A Survey of Text Summarization Extractive Techniques

Article *in* Journal of Emerging Technologies in Web Intelligence · August 2010

DOI: 10.4304/jetwi.2.3.258-268

2 authors:

Vishal Gupta
Panjab University
41 PUBLICATIONS **2,425** CITATIONS
SEE PROFILE

Gurpreet Lehal
Punjabi University, Patiala
98 PUBLICATIONS **2,631** CITATIONS
SEE PROFILE

A Survey of Text Summarization Extractive Techniques

Vishal Gupta
University Institute of Engineering & Technology,
Computer Science & Engineering, Panjab University Chandigarh, India,
Email: vishal@pu.ac.in

Gurpreet Singh Lehal
Department of Computer Science,
Punjabi University Patiala, Punjab, India,
Email: gslehal@yahoo.com

***Abstract*— Text Summarization is condensing the source text into a shorter version preserving its information content and overall meaning. It is very difficult for human beings to manually summarize large documents of text. Text Summarization methods can be classified into extractive and abstractive summarization. An extractive summarization method consists of selecting important sentences, paragraphs etc. from the original document and concatenating them into shorter form. The importance of sentences is decided based on statistical and linguistic features of sentences. An abstractive summarization method consists 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. In this paper, a Survey of Text Summarization Extractive techniques has been presented.**

***Index Terms*—Text Summarization, extractive summary, abstractive summary**

I. INTRODUCTION

Text summarization [1] has become an important and timely tool for assisting and interpreting text information in today's fast-growing information age. It is very difficult for human beings to manually summarize large documents of text. There is an abundance of text material available on the internet. However, usually the Internet provides more information than is needed. Therefore, a twofold problem is encountered: searching for relevant documents through an overwhelming number of documents available, and absorbing a large quantity of relevant information. The goal of automatic text summarization is condensing the source text into a shorter version preserving its information content and overall meaning.

A summary [4] can be employed in an indicative way as a pointer to some parts of the original document, or in an informative way to cover all relevant information of the text. In both cases the most important advantage of using a summary is its reduced reading time. A good summary system should reflect the diverse topics of the document while keeping redundancy to a minimum. Summarization tools may also search for headings and other markers of subtopics in order to identify the key points of a document. Microsoft Word's AutoSummarize function is a simple example of text summarization.

Text Summarization methods can be classified into extractive and abstractive summarization. An extractive summarization method consists of selecting important sentences, paragraphs etc. from the original document and concatenating them into shorter form. The importance of sentences is decided based on statistical and linguistic features of sentences.

An Abstractive summarization [32][33] attempts to develop an understanding of the main concepts in a document and then express those concepts in clear natural language. 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. This paper focuses on extractive text summarization methods.

Extractive summaries [2]are formulated by extracting key text segments (sentences or passages) from the text, based on statistical analysis of individual or mixed surface level features such as word/phrase frequency, location or cue words to locate the sentences to be extracted. The "most important" content is treated as the "most frequent" or the "most favorably positioned" content. Such an approach thus avoids any efforts on deep text understanding. They are conceptually simple, easy to implement.

Extractive text summarization process [31] can be divided into two steps: 1) Pre Processing step and 2) Processing step.

Pre Processing is structured representation of the original text. It usually includes: a) Sentences boundary identification. In English, sentence boundary is identified with presence of dot at the end of sentence. b) Stop-Word Elimination—Common words with no semantics and

Manuscript received January 12, 2010; revised March 22, 2010; accepted April 29, 2010.
Corresponding author: Vishal Gupta

doi:10.4304/jetwi.2.3.258-268

which do not aggregate relevant information to the task are eliminated. c) Stemming—The purpose of stemming is to obtain the stem or radix of each word, which emphasize its semantics.

In Processing step, features influencing the relevance of sentences are decided and calculated and then weights are assigned to these features using weight learning method. Final score of each sentence is determined using Feature-weight equation. Top ranked sentences are selected for final summary.

Problems with the extractive summary [46] [47] are:

1. Extracted sentences usually tend to be longer than average. Due to this, parts of the segments that are not essential for summary also get included, consuming space.
2. Important or relevant information is usually spread across sentences, and extractive summaries cannot capture this (unless the summary is long enough to hold all those sentences).
3. Conflicting information may not be presented accurately.
4. Pure extraction often leads to problems in overall coherence of the summary—a frequent issue concerns "dangling" anaphora. Sentences often contain pronouns, which lose their referents when extracted out of context. Worse yet, stitching together decontextualized extracts may lead to a misleading interpretation of anaphors (resulting in an inaccurate representation of source information, i.e., low fidelity). Similar issues exist with temporal expressions. These problems become more severe in the multi-document case, since extracts are drawn from different sources. A general approach to addressing these issues involves post-processing extracts, for example, replacing pronouns with their antecedents, replacing relative temporal expression with actual dates, etc.

Problems with the abstractive summary [46] are:

The biggest challenge for abstractive summary is the representation problem. Systems' capabilities are constrained by the richness of their representations and their ability to generate such structures—systems cannot summarize what their representations cannot capture. In limited domains, it may be feasible to devise appropriate structures, but a general-purpose solution depends on open-domain semantic analysis. Systems that can truly "understand" natural language are beyond the capabilities of today's technology.

Summary evaluation [34][36][37] is a very important aspect for text summarization. Generally, summaries can be evaluated using intrinsic or extrinsic measures. While intrinsic methods attempt to measure summary quality using human evaluation and extrinsic methods measure the same through a task-based [35] performance measure such the information retrieval-oriented task.

Newsblaster is a good example of a text summarizer, that helps users find the news that is of the most interest to them. The system automatically collects, clusters, categorizes, and summarizes news from several sites on the web (CNN, Reuters, Fox News, etc.) on a daily basis, and it provides users a user-friendly interface to browse the results.

