(IJISE) 2017, Vol. No. 6, Jul-Dec e-ISSN: 2454-6402, p-ISSN: 2454-812X

ANALYSIS FOR TEXT SUMMARIZATION ALGORITHM USING DIFFERENT DATASETS

GAURAV MIDHA
Department of Polymer Science and Chemical Technology
Delhi Technological University

1. INTRODUCTION

1.1 Introduction

Text summarization is a technique to create a succinct and vital snippet of data from a bigger arrangement of content which can be a content report, an article or a blog. Text Summarization intends to give a rundown of given content while safeguarding its data and expectation. The rundown is a little snippet of data that portrays an arrangement of passages or records. Outline produced is by and large under 40% of the first content information and it ought to be even not as much as that on account of substantial datasets. The synopsis ought to hold the vital information introduce in the record, ought to be controllable, short and concise. Synopsis of content information is done from numerous points of view contingent on the different parametersin light of the position and organization of words and sentences.

Automatic Text Summarization[1] aggregates the information from a few archives to show the last shorter snippet of data subsequently, which is shorter, instructive and jelly the genuine goal of data. These little compressed adaptations spare significant time by displaying unambiguous essential data. With the expanding measure of advanced information, it has turned out to be hard to recover the required and compact data. Programmed content outline takes into account the very need of the time.

There are techniques which are useful to create an outline. In the first place Division which sorts the synopsis approaches depends on the substance of the outline delivered. There are two methodologies Extraction and Abstraction[2]. As the name proposes, Extraction is area autonomous, it for the most part goes for discovering the essential sentences and later exhibiting an arrangement of imperative sentences as Summary. Despite what might be expected, Abstraction is area subordinate, it forms the accessible data and new sentences are set up by understanding the substance, likewise thinks about human information by setting up the objective to create a synopsis.

2. SINGLE DOCUMENT SUMMARIZATION ALGORITHMS

This chapter discusses the different text summarization algorithms which summarize single documents. Each of these algorithms is explained briefly. These algorithms are then implemented and compared for chosen datasets.

2.1 Text Summarization Algorithms

Text summarization is done to shorten the text and get to the main point of the document. Summaries are easy to read and understand. Following text summarization algorithms are studied in this thesis.

1. TextRank
2. TextTeaser
3. Summary using Word features

2.1.1 TextRank

TextRank[5], an unsupervised algorithm based on weighted-graphs from a paper by Mihalcea etal. It is built on top of the popular Page Rank algorithm that Google used for ranking web pages. TextRank works as follows:

1. Pre-process the text: remove stop words and stem the remaining words.
2. Create a graph where vertices are sentences.
3. Connect every sentence to every other sentence by an edge. The weight of the edge is how similar the two sentences are.
4. Run the PageRank algorithm on the graph.
5. Pick the vertices(sentences) with the highest PageRank score

In original TextRank the weight of an edge between two sentences is the percentage of words appearing in both of them. This TextRank uses a function to see how similar the sentences are.

Graph based algorithms are used to rank the text sentences or words for summarization. To enable working with text on these algorithms, text is represented as graph, where a word depicts the nodes of the graph and edges represent meaningful relations among nodes. Edges represent the connection between two vertices of the graph. Sentences or collocations may also be assigned as vertices of the graph depending upon the size of input dataset. Edges may represent lexical relations, content overlap etc.

Keyword Extraction:

Keyword is a method to locate the main keywords in the document which represent the subject of the present information. These identified words contains the most relevant content of the document. An automatic index of a document collection may be prepared by accumulating lists of these words. Keyword extraction can be efficiently used in making dictionaries associated to specific domains. Keywords chosen by this method are present in the original text. Formation of new words or similar words is not considered as Keyword extraction. Selected keywords represent useful entries for information retrieval, data mining and text summary generation. A very simple approach to identify significant keywords is by calculating term frequency. Others may include popularity, context, position etc. to find out the key phrases.

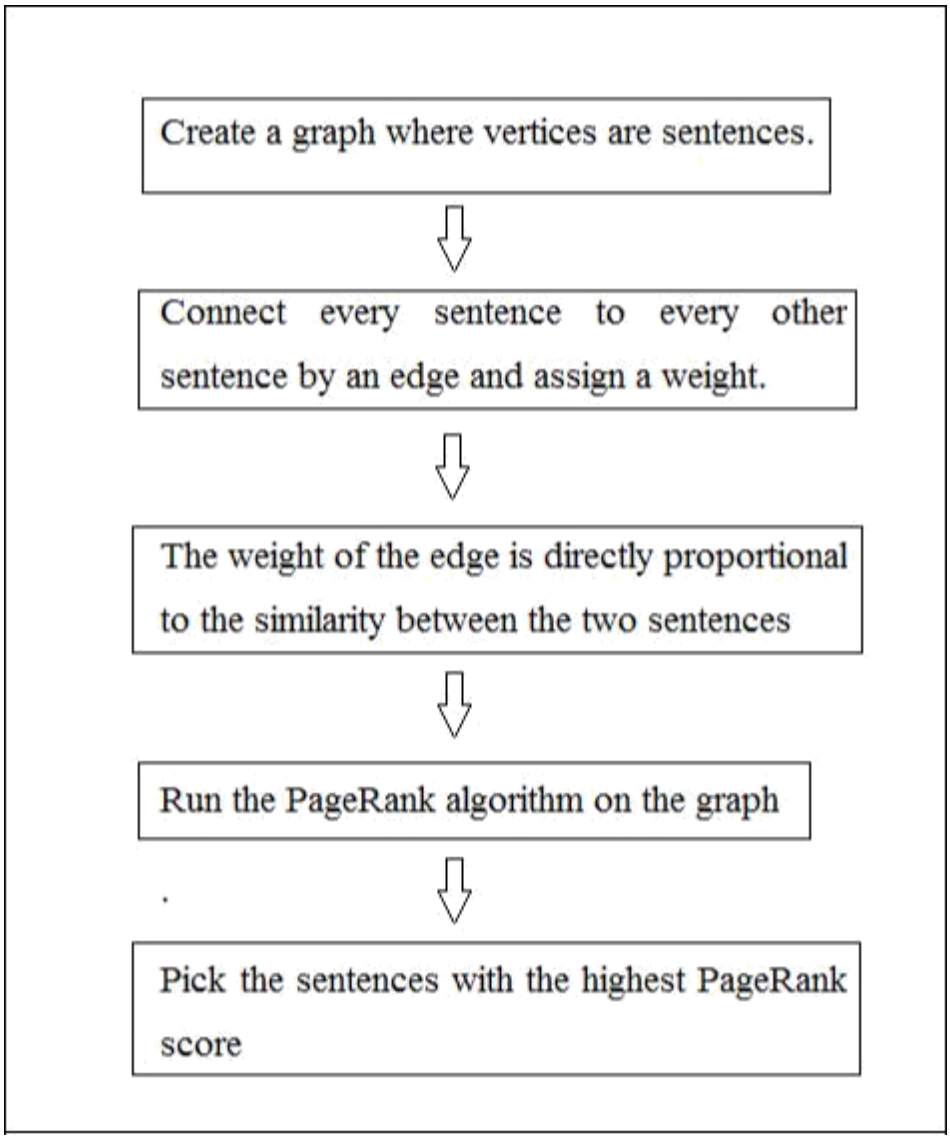

Figure 2.1: Flow chart of TextRank[14]

PageRank[14] presents a popular method to calculate the importance of a page in a set of pages joined together by links. It works by measuring the quanitative and qualitative score of links associated to a specific page. It computes an approximate score on the basis of that more websites are likely to contain forward links to important and popular websites. This algorithm analyse the links among different pages and assigns a numerical score to each element of the connected document set. It measures the relative importance of an entity in a set, like in the World Wide Web. The PageRank algorithm can be used to evaluate importance for any collection of elements which has references among themselves. For an element D, P(D) represents the associated PageRank.

2.2 TextTeaser

TextTeaseris based upon sentence features[16], which is a heuristic approach for extractive text summarization.

TextTeaser associates a score with every sentence. This score is a linear combination of features extracted from that sentence. Features that TextTeaser looks at are:

- *titleFeature*: The count of words which are common to the title of the documentand sentence.
- *sentenceLength*: Authors of TextTeaser defined a constant "ideal" (with value 20), which represents the ideal length of the summary, in terms of a number of words. *sentenceLength* is calculated as a normalized distance from this value.
- *sentencePosition*: Normalized sentence number (position in the list of sentences).Introduction and conclusion will have a higher score for this feature.
- *keywordFrequency*: Term frequency in the bag-of-words model (after removingstop words). Keyword frequency is just the frequency of the words used in the whole text.

More on the sentence features for summarization see Sentence Extraction Based SingleDocument Summarization by Jagadeesh et al [16].

