topic Multi-Document Summarizer, International Journal of Computer Science & Information Technology (IJCSIT), 5(6), 2013.

[14]. V. Gupta and G. S, Lehal, A Survey of Text Summarization Extractive Techniques, Journal of Emerging Technologies in Web Intelligence, 2(3), 2010.

[15]. N. R. Kasture, N. Yargal, N. N. Singh, N. Kulkarni and V. Mathur, A Survey on Methods of Abstractive Text Summarization, International Journal for Research in Emerging Science and Technology,1(6), 2014.

[16]. R. Ferreira, L. de S. Cabral, R. D. Lins , G. P. Silva, F. Freitas , G. D. C. Cavalcanti, R. Lima, S. J. Simskeand L. Favaro, Assessing sentence scoring techniques for extractive text summarization, ELSIVIER International Journal of Expert systems with Applications, 2013.

[17]. J. Zhang, L. Sun, and Q. Zhou, Cue-based Hub-Authority approach for Multi- document Text Summarization, IEEE International Conference on Natural Language Processing and Knowledge Engineering, China, 2005, 642 – 645.

[18]. Y. Ouyang, W. Li, and Q. Lu, An Integrated Multi-document Summarization Approach based on Word Hierarchical Representation, Proceedings of the ACL-IJCNLP, China, 2009, 113-116.
[19]. X. Li, J. Zhang and M. Xing, Automatic Summarization for Chinese text based on Sub Topic Partition and Sentence Features, IEEE 2nd International Symposium on Intelligence Information Processing and Trusted Computing (IPTC), China, 2011.
[20]. P. Hu, T. He and H. Wang, Multi-View Sentence Ranking for Query-Biased Summarization, IEEE International Conference on Computational Intelligence and Software Engineering (CISE), China, 2010.
[21]. K. Sarkar, Sentence Clustering-based Summarization of Multiple Text Documents, TECHNIA – International Journal of Computing Science and Communication Technologies, 2(1), 2009.
[22]. T. J. Siddiki and V. K. Gupta, Multi-document Summarization using Sentence Clustering, IEEE Proceedings of 4th International Conference on Intelligent Human Computer Interaction, India, 2012.
[23]. A. R. Deshpande, and Lobo L. M. R. J, Text Summarization using Clustering Technique, International Journal of Engineering Trends and Technology (IJETT), 4(8), 2013.
[24]. A. Agrawal and U. Gupta, Extraction based approach for text summarization using k-means clustering, International Journal of Scientific and Research Publications, 4(11), 2014.
[25]. M. A. Uddin, K. Z. Sultana and M. A. Alom, A Multi-Document Text Summarization for Bengali Text, IEEE International Forum on Strategic Technology (IFOST), Bangladesh, 2014.
[26]. M. I. A. Efat, M. Ibrahim , H. Kayesh, Automated Bangla Text Summarization by Sentence Scoring and Ranking, IEEE International Conference on Informatics, Electronics & Vision (ICIEV), Bangladesh, 2013.
[27]. L. C. Reddy and Venkatadri. M, A Review on Data mining from Past to the Future, International Journal of Computer Applications, 15(7), 2011.
[28]. H. Dave and S. Jaswal, Multiple Text Document Summarization System using Hybrid Summarization Technique,1st International Conference on Next Generation Computing Technologies (NGCT-2015), India, 2015, 4-5.
[29]. J. A. Kwak and H.-S. Yong, Ontology Matching Based On Hypernym, Hyponym, Holonym, and Meronym Sets in Wordnet, International Journal of Web Semantic Technology (IJWesT),1(2), 2010.
[30]. F. E. Gunawan, A. V. Juandi and B. Soewito, An Automatic Text Summarization using Text Features and Singular Value Decomposition for Popular Articles in Indonesia Language, IEEE International Seminar on Intelligent Technology and Its Applications, 2015.