II. TEXT SUMMARIZATION EARLY HISTORY

Interest in automatic text summarization, arose as early as the fifties. An important paper of these days is the one in 1958, suggested to weight the sentences of a document as a function of high frequency words[7], disregarding the very high frequency common words. Automatic text summarization system [8] in 1969, which, in addition to the standard keyword method (i.e., frequency depending weights), also used the following three methods for determining the sentence weights:

1. Cue Method: This is based on the hypothesis that the relevance of a sentence is computed by the presence or absence of certain cue words in the cue dictionary.
2. Title Method: Here, the sentence weight is computed as a sum of all the content words appearing in the title and (sub-) headings of a text.
3. Location Method: This method is based on the assumption that sentences occurring in initial position of both text and individual paragraphs have a higher probability of being relevant. The results showed, that the best correlation between the automatic and human-made extracts was achieved using a combination of these three latter methods.

The Trainable Document Summarizer [9] in 1995 performs sentence extracting task, based on a number of weighting heuristics. Following features were used and evaluated:

1. Sentence Length Cut-O Feature: sentences containing less than a pre-specified number of words are not included in the abstract
2. Fixed-Phrase Feature: sentences containing certain cue words and phrases are included
3. Paragraph Feature: this is basically equivalent to Location Method feature in [8]
4. Thematic Word Feature: the most frequent words are defined as thematic words. Sentence scores are functions of the thematic words' frequencies
5. Uppercase Word Feature: upper-case words (with certain obvious exceptions) are treated as thematic words, as well.

A Corpus was used in this method, which contained 188 document/summary pairs from 21 publications in a scientific/technical domain. The summaries were produced by professional experts and the sentences occurring in the summaries were aligned to the original document texts, indicating also the degree of similarity as mentioned earlier, the vast majority (about 80%) of the summary sentences could be classified as direct sentence matches.

The ANES text extraction system [10] in 1995 is a system that performs automatic, domain-independent condensation of news data. The process of summary generation has four major constituents:

1. Corpus analysis: this is mainly a calculation of the tf*idf -weights for all terms

2. Statistical selection of signature words: terms with a high tf*idf-weight
plus headline-words
3. Sentence weighting: summing over all signature word weights, modifying the weights by some other factors, such as relative location
4. Sentence selection: Selecting high scored sentences.

Hidden Markov Models (HMMs) [11]: As prove to be a mathematically sound frame-work for document retrieval. If one approaches the task of text abstracting from such a probabilistic modeling perspective, it might well be possible that HMMs could be employed for this purpose, as well.

Clustering: Building links [12] and/or clusters between index terms, phrases and/or other subparts of the documents has been employed by standard information retrieval. Although this is not an issue in any of the above mentioned abstracting systems, it seems to be worth of consideration when building such systems.

III. FEATURES FOR EXTRACTIVE TEXT SUMMARIZATION

Some features [2][5][29] to be considered for including a sentence in final summary are:

A. Content word (Keyword) feature:

Content words or Keywords are usually nouns and determined using tf × idf measure. Sentences having keywords are of greater chances to be included in summary. Another keyword extraction method [23][31] is given below, having three modules:
1) Morphological Analysis
2) Noun Phrase (NP) Extraction and Scoring
3) Noun Phrase (NP) Clustering and Scoring
Figure1 shows a pictorial representation of the keyword extraction method.

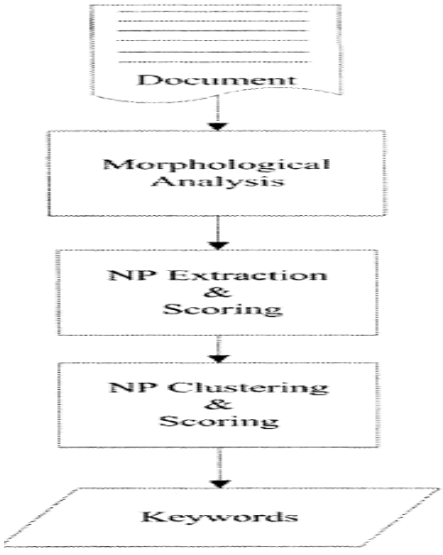

Figure 1. Keyword extraction method

B. Title word feature:

Sentences containing words that appear in the title are also indicative of the theme of the document. These sentences are having greater chances for including in summary.

C. Sentence location feature:

Usually first and last sentence of first and last paragraph of a text document are more important and are having greater chances to be included in summary.

D. Sentence Length feature:

Very large and very short sentences are usually not included in summary.

E. Proper Noun feature:

Proper noun is name of a person, place and concept etc. Sentences containing proper nouns are having greater chances for including in summary.

F. Upper-case word feature:

Sentences containing acronyms or proper names are included.

G. Cue-Phrase Feature:

Sentences containing any cue phrase (e.g. “in conclusion”, “this letter”, “this report”, “summary”, “argue”, “purpose”, “develop”, “attempt” etc.) are most likely to be in summaries.

H. Biased Word Feature:

If a word appearing in a sentence is from biased word list, then that sentence is important. Biased word list is previously defined and may contain domain specific words.

I. Font based feature:

Sentences containing words appearing in upper case, bold, italics or Underlined fonts are usually more important.

J. Pronouns:

Pronouns such as “she, they, it” cannot be included in summary unless they are expanded into corresponding nouns.

K. Sentence-to-Sentence Cohesion:

For each sentence s compute the similarity between s and each other sentence s’ of the document, then add up those similarity values, obtaining the raw value of this feature for s. The process is repeated for all sentences.

L. Sentence-to-Centroid Cohesion:

For each sentence s as compute the vector representing the centroid of the document, which is the arithmetic average over the corresponding coordinate values of all the sentences of the document; then compute the similarity between the centroid and each sentence, obtaining the raw value of this feature for each sentence.