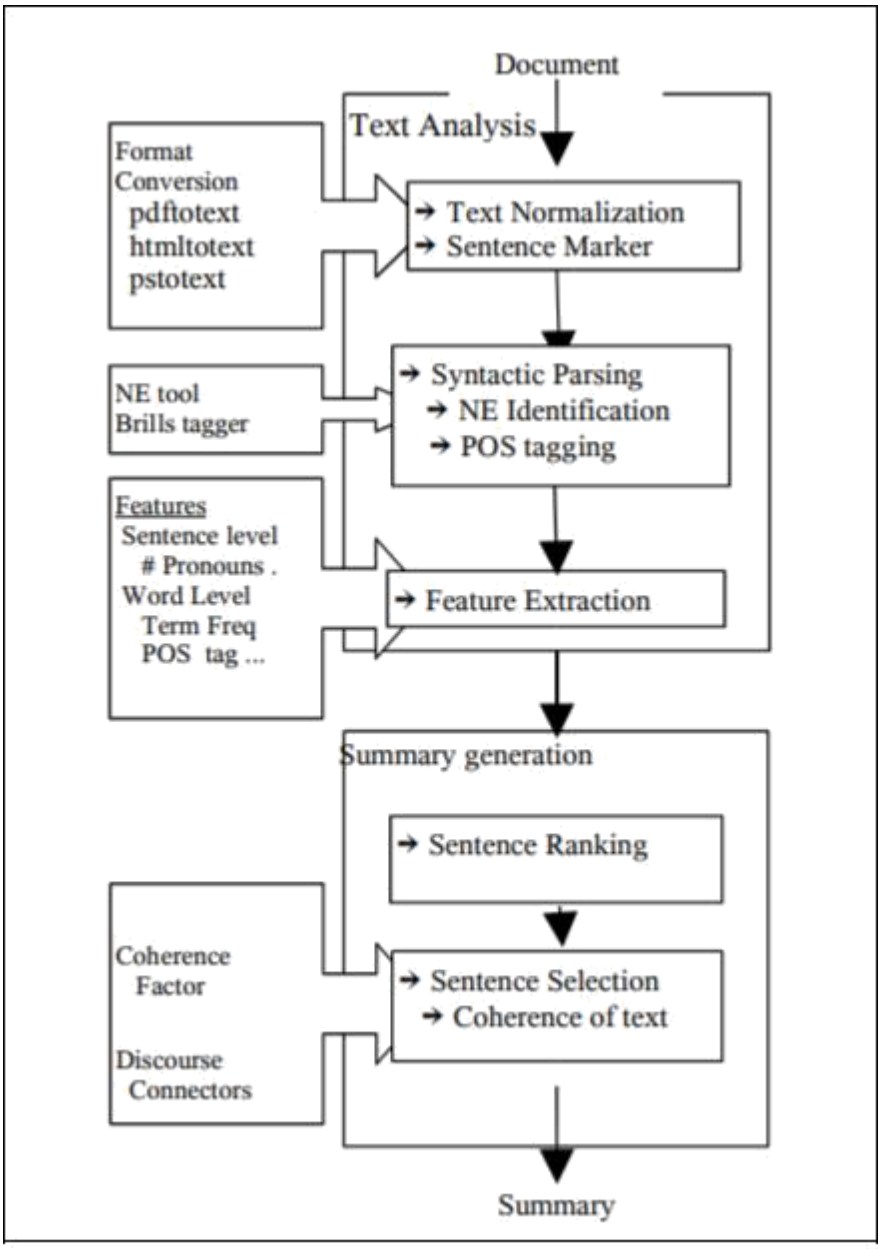

Figure 2.2: Flowchart of TextTeaser[16]

Sentence Marker: It is used to split the document into sentence units.

Syntactic Parsing: It is done by sentence structure analysis using NLP tools like Brills tagger [Brill], named entity extractor, etc. This extractor recognizes named entities (like persons, organizations, and locations etc), temporal expressions (time and date) and specific numerical values expression from textual data.

Feature Extraction: Both the word level features are extracted to be used in the calculation of the relevance and importance of the sentence present in the document. The word level features are listed below:

1. Length of the word l(w)
2. Familiarity of the word f(w)
3. Parts of speech tag p(w)
4. Term frequency tf(w)
5. Font style F(**w)**
6. Occurrence in headings O(w)

The sentence level features are:

1. Length of the sentence
2. Presence of the verb
3. Pronouns referring to preceding sentences
4. Position of the sentence in source document

Sentence Ranking and Summary Generation

Most of the times word features depends on the context of its occurrence, i.e they may depend on the sentence position and number also(ex. POS tag, familiarity, ..). Similarly, the word score also depends on the sentence number in the document. Once the feature vector is extracted for each sentence, the score of a sentence is calculated by obtaining the total sum of individual words as :

$$\text{Score}(l, w) = \prod$$

$$\text{Score}(l) = \sum \text{Score } l \text{ wi}$$

where l, represents the sentence number and „w‟ represents the word present in the sentence, and f i (w) represents the ith feature value.

After the sentence scores are assigned, sentences are selected to form good summary. One method is to extract the top N sentences but this may lead to the coherence problem.

Coherence Score(CS): Coherence score[32] is used to identify the amount of commoninformation between the set of already selected sentences and the new sentence to be included. A list of words is used to evaluate the coherence of the sentences.

Let S_w represents the set of words in the already selected sentences, and lw denotes the set of words present in the new sentence to be selected, then coherence score is obtained by the total sum of the common word scores. Now the score of the new sentence is computed by

$$CF \times CS\ (l) + (1 - CF\ ) \times SPW\ (l)$$

where CF denotes the Coherence Factor.

2.3 Summary Algorithm based on Word Features

This Algorithm[33] aims to provide an efficient manner of reducing a document to an understandable text, which is done by selecting the most important sentences. The algorithm has following key steps:

- Ranking sentences using the below-described algorithm.
- Transition phrases and unnecessary clauses are removed.
- Excessive examples are removed.
- Reorganizing the summary to focus on a topic; by selection of a keyword.

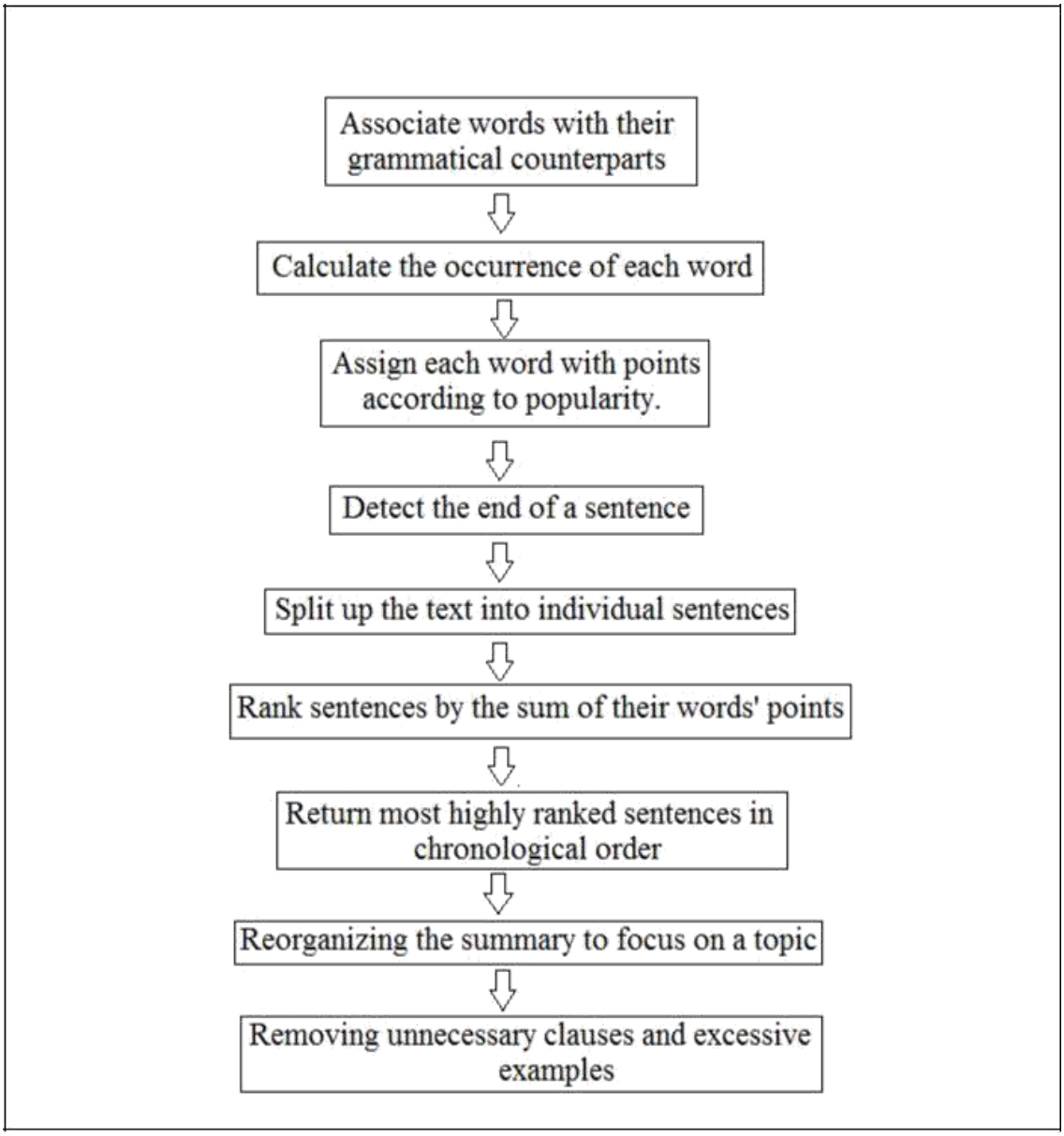

Figure 2.3: Flowchart of Summary Algorithm

The core algorithm has 7 key steps listed below:

1. Associate each word with the grammatical equivalents. (e.g. "light" and "lights")
2. Compute the frequency of each word in the document.
3. Assign each word with points depending on their popularity.

4. Determine the correct ending of a sentence. (e.g "4.5" does not).

5. Separate individual sentences from the text.

6. Rank sentences on the basis of obtained sum of associated words' points.

7. Select X topmost sentences.

3. MULTI-DOCUMENT ALGORITHMS

Multi-document summarization is an automatic procedure to create a summary which includesimportant information on key topics from multiple documents. It creates a concise and comprehensive summary. Here, Algorithms are presented to perform summarization based on different methods to evaluate the accuracy of produced summary for different Multi-document datasets.