M. Occurrence of non-essential information:

Some words are indicators of non-essential information. These words are speech markers such as "because", "furthermore", and "additionally", and typically occur in the beginning of a sentence. This is also a binary feature, taking on the value "true" if the sentence contains at least one of these discourse markers, and "false" otherwise.

N. Discourse analysis:

Discourse level information [38], in a text is one of good feature for text summarization. In order to produce a coherent, fluent summary, and to determine the flow of the author's argument, it is necessary to determine the overall discourse structure of the text and then removing sentences peripheral to the main message of the text.

These features are important as, a number of methods of text summarization are using them. These features are covering statistical and linguistic characteristics of a language.

IV. EXTRACTIVE SUMMARIZATION METHODS

Extractive summarizers [13][14][30] aim at picking out the most relevant sentences in the document while also maintaining a low redundancy in the summary.

A. Term Frequency-Inverse Document Frequency (TF-IDF) method:

Bag-of-words model is built at sentence level, with the usual weighted term-frequency and inverse sentence-frequency paradigm [16], where sentence-frequency is the number of sentences in the document that contain that term. These sentence vectors are then scored by similarity to the query and the highest scoring sentences are picked to be part of the summary. This is a direct adaptation of Information Retrieval paradigm to summarization. Summarization is query-specific, but can be adapted to be generic as described below.

To generate a generic summary, non stop-words that occur most frequently in the document(s) may be taken as the query words. Since these words represent the theme of the document, they generate generic summaries. Term-frequency is usually 0 or 1 for sentences—since normally the same content-word does not appear many times in a given sentence. If users create query words the way they create for information retrieval, then the query based summary generation would become generic summarization.

B. Cluster based method:

Documents are usually written such that they address different topics one after the other in an organized manner. They are normally broken up explicitly or implicitly into sections. This organization applies even to summaries of documents. It is intuitive to think that summaries should address different "themes" appearing in the documents. Some summarizers incorporate this aspect through clustering. If the document collection for which summary is being produced is of totally different topics, document clustering becomes almost essential to generate a meaningful summary.

Documents are represented using term frequency-inverse document frequency (TF-IDF) [17] of scores of words. Term frequency used in this context is the average number of occurrences (per document) over the cluster. IDF value is computed based on the entire corpus. The summarizer takes already clustered documents as input. Each cluster is considered a theme. The theme is represented by words with top ranking term frequency, inverse document frequency (TF-IDF) scores in that cluster.

Sentence selection is based on similarity of the sentences to the theme of the cluster C_i .The next factor that is considered for sentence selection is the location of the sentence in the document (L_i). In the context of newswire articles, the closer to the beginning a sentence appears, the higher its weight age for inclusion in summary. The last factor that increases the score of a sentence is its similarity to the first sentence in the document to which it belongs (F_i).

The overall score (Si) of a sentence i is a weighted sum of the above three factors:

$$S_i = W_1 * C_i + W_2 * F_i + W_3 * L_i \quad \text{.....(2)}$$

where S_i is the score of sentence $C_{i,,}$ F_i are the scores of the sentence i based on the similarity to theme of cluster and first sentence of the document it belongs to, respectively. L_i is the score of the sentence based on its location in the document. $w_{1,}$ w_2 andw_3 are the weights for linear combination of the three scores. Note the similarity between the sentence score in equations (1) and (2). The role of F in (2) is similar to that of T in (1). The difference however, is that $S_{i,}$ in (2) is further re-scored using a redundancy factor. Once the documents are clustered, sentence selection from within the cluster to form its summary is local to the documents in the cluster. The IDF value based on the corpus statistics seems counter-intuitive. A better choice may be to take the Average-TF alone to determine the theme of the cluster, and then rely on the "anti redundancy" factor to cover the important 'themes' within the cluster.

C. Graph theoretic approach:

As seen in the previous methods, the first step involved in the process of summarizing one or more documents is identifying the issues or topics addressed in the document. Graph theoretic representation [18] of passages provides a method of identification of these themes. After the common preprocessing steps, namely, stop word removal and stemming, sentences in the documents are represented as nodes in an undirected graph.

There is a node for every sentence. Two sentences are connected with an edge if the two sentences share some common words, or in other words, their (cosine, or such) similarity is above some threshold. This representation yields two results: The partitions contained in the graph (that is those sub-graphs that are unconnected to the other sub graphs), form distinct topics covered in the documents. This allows a choice of coverage in the summary. For query-specific summaries, sentences may be selected only from the pertinent sub graph, while for generic summaries, representative sentences may be chosen from each of the sub-graphs.

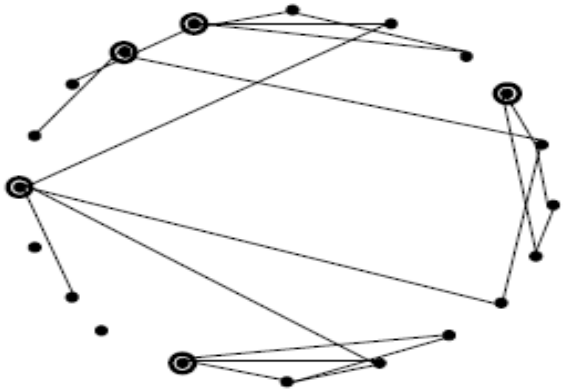

Figure 2. Graph theoretic approach

The second result yielded by the graph-theoretic method is the identification of the important sentences in the document. The nodes with high cardinality (number of edges connected to that node), are the important sentences in the partition, and hence carry higher preference to be included in the summary. Figure2 shows an example graph for a document. It can be seen that there are about 3-4 topics in the document; the nodes that are encircled can be seen to be informative sentences in the document, since they share information with many other sentences in the document. The graph theoretic method may also be adapted easily for visualization of inter- and intra-document similarity.