This Subsection includes various Multi-document text Summarization algorithms based on:

- □ Summarization Using ILP Based Multi-Sentence Compression
- □ LDA topic model
- □ Sentence Clustering

3.1 Multi-Document Abstractive Summarization Using ILP Based Multi-Sentence Compression

Abstractive summarization is an ideal form of summarization as it alters the given source documents sentences to form new informative, non-redundant and coherent sentences to be included in the final summary. Sentences produced should be easily understandable and readable. To form completely new phrases matching to the human understanding is not yet achieved but this algorithm tries to maximize information content by combining words from multiple sentences.

This Algorithm performs Multiple document summarization using integer linear programming model[34] which aims to produce coherent and highly informative sentences. First, Algorithm employs LexRank[35] to find out the most important document from the set of source documents. Then, the sentences belonging to the most important document are aligned to the sentences of another document to generate clusters of similar sentences. In each of the generated

cluster, k-shortest paths from the sentences are generated with the help of word-graph structure. Finally, sentences are selected by the help of shortest paths generated employing a novel integer linear programming method in order to form new informative coherent sentences. Above stated shortest paths are represented as binary variables in the ILP method and number of words in a sentence path, information and quality score are considered in the function.

Steps in the Algorithm:

There are two main steps in the algorithm:

1. Sentence Clustering

2. Summary Sentence Generation

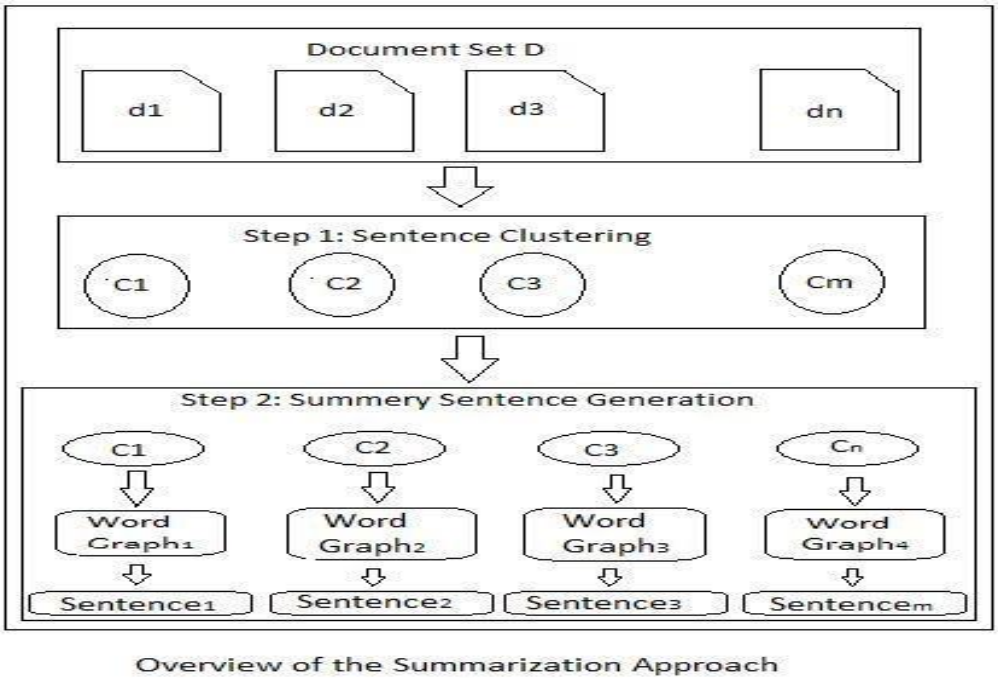

Figure 3.1 Overview of the Summarization approach

The above stated two steps of the Algorithm are further divided into the following steps:

Step 1: Sentence Clustering

Clusters of sentences are created using each sentence from the most important document, , in a document set D. It is assumed that is comprised of the most important content across

all the documents present in the set D. The document which contains more similar information to central content is most informative.

(Step 1.1) Document Importance

We propose several techniques to identify

LexRank: LexRank [35] creates a sentence graph where the edges represent weights which arecalculated by the help of inter-sentence cosine similarities. While in this algorithm, a graph of documents is constructed to calculate the importance of a document. The equation below shows a formula to calculate LexRank score for a node in a graph using weighted links present among nodes. This computed score represents the importance of the document in the set of input documents. Let p(x) denotes the centrality of node x in the equation below:

$$p(x) = \frac{\sum}{\sum}$$

where adj[x] denotes the set of adjacent nodes to u and N represents the total number of nodes present in the graph, „d" denotes damping factor(set to 0.85). Document representing the node with the highest LexRank score is a most important document, D_{imp} for the set of input documents.

Pair-wise Cosine Similarity: It is used to calculate the average cosine similarity between the current document d_i and the other documents present in the input dataset. The equation to find out average cosine similarity is:

$$AveCosSim(d_i) = \frac{\sum_{(i \neq j)}}{| |}$$

where [D] denotes the number of total documents present in the document set D.

Overall document collection similarity: This method is used to calculate the cosine similarity between the current document(d_i) and the whole input document set. We obtain the document set by concatenating the data from all the documents of the dataset D. This is calculated as:

$$DocSetSim(di) = CosSim(di,D):$$

After selecting the most important document, d_i from the input dataset D, we create the clusters by aligning sentences and arranging them based on their original positions in the input documents.

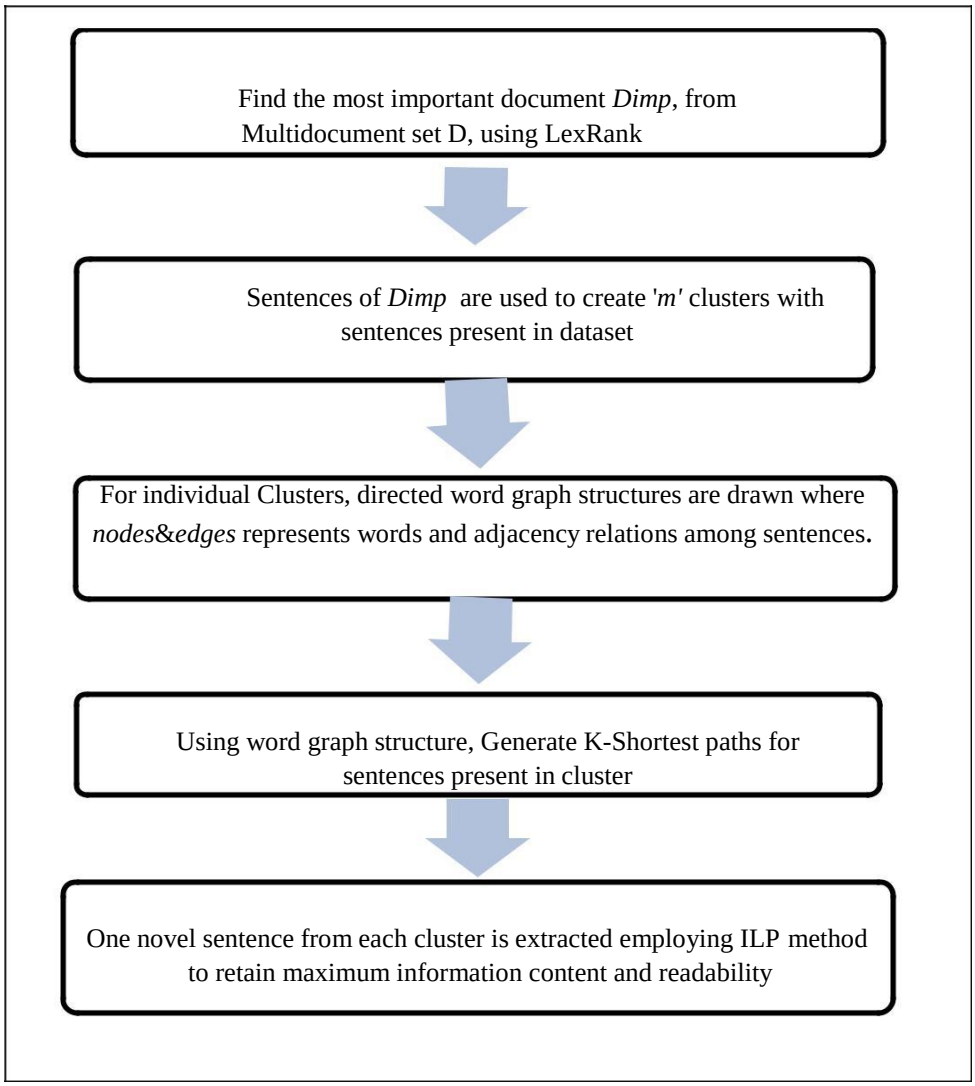

Figure 3.2: **Flowchart for Multidocument Abstractive Summarization using ILP based Sentence Compression**

3.2 Multi Document Summarization Algorithm based on LDA Topic Model

This Multi Document Summarization Algorithm[36] is based on the Latent Dirichlet Allocation (LDA) topic model which takes a multiple numbers of documents as input and generates a final output summary including an important piece of information from all the input documents. Latent Dirichlet allocation is a popular topic model which finds topics on the basis of word frequency i.e. occurrences of a word from a set of input documents. It presents the input text as a mixture of latent topics; these topics represent the key concepts in the document. LDA isparticularly designed for identifying a reasonably accurate number of topics within a given document set. LDA (Latent Dirichlet Allocation) Model is used to find the important topics inthe input provided.