D. Machine Learning approach

Given a set of training document and their extractive summaries, the summarization process is modeled as a classification problem: sentences are classified as summary sentences and non-summary sentences based on the features that they possess. The classification probabilities are learnt statistically [3] from the training data, using Bayes' rule:

$$P(s \in S \mid F1, F2, ..., FN) = P(F1, F2, ..., FN \mid s \in S) * P(s \in S) / P(F1, F2, ..., FN)$$

where s is a sentence from the document collection, F1, F2…FN are features used in classification. S is the summary to be generated, and P (s∈< S | F1, F2, ..., FN) is the probability that sentence s will be chosen to form the summary given that it possesses features F1,F2…FN.

E. LSA Method

Singular Value Decomposition (SVD) [13] is a very powerful mathematical tool that can find principal orthogonal dimensions of multidimensional data. It has applications in many areas and is known by different names: Karhunen-Loeve Transform in image processing, Principal Component Analysis (PCA) in signal processes and Latent Semantic Analysis (LSA) in text processing. It gets this name LSA because SVD applied to document-word matrices, groups documents that are semantically related to each other, even when they do not share common words.

Words that usually occur in related contexts are also related in the same singular space. This method can be applied to extract the topic-words and content-sentences from documents. The advantage of using LSA vectors for summarization rather than the word vectors is that conceptual (or semantic) relations as represented in human brain are automatically captured in the LSA, while using word vectors without the LSA transformation requires design of explicit methods to derive conceptual relations. Since SVD finds principal and mutually orthogonal dimensions of the sentence vectors, picking out a representative sentence from each of the dimensions ensures relevance to the document, and orthogonality ensures non-redundancy. It is to be noted that this property applies only to data that has principal dimensions inherently—however, LSA would probably work since most of the text data has such principal dimensions owing to the variety of topics it addresses.

F. An approach to concept-obtained text summarization

The idea of this approach is to obtain concepts of words based on HowNet [19][20], and use concept as feature, instead of word. This approach uses conceptual vector space model to form a rough summarization, and then calculate degree of semantic similarity of sentence for reducing its redundancy. A good summary system should extract the diverse topics of the document while keeping redundancy to a minimum. This method consists of the following three main stages:

Stage 1: Using Hownet as tool to obtain concept of text, and establishing conceptual vector space model.

Stage 2: Calculate importance of concept based on conceptual vector space model.

Stage 3: Generate the final summary by calculating importance of sentence and reducing the redundancy of summarization.

G. Text summarization with neural networks

This method involves training the neural networks to learn the types of sentences that should be included in the summary. This is accomplished by training the network with sentences in several test paragraphs where each

sentence is identified as to whether it should be included in the summary or not. This is done by a human reader. The neural network [21] learns the patterns inherent in sentences that should be included in the summary and those that should not be included. It uses three-layered Feed forward neural network, which has been proven to be a universal function approximator.

The first phase of the process involves training the neural networks to learn the types of sentences that should be included in the summary. This is accomplished by training the network with sentences in several test paragraphs where each sentence is identified as to whether it should be included in the summary or not. This is done by a human reader. The neural network learns the patterns inherent in sentences that should be included in the summary and those that should not be included. The Neural Network [27] after Training is shown in figure3.

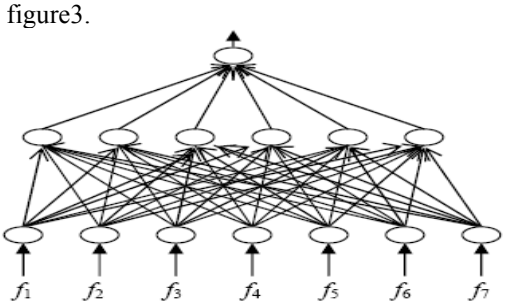

Figure 3. Neural Network after Training

Once the network has learned the features that must exist in summary sentences, we need to discover the trends and relationships among the features that are inherent in the majority of sentences. This is accomplished by the feature fusion phase, which consists of two steps: 1) eliminating uncommon features; and 2) collapsing the effects of common features. The connections having very small weights after training can be pruned without affecting the performance of the network. As a result, any input or hidden layer neuron having no emanating connections can be safely removed from the network. In addition, any hidden layer neuron having no abutting connections can be removed. This corresponds to eliminating uncommon features from the network [27] as shown in figure4.

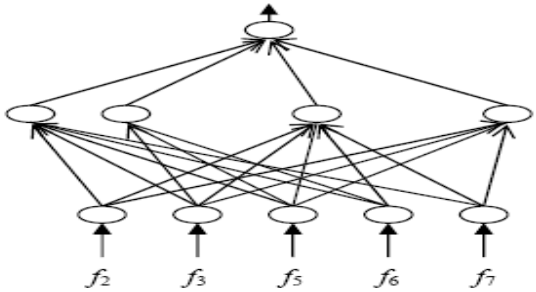

Figure 4. Neural Network after Pruning

The hidden layer activation values for each hidden layer neuron are clustered utilizing an adaptive clustering technique. Each cluster is identified by its centroid and frequency. The activation value of each hidden layer neuron is replaced by the centroid of the cluster, which the activation value belongs to. This corresponds to collapsing the effects of common features. The combination of these two steps corresponds to

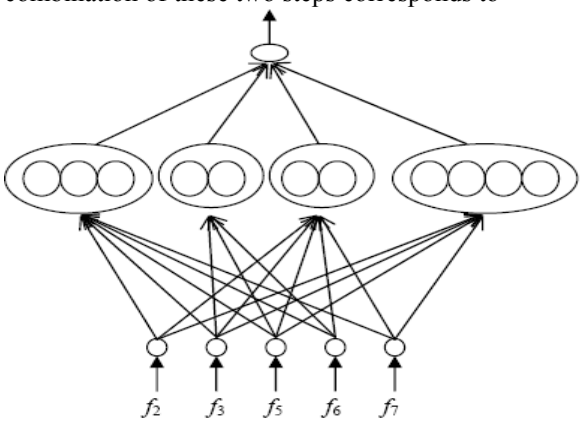

Figure 5. Neural Network after feature fusion

generalizing the effects of features, as a whole, and providing control parameters for sentence ranking. The Neural Network [27] after feature fusion is shown in figure5.