These latent topics are useful to employ sentence ranking methods in order to obtain good quality summary. The sentence ranking mechanism calculates the posterior probability of each sentence based on two factors i.e. the topic distribution of the sentence and topic importance. Here, Topic Distribution denotes the degree to which a sentence belongs to a particular identifiedtopic and Topic Importance denotes the importance of the topic depending upon the amount of information covered by this topic in the documents provided. After obtaining the probability for each of the existing sentences, it extracts the important sentences to be included in the final optimized summary based upon the above calculated posterior probability.

Topic-Importance: Topic importance represents the significant portion of the document coveredby a topic. A topic covering a large amount of content of the document will be assigned higher probability value and vice-versa. All the latent topics identified by LDA will havedifferent probabilities. Topic Importance refers to the posterior probability of the topic in the document. All identified topics in a document are not of equal probabilities, as it depends upon the part of the document which can be represented by a topic. Before selection of sentences to beincluded in the final summary based upon posterior probability, the topic importance should be calculated.

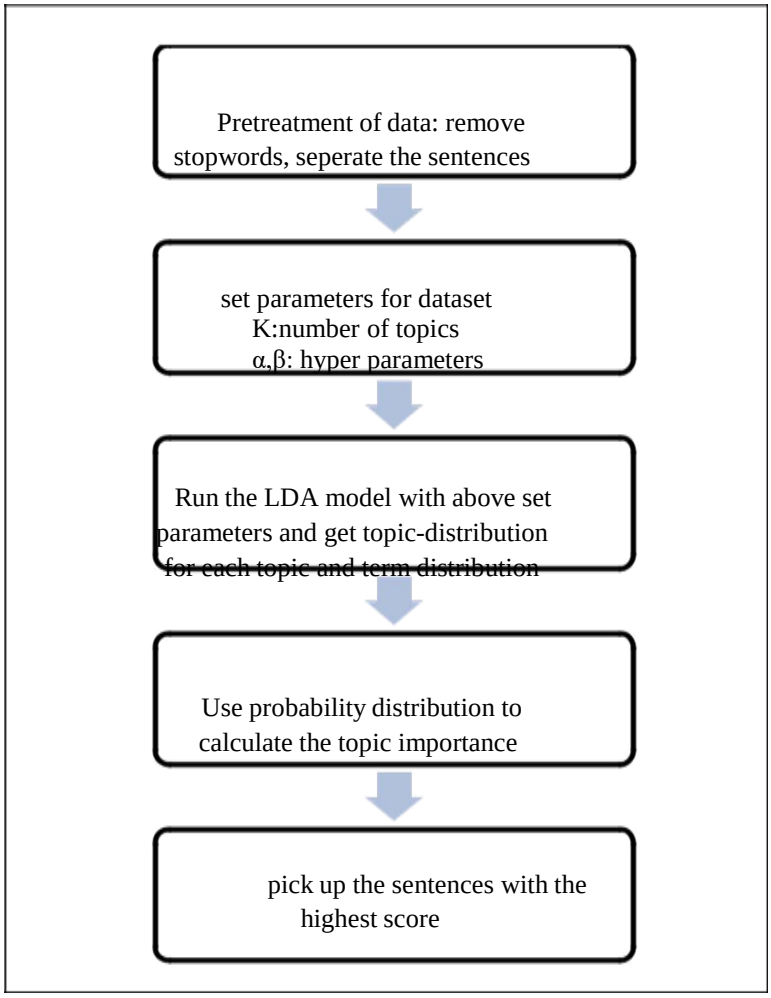

Figure 3.3: Flow-Chart: Multi-Document Summarization based on LDA topic Model

Topic importance for a topic distributed among a set of documents is calculated by evaluating the distribution of topic over all the input documents. The prior probability for all the documents is equal, which implies that initial order of documents has no impact on the value of Topic importance.

The formula to calculate Topic Importance is (1)

$$P(T = \ | D) \quad \sum \quad T = \ | D=d)$$

Where P(T = | D) refers to the topic importance and (T = | D=d) refers to the topic-distribution obtained by LDA Model for a document.

Topic importance is directly proportional to the content covered and the length of the document. Topic covered in a lengthy document has a higher weight assigned in Topic importance. The formula to calculate Topic importance in this case is

$$\frac{\sum \quad |}{P(T = \ | D) \qquad \sum}$$

Where is the total number of words in a document *d* .

Sentence Ranking Algorithm:

Sentence ranking is done to select the sentences to be included in the final summary by evaluating the score for each sentence i.e. the posterior probability. The probability depends on two factors, first the topic importance and other is topic distribution. The sentences with high- weight posterior probability are used to form summaries. So, to evaluate the Posterior probabilityof a sentence, the following method is used.

The degree of a sentence associated with a certain topic is represented by the Conditional Probability. The Conditional Probability is calculated as ()

$$P(S = \quad \prod \quad |T= k,D) \quad |$$

Where P(S = |T= k,D) represents the conditional probability; P represents the term distribution associated to a topic identified by LDA Model.

The Length of a sentence in a document represents the degree of information it contains. The length of a sentence for information quality is only considered after removing stop words and function words. Product result of calculated probabilities of words may reduce the value to very low for long sentences. So, Product is replaced by summation of calculated probabilities. We get the new formula as :

P(S = |T= k,D) ∑ |

For each sentence, the topic-distribution is determined by the joint probability distribution and it is defined by the conditional probability above and the topic-importance as follows:

$$P(\mathrm{S}=s_j^d \mid \mathrm{D}) = \sum_{k=1}^{K} P(\mathrm{S}=s_j^d, \mathrm{T}=k \mid \mathrm{D})$$

Where P(S = | D) represents the posterior probability of sentences.

A sentence with lower probability words might have a greater value than a shorter sentence having higher probability words. Thus, the posterior probability of sentences is normalized by the sentence length, we calculate the posterior probability as given below:

$$P(\mathrm{S}=\mathrm{s}_j^d) \propto \frac{\sum_{k=1}^{K} \sum_{w_i^d \in s_j^d} P(\mathrm{w}=w_i^d \mid \mathrm{T}=k) P(\mathrm{T}=k)}{\mathrm{Len}(\mathrm{S}=\mathrm{s}_j^d)}$$

Where Len(S=) represents length of the sentence.

Based on this Posterior probability, the top sentences are selected for the final Multi-document summary.

3.3 Multi-Document Summarization Using Sentence Clustering

This Multi-Document text Summarization algorithm[37] uses clustering technique to extract an important piece of information from input documents. The sentence is considered as the most basic entity while performing Text Summarization. Clustering of sentences, paragraphs or text documents are performed on input dataset to produce a good multi-document summary.

This technique aims to produce Multi-document summary based on Single Document Summarization and sentence Clustering. In the algorithm, Single document summaries are produced by preprocessing and feature extraction of each document present in the dataset. The prepared summaries are combined by semantic based sentence clustering. Important sentences to

be selected for final multi-document summary are chosen from these clusters with similar sentences. Non- redundant, coherent and important sentences are extracted for the summary.

The figure below depicts the approach for text summarization of more than one document. As from the figure shown, each input document first undergoes pre-processing and then document features are selected, which contributes in single document summary generation. The individual summaries are further clustered based on sentence similarity. Newly formed cluster‟s sentences are selected to prepare the final multi-document summary. The selected sentences are presented inthe same order as present in the source document. While cluster generating process of single document summaries, semantic and syntactic similarity among different sentences is considered. The semantic similarity of words of the cluster is added together to get the final similarity score between sentences.

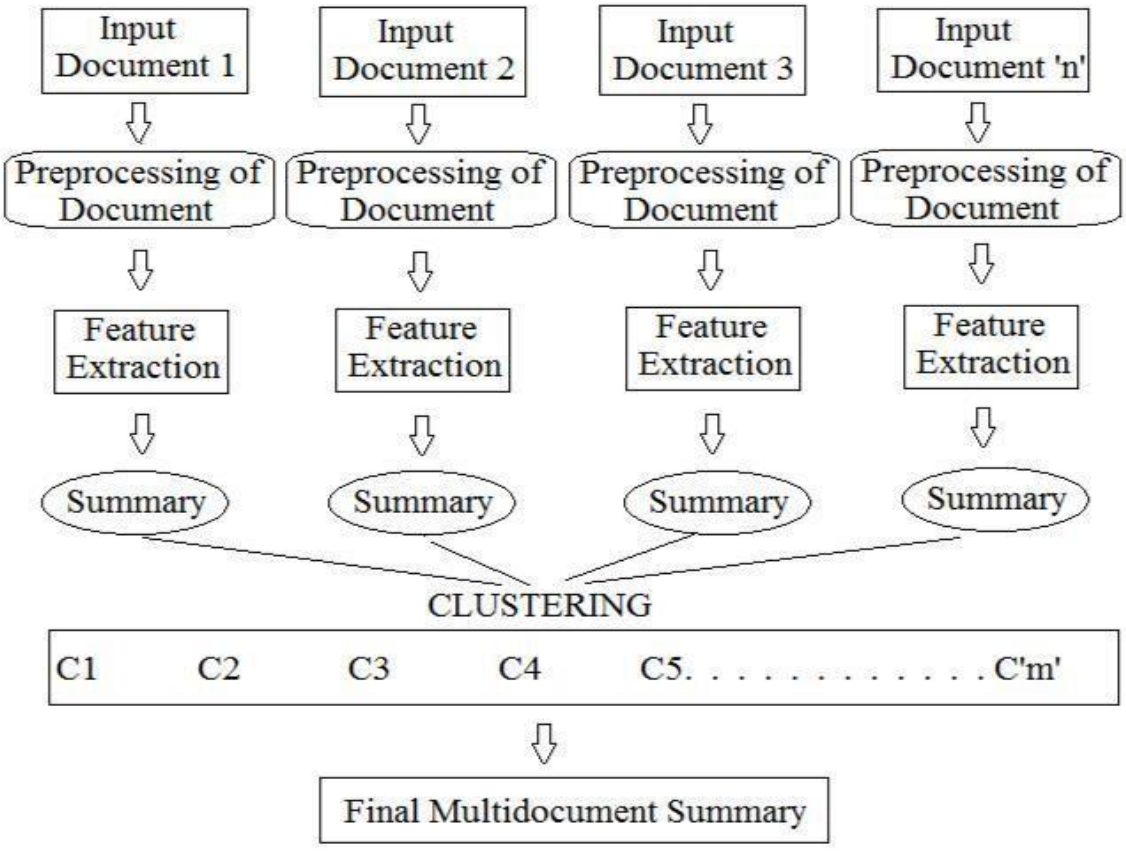

Figure 3.4: Steps in Multi Document Algorithms

The significant steps in multi-documents summarization are as follows:

1. Pre-processing

Preprocessing of input text plays a significant part in text summarization. Removing stopwords, stemming, separate each sentence by finding coreect end to a sentence and tokenization are few important steps to perform Pre-processing. Stopwords need to be separated from input data as they don‟t contribute to quality of summary, hence not considered for final summary generation..

Stemming helps in discovering the root of similar words and to decrease the number of morphological variants. For example , the words like summary, summarize, summarization, summaries all are derived from the word „summary. Various suffixes of the word are removed to reduce ambiguity. Porter Stemmer [38] is used for this algorithm.

To get a Sentence as a unit for different processing, sentence splitting is used which determines where the sentence end. The end markers(. ? !) for sentence splitting may not give desired results in certain cases. Text data like numbers, Abbreviations etc. (7.9, i.e., Ms., Dr., etc.) results in the wrong identification of sentence boundaries. In order to identify correct boundaries simple heuristics and regular expressions are considered. Tokenization explores the text and seperates it intowords, symbols etc.

2. Feature extraction

Feature extraction involves representing text data in form of feature sets. Features are properties of existing data which are useful to identify the importance of words and sentences in the document. Here, the following features are extracted:

Document feature: Each sentence in a document is assigned a weight within a document, which is termed as document feature. The weight of a sentence is calculated by adding the weights of all the content words existing in the document.

$$\text{Document Feature(DF)} = w1+w2+\ldots\ldots\ldots\ldots\ldots\ldots wn$$

Here DF represents the document feature of a whole document and w_i is the symbol for normalized weight associated to the i^{th} word of the sentence. For calculating Document feature, words with normalized frequency greater than a certain value are considered.

Sentence reference index (SRI) feature: Sentence containing pronoun represents a reference to preceding sentence. SRI feature increases the weight of the referred sentence. In Order to recognize such a sentence a list of pronouns is prepared.

Location feature: The location of a sentence in a document is considered while weight calculation. Higher weight is assigned to starting and ending sentences and lower weight to middle paragraph"s sentences. Top and bottom sentences are assumed to have definition and conclusion of the document.

Concept similarity feature: It is defined by the number of synsets associated with query words simlilarity with the sentence. WordNet[13] is used to get a set of synsets for assigning concept similarity weight to a sentence. For example, WordNet gives below synsets for the unit "dog":

Dog: dog, domestic dog, Canine, Carnivore, Mammal, vertebrate, chordate

3. Single document summary generation

The weight of a sentence is evaluated by calculating total sum of individual features as below:

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

Where DF is document feature, LF is location feature, SRI is sentence reference index feature, CS is concept similarity feature and alphabets v, w, x, y are constant values. The constant values are fixed experimentally to v=0.5, w = 0.2, x = 0.2 and y = 0.1 are used to compute SW, is a symbol for sentence weight.

The sentence weights are calculated as shown below:

$$\text{Normalized Weight} = \frac{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}{}$$

Normalized weight is used for ranking of sentences. Top *k* sentences are selected to form single document summary from the source document.

4. Multi-document summary generation

The prepared single document summaries are combined together with the help of sentence clustering and then from each cluster, top k sentences are selected for the formation of the final

multi-document summary. The sentences in the final summary maintain the same order of their position as in the source documents.

Sentence Clustering: Sentence similarity is used to perform clustering of single documentsummaries. The sentence similarity for clustering is calculated using syntactic and semantic similarity measures proposed by Liu [10].

Syntactic Similarity: Liu et al. [10] used a method to calculate the syntactic similarity betweentwo sentences using their word order. Each word is assigned a unique index which is usedto represent an original order (v_0) and a relative order (v_r). The index number of the first sentence represents the original order. Common words in both the sentences are used to create relative order vector.

For example, the original and the relative word order vector for the two sentences *The building istaller than the pole* (S1) and *The pole is taller than the building* (S2) is calculated as below:

Index no. for S_1: {1, 2, 3, 4, 5, 6, 7}

Index no. for S_2: {1, 2, 3, 4, 5, 6, 7}

Original order vector v_0= {1, 2, 3, 4, 5, 6, 7}

Relative order Vector v_r= {1, 7, 3, 4, 5, 6, 2}

Liu et al. [10] used correlation coefficient between the original and relative vector to calculate the similarity:

$$Sim_0(S_1, S_2) = \frac{\sum(v_0 * v_r) - \frac{\sum v_0 * \sum v_r}{k}}{\sqrt{(\sum v_0^2 - \frac{(\sum v_0)^2}{k})(\sum v_r^2 - \frac{(\sum v_r)^2}{k})}}$$

Where k represents the total number of words in an original sentence. Syntactic similarity can have maximum value = 1, when the S1 and S2 word order is identical.

Semantic similarity: Li et al. [8] proposed a method for computing the semantic similarity.First, WordNet [40] is used to calculate the semantic similarity between words. Semantic similarity between sentences is obtained by adding these calculated word similarities. Words are arranged in a Semantic based hierarchy by WordNet.

Semantic similarity between words: Semantic similarity between words is computed by usingan edge count based method. If the words have common features more than different features, they are assumed to be more similar. Common and different features between words are determined by the path length and depth of subsume in Wordnet hierarchy.

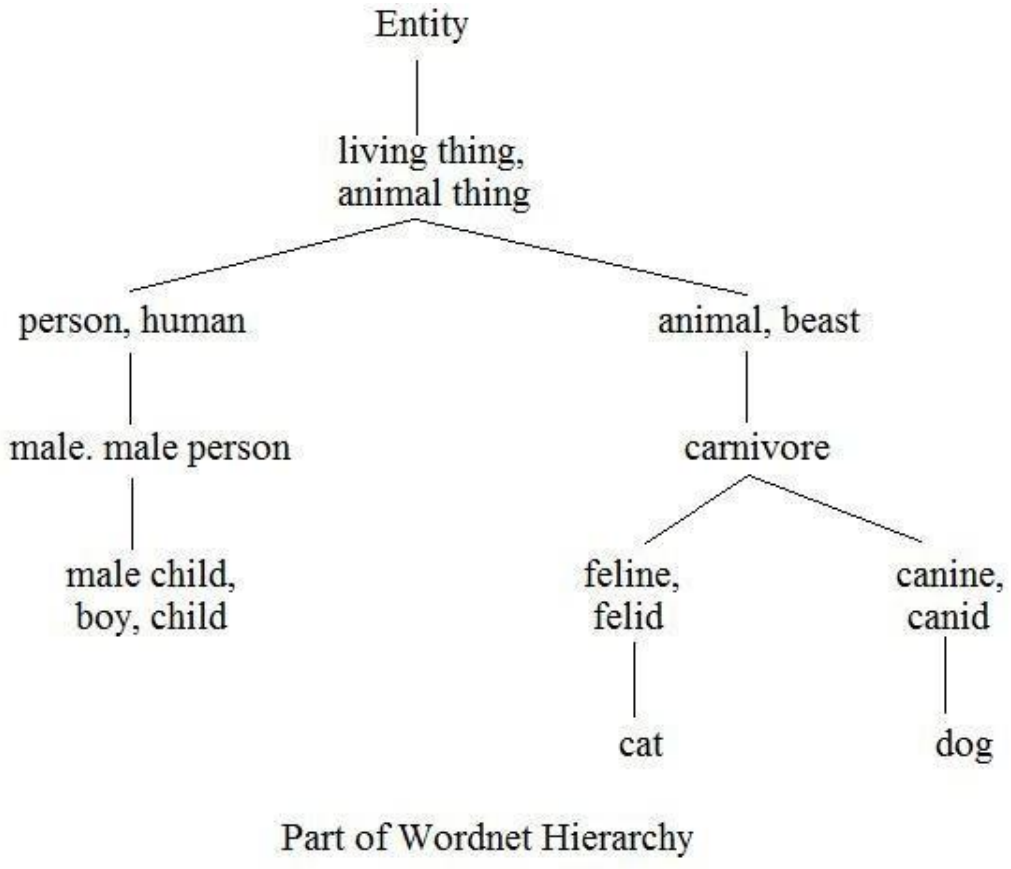

Figure 3.5: Parts of Word net Hierarchy

- **Shortest Path Length (*l*):** It is the shortest path distance between two words in Wordnethierarchy. Lesser the length of the shortest path between two words, more similar they are and vice-versa. For the same words, the shortest path length is 0.

- **Depth of Subsumer**: The depth of subsumer (*d*) is described as the length of thecommon word between two words [8, 10]. The more is the value of depth, the less will be the similarity between the words. The semantic similarity can be calculated as :

$$S_w(w_1, w_2) = \_\_\_\_\_$$

Where d represents the depth of subsumer, l represents shortest path length and f is a transfer function i.e. $f(x) = e^x$-1.