H. Automatic text summarization based on fuzzy logic

This method considers each characteristic of a text such as sentence length, similarity to little, similarity to key word and etc. as the input of fuzzy system[2][22]. Then, it enters all the rules needed for summarization, in the knowledge base of system. After ward, a value from zero to one is obtained for each sentence in the output based on sentence characteristics and the available rules in the knowledge base. The obtained value in the output determines the degree of the importance of the sentence in the final summary. The input membership function for each feature is divided into three membership functions which are composed of insignificant values (low L), very low (VL), medium (M), significant values (High h) and very high (VH). The important sentences are extracted using IF-THEN rules according to the feature criteria.

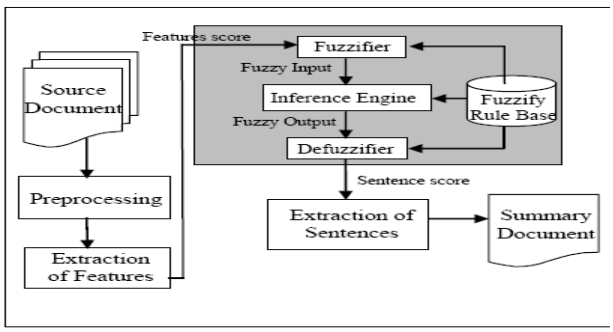

Figure 6. Text summarization based on fuzzy logic system architecture

Text summarization based on fuzzy logic system architecture [28] is shown in figure6. Fuzzy logic system design usually implicates selecting fuzzy rules and

membership function. The selection of fuzzy rules and membership functions directly affect the performance of the fuzzy logic system. The fuzzy logic system consists of four components: fuzzifier, inference engine, defuzzifier, and the fuzzy knowledge base. In the fuzzifier, crisp inputs are translated into linguistic values using a membership function to be used to the input linguistic variables. After fuzzification, the inference engine refers to the rule base containing fuzzy IFTHEN rules to derive the linguistic values. In the last step, the output linguistic variables from the inference are converted to the final crisp values by the defuzzifier using membership function for representing the final sentence score.

I. Text summarization using regression for estimating feature weights

Mathematical regression [6] is a good model to estimate the text feature weights. In this model, a mathematical function can relate output to input. The feature parameters of many manually summarized English documents are used as independent input variables and corresponding dependent outputs are specified in training phase. A relation between inputs and outputs is established. Then testing data are introduced to the system model for evaluation of its efficiency. In matrix notation we can represent regression as follow:

$$\begin{bmatrix} Y_0 \\ Y_1 \\ . \\ Y_m \end{bmatrix} = \begin{bmatrix} X_{01} & X_{02} & .... & X_{010} \\ . & . & .... & . \\ . & . & .... & . \\ X_{m1} & X_{m2} & .... & X_{m10} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ . \\ w_m \end{bmatrix}$$

Where

$[Y]$ is output vector.

$[X]$ is the input matrix (feature parameters)

$[w]$ is linear statistical model of system (the weights $w_1, w_2 \ldots\ldots\ldots w_{10}$ in the equation)

m is total number of sentences in the training corpus

J. Multi-document extractive summarization

Multi document extractive summarization deals with extraction of summarized information from multiple texts written about the same topic. Resulting summary report allows individual users, so as professional information consumers, to quickly familiarize themselves with information contained in a large cluster of documents. Multi-document summarization creates information reports that are both concise and comprehensive. With different opinions being put together & outlined, every topic is described from multiple perspectives within a single document.

NeATS [15] is a multi-document summarization system that attempts to extract relevant or interesting portions from a set of documents about some topic and present them in coherent order. It is an extraction-based multi-document summarization system. Given an input of a collection of sets of newspaper articles, NeATS generates summaries in three stages: content selection, filtering, and presentation.

The goal of content selection is to identify important concepts mentioned in a document collection. In a key step for locating important sentences, NeATS computes the likelihood ratio to identify key concepts in unigrams, bigrams, and trigrams, using the on- topic document collection as the relevant set and the off-topic document collection as the irrelevant set. With the individual key concepts available, these concepts are clustered in order to identify major subtopics within the main topic. Clusters are formed through strict lexical connection. Each sentence in the document set is then ranked, using the key concept structures.

NeATS uses three different filters: sentence position, stigma words, and maximum marginal relevancy. Sentence position is a good content filter, that only retains the leading 10 sentences. Some sentences start with stigma words like:

- Conjunctions (e.g., but, although, however)
- The verb *say* and its derivatives
- Quotation marks
- Pronouns such as he, she, and they

usually cause discontinuity in summaries. The scores of these sentences are reduced to avoid including them in short summaries. Redundancy issue is addressed in maximum marginal relevancy filter. A sentence is added to the summary if and only if its content has less than *X* percent overlap with the summary. The overlap ratio is computed using simple stemmed word overlap and the threshold *X* is set empirically.

Hub/Authority [39] framework is multi document summarization system which, firstly detect the sub-topics in multi-documents by sentence clustering and extract the feature words (or phrase) of different sub-topics. Secondly, all feature words and the cue phrases are used as the vertex of Hub and all sentences are regarded as the vertex of Authority. If the sentence contains the words in Hub, there is an edge between the Hub word and the Authority sentence. The initial weight of each vertex considers both the content and the cues such as cue phrase and first sentence. Through the mutual reinforcement mechanism of the Hub-Authority algorithm, we can rank the importance of the sentences within the multi-documents. The assumption behind this cue-based Hub/Authority approach is that a good Hub word (or phrase) is the content that points to many good authorities sentences and a good authority sentence is a vertex that is pointed to by many good hub words. Thirdly, It has used the Markov Model to order the sub-topics that the final summarization should contain and output the text summarization according to the sentence

ranking score of all sentences within one sub-topic as user' requirement.