Semantic similarity value may vary between 0 and 1. If the two words are exactly similar then 1 and 0 for dissimilar words.

When d= 0, no common parent, then, $S_w(w_1, w_2) = 0$;

When l = 0, same synset, and $S_w(w_1, w_2) = 1$.

If both d and l are non-zero then the similarity can be calculated as:

$$S_w(w_1, w_2) = \_\_\_\_\_ \quad (0 < \alpha, \beta \leq 1)$$

Where α and β represent smoothing factors.

- **Information Content**: It is a measure of information represented by a word and iscalculated as:

$$I(w) = -\frac{\log p(w)}{\log(N+1)}$$

For calculating the frequency of words British National Corpus [12] is used. The corpus is huge and contains more than 100 million words. The probability of words is computed as:

$$P(w) = \_\_\_\_$$

Where *n* represents the frequency of the word in the corpus and *N* is the total number of words in the corpus.

Finally, the Semantic similarity is calculated by information content and semantic similarity between words, calculated as follows:

$$Sim_s(S_1, S_2) = \frac{\sum_{w_i \in S_1} \max_{w_j \in S_2} \left(S_w(w_i, w_j) * I_{w_i}\right)}{\sum_{w_i \in S_1} I_{w_i} + \sum_{w_j \in S_2} I_{w_j}}$$

Where *I(w)* represents the information content and $S_w(w_1, w_2)$ is the semantic similarity between words. The overall similarity between two sentences is computed as:

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

Where represents smoothing factor.

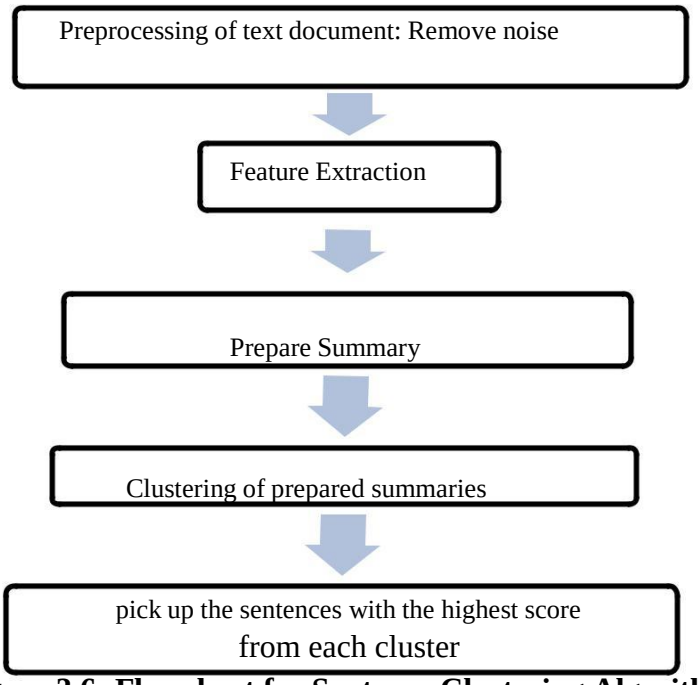

Figure 3.6: Flowchart for Sentence Clustering Algorithm

4. Multi Document Summary: The sentences of single document summaries are clustered usingsentence similarity. From each cluster, top k sentences are selected to form final summary. The extracted sentences are sorted according to their actual position in the original source document to prepare the multi-document summary.

4. COMPARISON AND EVALUATION

This chapter defines the measures: Similarity Score, ROUGE[12] and BLEU [13]metric, used to check the quality and accuracy of the system generated summaries. The accuracy of the single document and Multi-document Summarization algorithms, described in the previous chapters, is evaluated on the basis of these measures.

4.1 Measures

We have examined the summary of all the explained datasets by the previously described algorithms. Then we have evaluated the accuracy of each algorithm generated summary against the set of Human prepared summaries. The human prepared summary is assumed to have the highest accuracy as it includes the human understanding and evaluation. We have checked for the similar words in both the summaries and the provided a similarity score for each one of the algorithms produced summary.

Similarity Score is a measure used for checking similarity among text data. It considers thecommon words and the position of words between system generated summary and human prepared summary. It returns the similarity score value in the range of 0 to 1.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) measure is the most popularmeasure to identify the quality of system generated summary. It is a content overlap measure which tests how an algorithm generated summary matches to the reference summaries produced by human interpretation and understanding. It is a recall-based measure which encourages the algorithms and systems to consider all the key topics in the summary. Recall measure can be calculated by using unigrams, bigrams or trigrams matching. For example, ROUGE-1 is evaluated as a count of unigrams in the system generated summary and reference summary.

If there are multiple summary references, the evaluated ROUGE-1 scores are averaged. ROUGE can only find out if the similar key concepts are discussed between system summary and human generated summary, but the coherence of the sentences cannot be checked. High-order n-gram ROUGE measures try to determine fluency of the summary.

BLEU metric

BLEU metric can be described as a modified form of precision, generally used for machine translation evaluation.

Precision represents the ratio of the number of common words in both gold and model translation/summary to that present in the model summary. Unlike ROUGE, BLEU takes the weighted average and directly accounts for variable length phrases.

The actual metric is just precision which is modified to avoid the problem when a model‟s translation/summary contains redundant information.

4.2 Single Document Summarization Algorithms

This subsection contains the comparison tables with evaluated scores for Single document summarization algorithms for the datasets described in APPENDIX 1.

- □ News blog – Demonetisation Dataset
- □ Medical domain - Alzheimer's Dataset
- □ Cricket related Dataset

Table 4.1, 4.2 and 4.3 represent the similarity score of summaries for different datasets.

Table 4.1 describes the evaluated scores calculated by the Single document summaries prepared by algorithms: TextRank, Texteaser and Summary by word features for News blog - Demonetisation dataset .

Table 4.1: Similarity score for News blog- Demonetisation Dataset

Algorithm	Similarity Score	ROUGE-1	Bleu metric
TextRank	0.72	0.576	0.269
Texteaser	0.58	0.473	0.311
Smmry tool	0.52	0.475	0.206

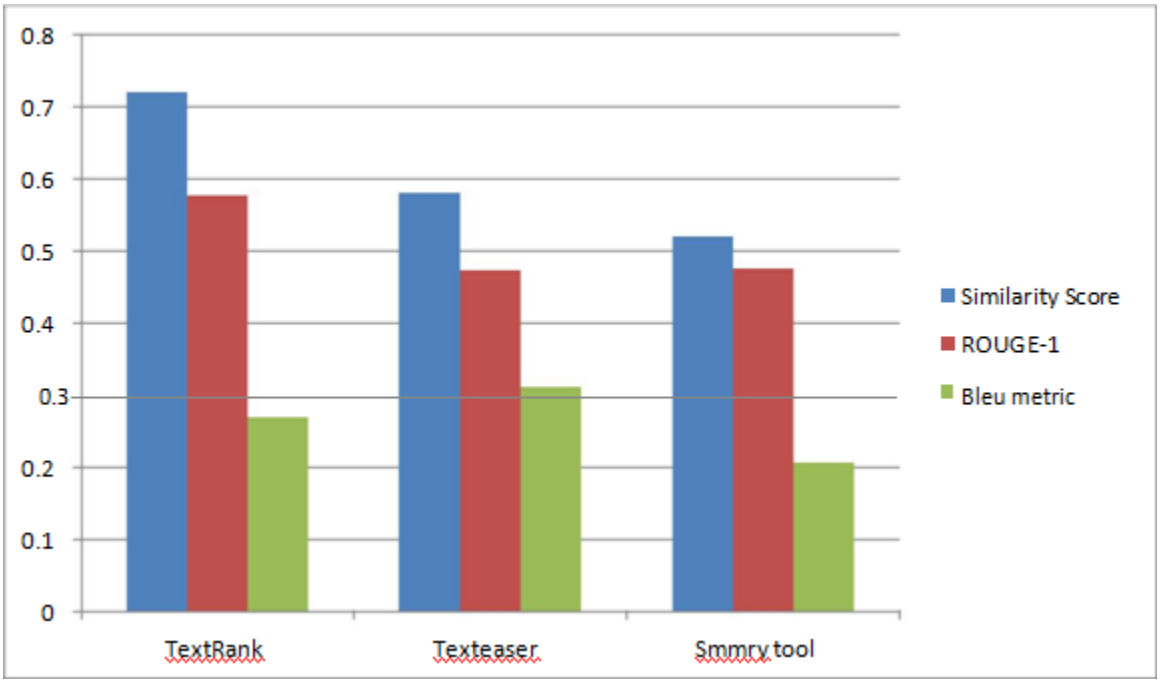

Figure 4.1: Graphical representation-News blog- Demonetisation Dataset

Table4.2 describes the evaluated scores calculated by the Single document summaries prepared by algorithms: TextRank, Texteaser and Summary by word features for Medical- Alzheimer‟sdataset .