Generic relation extraction (GRE) [40] is a novel multi document text summarization approach, which aims to build systems for relation identification and characterization that can be transferred across domains and tasks without modification of model parameters.

K. Query based extractive text summarization

In query based text summarization [42] system, the sentences in a given document are scored based on the frequency counts of terms (words or phrases). The sentences containing the query phrases are given higher scores than the ones containing single query words. Then, the sentences with highest scores are incorporated into the output summary together with their structural context. Portions of text may be extracted from different sections or subsections. The resulting summary is the union of such extracts. The number of extracted sentences and the extent to which their context is displayed depends on the summary frame size which is fixed to the size of the screen that can be seen without scrolling. In the sentence extraction algorithm, whenever a sentence is selected for the inclusion in the summary, some of the headings in that context are also selected.
The query based sentence extraction algorithm is as follows:

Algorithm:
1: Rank all the sentences according to their score.
2: Add the main title of the document to the summary.
3: Add the first level-1 heading to the summary.
4: While (summary size limit not exceeded)
5: Add the next highest scored sentence.
6: Add the structural context of the sentence:
(if any and not already included in the summary)
7: Add the highest level heading above the
extracted text (call this heading h).
8: Add the heading before h in the same level.
9: Add the heading after h in the same level.
10: Repeat steps 7, 8 and 9 for the next highest level
headings.
11: End while

An another query-specific summarization [43] method views a document as a set of interconnected text fragments (passages) and focuses on keyword queries, since keyword search is the most popular information discovery method on documents, because of its power and ease of use. Firstly, at the preprocessing stage, it adds structure to every document, which can then be viewed as a labeled, weighted graph, called the document graph. Then, at query time, given a set of keywords, it performs keyword proximity search on the document graphs to discover how the keywords are associated in the document graphs. For each document its summary is the minimum spanning tree on the corresponding document graph that contains all the keywords.

In query-specific opinion summarization system [44] (QOS), When input an opinion question, the system returns a summary with relevance to the opinion and target described by the question. The system has several modules to be able to do this: a question analysis and query reformulation module, a latent semantic indexing based sentence scoring module, a sentence polarity detection module, and a redundancy removal module.

Bayesian summarization [45] (BAYESUM) is a model for sentence extraction in query-focused summarization. BAYESUM leverages the common case in which multiple documents are relevant to a single query. Using these documents as reinforcement for query terms, BAYESUM is not afflicted by the paucity of information in short queries. For a collection of D documents and Q queries, assume a D × Q binary matrix r, where $r_{dq} = 1$ if an only if document d is relevant to query q. In multi document summarization, r_{dq} will be 1 exactly when d is in the document set corresponding to query q.

L. Multilingual Extractive Text summarization

Multilingual text summarization is to summarize the source text in different language to the target language final summary. SimFinderML [24] identifies similar pieces of text by computing similarity over multiple features. There are two types of features, composite features, and unary features. All features are computed over primitives, syntactic, linguistic, or knowledge-based information units extracted from the sentences. Both composite and unary features are constructed over the primitives. The primitives used and features computed can be set at run-time, allowing for easy experimentation with different settings, and making it easy to add new features and primitives. Support for new languages is added to the system by developing modules conforming to interfaces for text pre-processing and primitive extraction for the language, and using existing dictionary-based translation methods, or adding other language-specific translation methods.

MINDS [25] integrates multi-lingual summarization and multi document summarization capabilities using a multiengine, core summarization system and provides fast, interactive document access through hypertext summaries. Core summarization problem of MINDS is taking a single text and producing a shorter text in the same language that contains all the main points in the input text. It is using a robust, graded approach for building the core engine by incorporating statistical, syntactic and documents structure analyses among other techniques. This approach is less expensive and more robust than a summarization technique based entirely on a single method. The core engine is being designed in such a way that as additional resources, such as lexical and other knowledge bases or text processing and MT engines, become available from other ongoing research efforts they can be incorporated into the overall multi-engine MINDS system. Ideally the core engine itself will remain language independent. A prototype core engine has been built for English, Spanish, Russian, and Japanese documents.

MEAD [26] is the multi-lingual summarization and evaluation method. MEAD's architecture consists of four stages. First, documents in a cluster are converted to MEAD's internal (XML-based) format. Second, given a configuration file or command-line options, a number of features are extracted for each sentence of the cluster. Third, these features are combined into a composite score for each sentence. Fourth, these scores can be further refined after considering possible cross-sentence dependencies (e.g., repeated sentences, chronological ordering, source preferences, etc.) In addition to a number of command-line utilities, MEAD provides a Perl API which lets external programs access its internal libraries.

V. Conclusions

This survey paper is concentrating on extractive summarization methods. An extractive summary is selection of important sentences from the original text. The importance of sentences is decided based on statistical and linguistic features of sentences.

Many variations of the extractive approach [41] have been tried in the last ten years. However, it is hard to say how much greater interpretive sophistication, at sentence or text level, contributes to performance. Without the use of NLP, the generated summary may suffer from lack of cohesion and semantics. If texts containing multiple topics, the generated summary might not be balanced. Deciding proper weights of individual features is very important as quality of final summary is depending on it. We should devote more time in deciding feature weights.

The biggest challenge for text summarization is to summarize content from a number of textual and semi structured sources, including databases and web pages, in the right way (language, format, size, time) for a specific user. The text summarization software should produce the effective summary in less time and with least redundancy. Summaries can be evaluated using intrinsic or extrinsic measures. While intrinsic methods attempt to measure summary quality using human evaluation and extrinsic methods measure the same through a task based performance measure [35] such the information retrieval-oriented task.

VI. References

[1] Karel Jezek and Josef Steinberger, "Automatic Text summarization", Vaclav Snasel (Ed.): Znalosti 2008, pp.1-12, ISBN 978-80-227-2827-0, FIIT STU Brarislava, Ustav Informatiky a softveroveho inzinierstva, 2008.

[2] Farshad Kyoomarsi, Hamid Khosravi, Esfandiar Eslami and Pooya Khosravyan Dehkordy, "Optimizing Text Summarization Based on Fuzzy Logic", In proceedings of Seventh IEEE/ACIS International Conference on Computer and Information Science, IEEE, University of Shahid Bahonar Kerman, UK, 347-352, 2008.

[3] Joel larocca Neto, Alex A. Freitas and Celso A.A.Kaestner, "Automatic Text Summarization using a Machine Learning Approach", Book: Advances in Artificial Intelligence: Lecture Notes in computer science, Springer Berlin / Heidelberg, Vol 2507/2002, 205-215, 2002.

[4] Weiguo Fan, Linda Wallace, Stephanie Rich, and Zhongju Zhang, "Tapping into the Power of Text Mining", Journal of ACM, Blacksburg, 2005.

[5] Fang Chen, Kesong Han and Guilin Chen, "An Approach to sentence selection based text summarization", Proceedings of IEEE TENCON02, 489-493, 2002.

[6] Mohamed Abdel Fattah and Fuji Ren, "Automatic Text Summarization", Proceedings of World Academy of Science, Engineering and Technology, Vol 27,ISSN 1307-6884, 192-195, Feb 2008.

[7] H. P. Luhn**, "**The Automatic Creation of Literature Abstracts", Presented at IRE National Convention, New York, 159-165, 1958.

[8] H. P. Edmundson.," New methods in automatic extracting", Journal of the ACM, 16(2):264-285, April 1969.

[9] J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer", In Proceedings of the 18th ACM-SIGIR Conference, pages 68-73, 1995

[10] Ronald Brandow, Karl Mitze, and Lisa F. Rau. "Automatic condensation of electronic publications by sentence selection. Information Processing and Management", 31(5):675-685,1995.

[11] E. Mittendorf and P. Schauble, " Document and passage retrieval based on hidden markov models", In Proceedings of the 17th ACM-SIGIR Conference, pages 318-327,1994.

[12] A. Bookstein, S. T. Klein, and T. Raita, "Detecting content-bearing words by serial clustering", In Proceedings of the 18th ACM-SIGIR Conference, pages 319-327, 1995.

[13] Madhavi K. Ganapathiraju, "Overview of summarization methods", 11-742: Self-paced lab in Information Retrieval, November 26, 2002.

[14] Klaus Zechner, "A Literature Survey on Information Extraction and Text Summarization", Computational Linguistics Program, Carnegie Mellon University, April 14, 1997.

[15] Chin-Yew Lin and Eduard Hovy," From Single to Multi-document Summarization: A Prototype System and its Evaluation", Proceedings of the ACL conference, pp. 457–464. Philadelphia, PA. 2002.

[16] Rene Arnulfo Garcia-Herandez and Yulia Ledeneva, "Word Sequence Models for Single Text Summarization", IEEE,44-48, 2009.

[17] Yongzheng, Nur and Evangelos, "Narrative Text Classification for Automatic Key Phrase Extraction in Web Document Corpora", WIDM'5, 51-57, Bremen Germany,2005.

[18] Canasai Kruengkari and Chuleerat Jaruskulchai, "Generic Text Summarization Using Local and Global Properties of Sentences", Proceedings of the IEEE/WIC international Conference on Web Intelligence (WI'03) , 2003.

[19] Meng Wang, Xiaorong Wang and Chao Xu, "An Approach to Concept Oriented Text Summarization", in Proceedings of ISCIT'05, IEEE international conference, China,1290-1293, 2005.

[20] Azadeh Zamanifar, Behrouz minaei-Bidgoli and Mohsen Sharifi," A New Hybrid Farsi Text Summarization Technique Based on Term Co-Occurrence and Conceptual Property of Text ", In Proceedings of Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, IEEE, 635-639, Iran, 2008.