Table 4.2: Similarity score for Medical- Alzheimer's Dataset

Algorithm	Similarity Score	ROUGE-1	Bleu metric
TextRank	0.298	0.263	0.197
Texteaser	0.411	0.357	0.251
Smmry tool	0.493	0.417	0.323

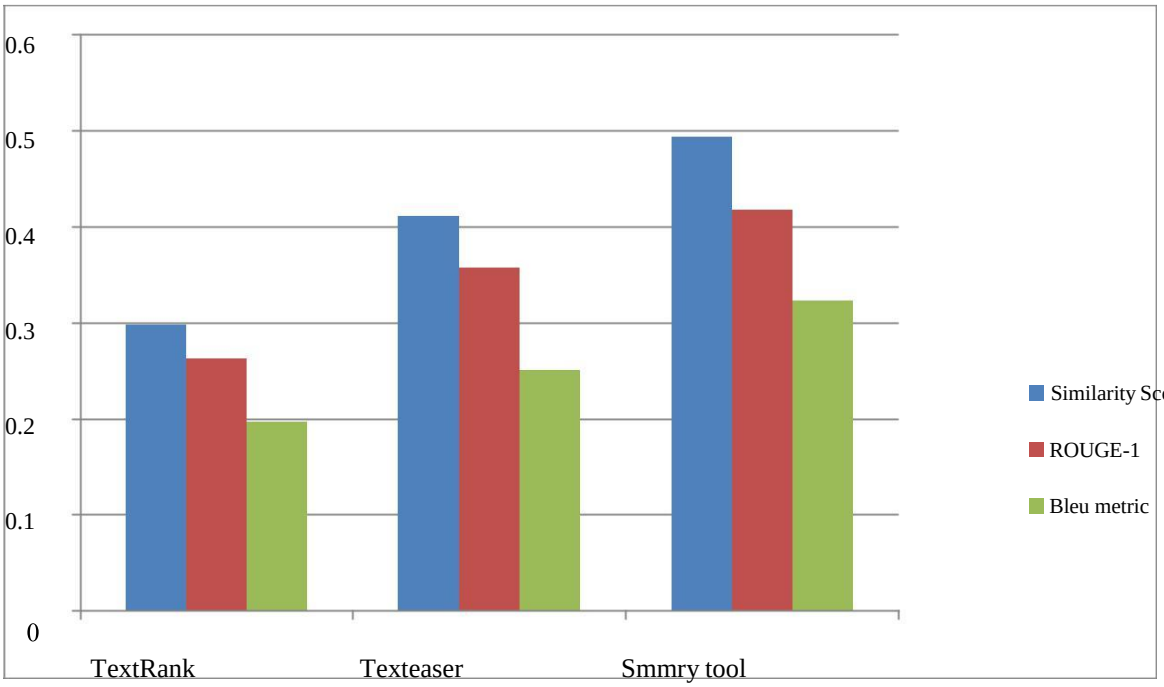

Figure 4.2: Graphical representation- Medical- Alzeihmer's Dataset

Table 4.3 and Figure 4.3 describes the evaluated scores calculated by the Single document summaries prepared by algorithms: TextRank, Texteaser and Summary by word features for Cricket related dataset .

Table4.3: Similarity score for Cricket related Dataset

Algorithm	Similarity Score	ROUGE-1	Bleu metric
TextRank	0.51	0.436	0.255
Texteaser	0.39	0.298	0.134
Smmry tool	0.39	0.264	0.211

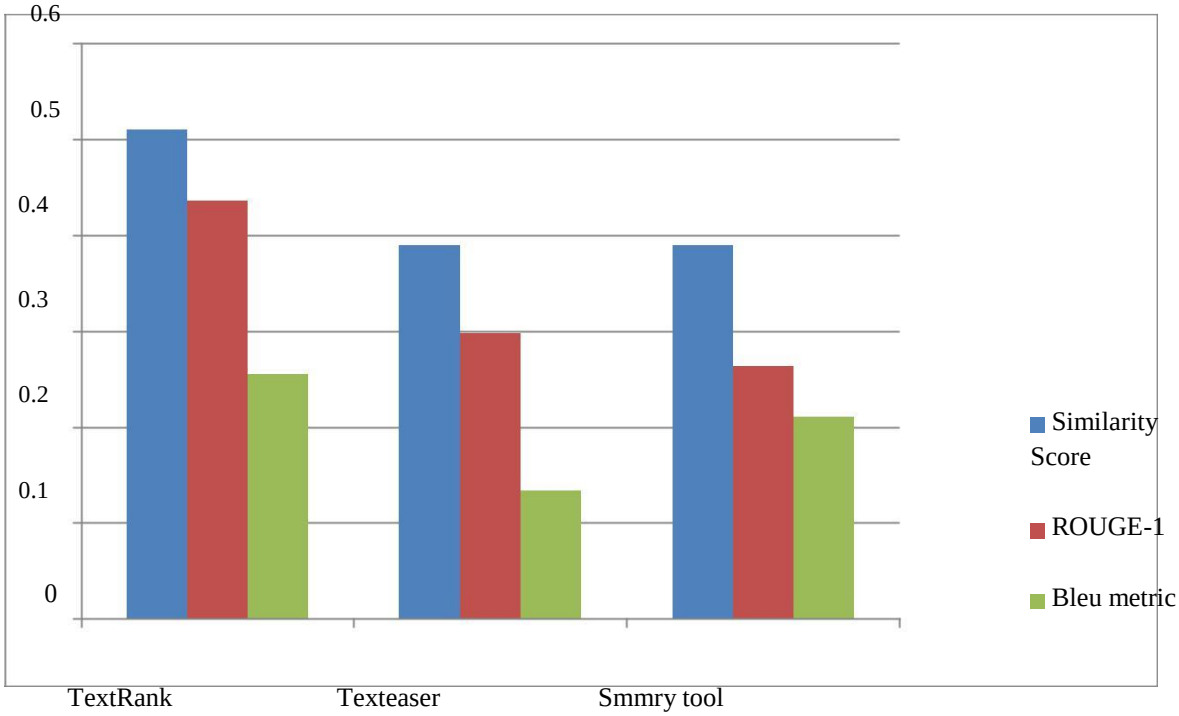

Figure 4.3: Graphical representation- Cricket related Dataset

From these tables, we have analyzed that the cricket domain data is best summarized by the TextRank Algorithm.

For medical dataset, Summary based on word features gives similarity score of 0.51, best among all other algorithms.

For News related dataset, text Rank gives 0.72 similarity score.

4.2 Multi-document Summarization Algorithms

The tables 4.4, 4.5 and 4.6 represent the similarity score of summaries for different datasets.

Table 4.4 describes the evaluated scores calculated by the Single document summaries prepared by algorithms: Multi-document Summarization using ILP based method,LDA topic model and Summarization based on sentence clustering for News blog-Demonetisation Dataset.

Table 4.4: Similarity score for News blog- Demonetisation Dataset

Algorithm based on	**Similarity Score**	**ROUGE-1**	**Bleu metric**
ILP based Sentence Fusion	0.24	0.29	0.18
LDA topic Model	0.38	0.431	0.34
Sentence Clustering	0.402	0.41	0.14

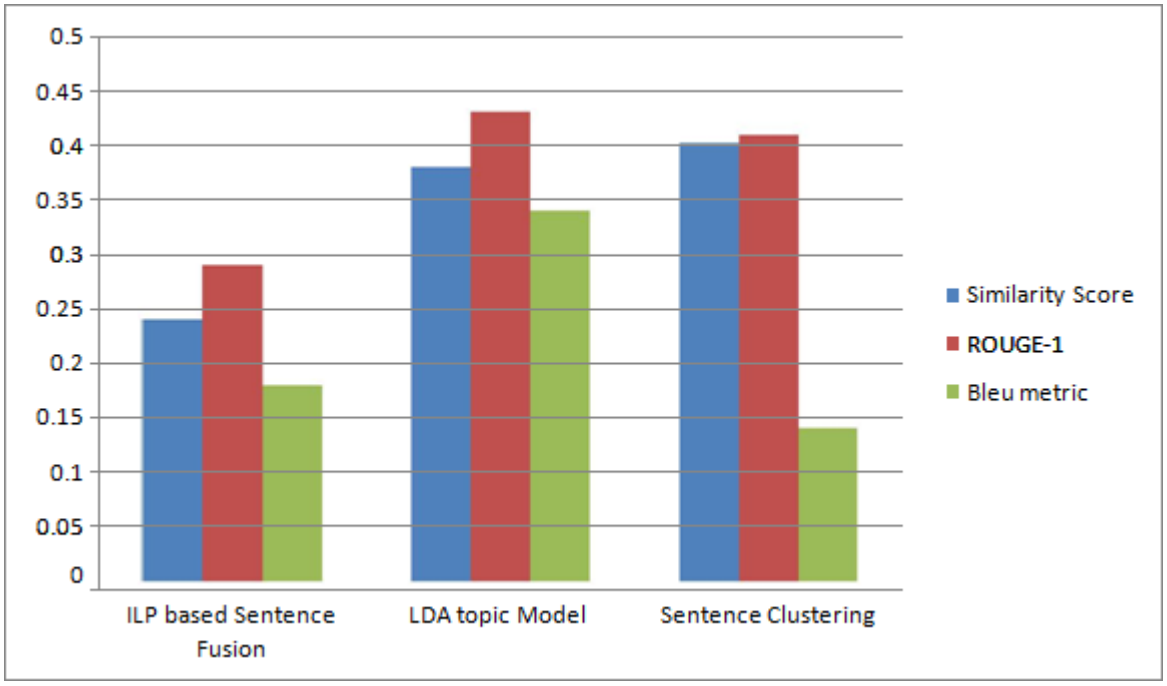

Figure 4.4: Graphical Representation-News blog- Demonetisation Dataset

Table 4.48 describes the evaluated scores calculated by the Single document summariesprepared by algorithms: Multi-document Summarization using ILP based topic model and Summarization based on sentence clustering for Medical- Alzheimer's Dataset.

Table 4.5: Similarity Score for Medical- Alzheimer's Dataset

Algorithm based on	Similarity Score	ROUGE-1	Bleu metric
ILP based Sentence Fusion	0.21	0.23	0.17
LDA topic Model	0.27	0.35	0.19
Sentence Clustering	0.43	0.34	0.28

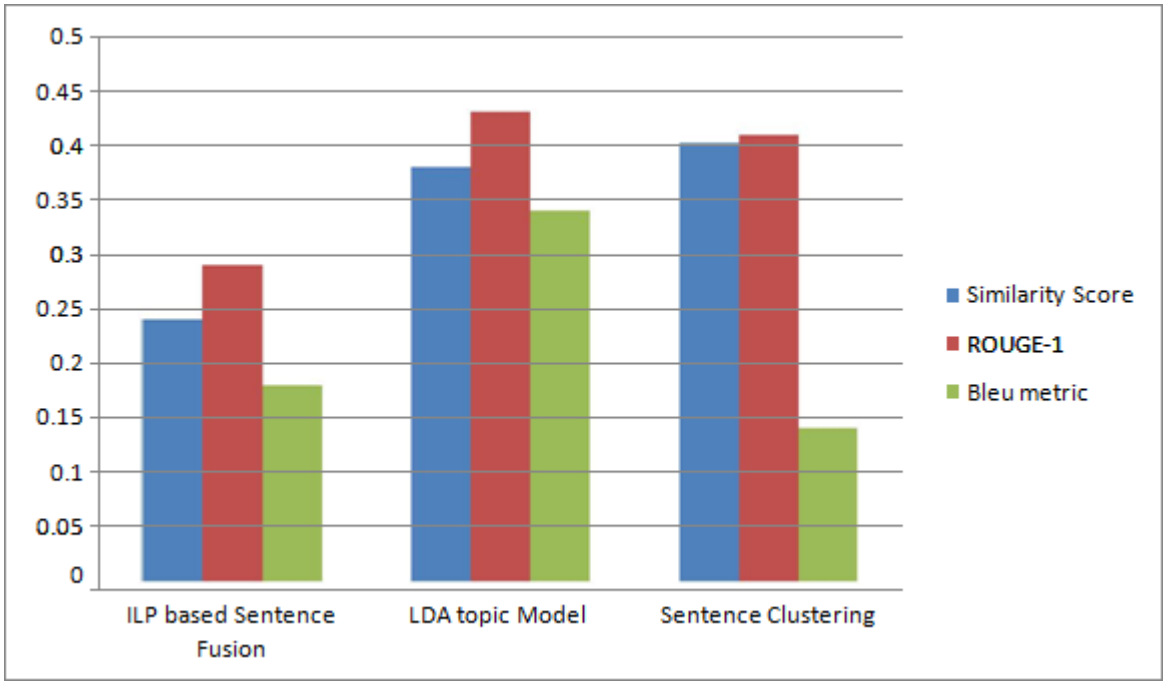

Figure 4.5: Graphical Representation- Medical- Alzeihmer's Dataset

Table 4.6 describes the evaluated scores calculated by the Single document summaries prepared by algorithms: Multi-document Summarization using ILP based method, LDA topic model and Summarization based on sentence clustering for Cricket related Dataset.

Table 4.6: Similarity Score for Cricket related Dataset

Algorithm based on	Similarity Score	ROUGE-1	Bleu metric
ILP based Sentence Fusion	0.18	0.21	0.11
LDA topic Model	0.29	0.31	0.24
Sentence Clustering	0.36	0.35	0.31

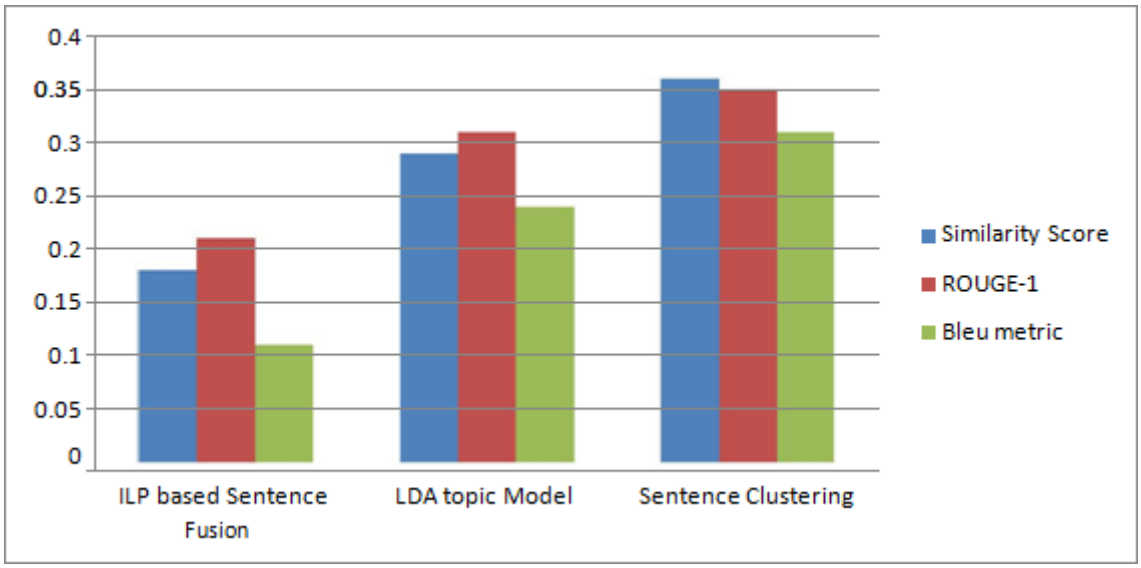

Figure 4.6: Graphical Representation- Cricket Related Dataset

From the above three tables, we have analyzed that the News domain- Demonetisation data is best summarized by the Multi-document Summarization based on LDA Topic Model.

For medical dataset and cricket dataset, Multi-document Summarization based on Sentence Clustering outperforms the other two algorithms.

Sentence Clustering based algorithm gives better results because of its sentence clustering of single document summaries and extractive nature.

5. CONCLUSIONS AND FUTURE SCOPE

Automatic Text Summarization is used to get an important piece of text from a largerdocument. A large number of algorithms designed and implemented to get a good, coherent and non-redundant summary a little similar to the human prepared summary.

Simple single document extractive algorithms have given better results in different domains as compared to abstractive summarization algorithms.

Extractive summarizers are used to select the important set of sentences from the source document based on top scoring Sentence-ranking method. These methods use different feature extraction and content selection methods like upper case words, the frequency of words, similarity chains, logical closeness etc. for selecting summary sentences.

Abstractive Summarizers prepare new sentences based on the important information existing in the source document. These summarizers make new sentences by the union of multiple sentences. They use word graphs to select a set of words to produce a coherent sentence.

Based on the Comparison results, it can be seen that by performing Automatic Text Summarization to get a gist of the input text documents equivalent to human interpreted summary is not yet fulfilled, but by improving the existing algorithms, the value of evaluation metrics is increasing.

Future Scope

With the rapid increase in the electronic data on internet and less time to read the documents based on a similar topic has called a need to design accurate and efficient Multi- document summarization systems. As research on text summarization started 50 years ago and a lot of work has been done in the extractive area in both the single and multiple document domains but there is still a long path to cover in this field. Abstractive summarizers aim to import more information in a single sentence rather than include the sentence as a whole.

Multi-document Abstractive Summarization is the area which is needed to be explored.

Over time, attention has drifted from summarizing scientific articles to news articles, electronic mail messages, advertisements, and blogs. Domain associated summarizes can be asolution to get more accurate summaries. Medical and Legal matters domain can be highly benefitted from this area of research even if they focus only on small details related to a general summarization process and not on building an entire domain dependent summarization system.

APPENDIX-I

Datasets

In this Chapter, The input datasets are defined:

Dataset 1:

Dataset 1 includes the text files containing the data from newspapers, internet and news blogs regarding the news demonetization. It involves the date on which notes of 500 and 1000 were declared illegal tender. It includes the date on which it was announced. Various rules imposed time to time and expert views on the move. Problems faced by the citizens and rules formed to facilitate the people are also specified. Total collection and immediate effects are also stated in the articles.

Number of characters (including spaces) :	40362
Number of characters (without spaces) :	32625
Number of words :	6728
Lexical Density :	27.1106
Number of sentences :	417
Number of syllables :	11214

Output Summary

Here, we expect the summary to include how it started, effects and final results and reason associated with the news.

Dataset 2:

Dataset 2 is a medical related data about a disease called Alzheimer's. Alzheimer's is a disease which causes the patient to forget about the things and then it proceeds to a point

where patient may no longer recognize their family members or the thing which has been done even before 5 minutes. The dataset includes the definition and introduction to the problem. Then it analyses the causes associated which are likely to cause Alzheimer's. It also includes the cure and how to approach the disease in the first phase. Different phases are described with the help of conditions of a patient.

Number of characters (including spaces) :	37045
Number of characters (without spaces) :	29515
Number of words :	5886
Lexical Density :	23.1057
Number of sentences :	356
Number of syllables :	10079

Output Summary

For this dataset, the summary is expected to include a brief introduction to the disease and then a little information about the different phases and cure for the problem.

Dataset 3:

This dataset includes cricket related data. It includes the history of cricket in India how it started and various milestones achieved in the times. About The Indian team lifted the World Cup and a little about the prominent players. It also includes the cricket control bodies on Indian as well as the international level i. e. the ICC and BCCI, and about how they work andorganize the international tournaments regularly. Our current team captains and teammates related data are also included.

Number of characters (including spaces) :	29302
Number of characters (without spaces) :	23494
Number of words :	5046
Lexical Density :	23.8605
Number of sentences :	234
Number of syllables :	8129

Output Summary

Here for this dataset we expect the summary to include a little history of the cricket and years when the Indian team won the international tournaments. A little about the control bodies and current cricket team and coach shall also be included in the summary.