[21] Khosrow Kaikhah, "Automatic Text Summarization with Neural Networks", in Proceedings of second international Conference on intelligent systems, IEEE, 40-44, Texas, USA, June 2004.
[22] Ladda Suanmali, Mohammed Salem, Binwahlan and Naomie Salim, "Sentence Features Fusion for Text summarization using Fuzzy Logic, IEEE, 142-145, 2009
[23] David B. Bracewell, Fuji REN and Shingo Kuriowa, "Multilingual Single Document Keyword Extraction for Information Retrieval", Proceedings of NLP-KE'05, IEEE, Tokushima, 2005.
[24] David Kirk Evans, "Identifying Similarity in Text: Multi Lingual Analysis for Summarization", PhD thesis, Graduate School of Arts and Sciences, Columbia University, 2005.
[25] Cowie, J., Mahesh, K., Nirenburg, S., and Zajaz, R., "MINDS-Multilingual INteractive document summarization", In Working Notes of the AAAI Spring Symposium on Intelligent Text Summarization (pp. 131–132). Menlo Park, CA: AAAI, 1998.
[26] Dragomir Radev, Timothy Allison, Sasha Blair-Goldensohn, John Blitzer, Arda C, elebi, Stanko Dimitrov, Elliott Drabek, Ali Hakim, Wai Lam, Danyu Liu, Jahna Otterbacher, Hong Qi, Horacio Saggion, Simone Teufel, Michael Topper, Adam Winkel, and Zhang Zhu., "MEAD - a platform for multi document multilingual text summarization", In Proceedings of LREC 2004, Lisbon, Portugal, May 2004.
[27] Khosrow Kaikhah "Text Summarization using Neural Networks", Department of Faculty Publications-Computer Science, Texas State University, eCommons,2004.
[28] Ladda Suanmali, Naomie Salim and Mohammed Salem Binwahlan, "Fuzzy Logic Based Method for Improving Text Summarization", (IJCSIS) International Journal of Computer Science and Information Security, Vol. 2, No. 1, 2009.
[29] Rasim M. Alguliev and Ramiz M. Aliguliyev, "Effective Summarization Method of Text Documents", in Proceedings of IEEE/WIC/ACM international conference on Web Intelligence (WI'05), 1-8, 2005.
[30] Berry Michael W., "Automatic Discovery of Similar Words", in "Survey of Text Mining: Clustering, Classification and Retrieval", Springer Verlag, New York, LLC, 24-43, 2004.
[31] Vishal Gupta, G.Sl Lehal, "A Survey of Text Mining Techniques and Applications", Journal of Emerging Technologies in Web Intelligence, VOL. 1, NO. 1, 60-76, AUGUST 2009.
[32] G Erkan and Dragomir R. Radev, "LexRank: Graph-based Centrality as Salience in Text Summarization", Journal of Artificial Intelligence Research, Re-search, Vol. 22, pp. 457-479 2004.
[33] Udo Hahn and Martin Romacker, "The SYNDIKATE text Knowledge base generator", Proceedings of the first International conference on Human language technology research, Association for Computational Linguistics , ACM, Morristown, NJ, USA , 2001.
[34] Ani Nenkova and ,Rebecca Passonneau, "Evaluating content selction in summarization: The Pyramid method", in HLT-NAACL, 145-152, 2004.
[35] Kathleen Mackeown, Ani Nenkova, David Elson, Rebecca Passonneau, and Julia Hirschberg "A task based evaluation of multidocument system",in SIGIR'05, ACM, 2005.
[36] Chin-yew Lin, "A package for automatic evaluation of summaries",in Proc. ACL workshop on text summarization branches out,2004.
[37] Eduard Hovy, Chin-Yew Lin, Liang Zhou, and Junichi Fukumoto, "Automated Summarization Evaluation with Basic Elements", In Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC), 2006.
[38] Samuel W K. Chan, Tom B. Y. Lai, W.J. Gao and Benjamin K. T'sou, "Mining discourse structures for chinese textual summarization", NAACL-ANLP workshop on Automatic Summarization, ACM, Seattle, Washington, 11-20, 2000.
[39] Junlin Zhanq, Le Sun and Quan Zhou, "A Cue-based Hub-Authority Approach for Multi-Document Text Summarization", in Proceeding of NLP-KE'05, IEEE,642-645, 2005
[40] Ben Hachey, "Multi-document summarization using generic relation extraction", Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing: Volume 1, 420-429, 2009.
[41] K. S. Jones, "Automatic summarizing: the state of the art," Information Processing and Management, Elsevier, Vol. 43, No. 6, pp. 1449–1481, 2007.
[42] F. Canan Pembe and Tunga Güngör, "Automated Query-biased and Structure-preserving Text Summarization on Web Documents", Proceedings of the International Symposium on Innovations in Intelligent Systems and Applications, İstanbul, June 2007.
[43] Ramakrishna Varadarajan and Vagelis Hristidis, "Structure-Based Query-Specific Document Summarization", in proceedings of CIKM'05, ACM, Bremen, Germany, 2005.
[44] Feng Jin, Minlie Huang and Xiaoyan Zhu, " A Query-specific Opinion Summarization System", in proceedings ofICCI '09, 8th IEEE international conference on cognitive informatics, Kowloon, Hong Kong, 428-433, 2009.
[45] Hal Daum´e III and Daniel Marcu, "Bayesian Query-Focused Summarization", Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, 305–312, Sydney, July 2006.
[46] Jimmy Lin., "Summarization.", Encyclopedia of Database Systems. Heidelberg, Germany: Springer-Verlag, 2009.
[47] Jackie CK Cheung, "Comparing Abstractive and Extractive Summarization of Evaluative Text: Controversiality and Content Selection", B. Sc. (Hons.) Thesis in the Department of Computer Science of the Faculty of Science, University of British Columbia, 2008.

VISHAL GUPTA

Vishal Gupta is Assistant Professor in Computer Science & Engineering at University Institute of Engineering & Technology, Panjab university Chandigarh. He has done MTech. in computer science & engineering from Punjabi University Patiala in 2005. He is among University toppers. He has done BTech. in Computer Science & Engineering from Govt. Engineering College Ferozepur in 2003. He is also pursuing his PhD in Computer Science &

Engineering from University College of Engineering, Punjabi University Patiala, under the supervision of Dr. Gurpreet Singh Lehal. He is state merit holder in 10^{th} and 12^{th} classes of Punjab School education board. He is devoting his research work in field of Natural Language Processing. He has developed a number of research projects in field of NLP including synonyms detection, automatic question answering and text summarization etc. One of his research paper on Punjabi language text processing was awarded as best research paper by Dr. V. Raja Raman at an International Conference.

DR. GURPREET SINGH LEHAL

Professor Gurpreet Singh Lehal received undergraduate degree in Mathematics in 1988 from Panjab University, Chandigarh, India, and Post Graduate degree in Computer Science in 1995 from Thapar Institute of Engineering & Technology, Patiala, India and Ph. D. degree in Computer Science from Punjabi University, Patiala, in 2002. He joined Thapar Corporate R&D Centre, Patiala, India, in 1988 and later in 1995 he joined Department of Computer Science at Punjabi University, Patiala. He is actively involved both in teaching and research. His current areas of research are- Natural Language Processing and Optical Character recognition. He has published more than 25 research papers in various international and national journals and refereed conferences. He has been actively involved in technical development of Punjabi and has to his credit the first Gurmukhi OCR, Punjabi word processor with spell checker and various transliteration software. He was the chief coordinator of the project "Resource Centre for Indian Language Technology Solutions- Punjabi", funded by the Ministry of Information Technology as well as the coordinator of the Special Assistance Programme (SAP-DRS) of the University Grants Commission (UGC), India. He was also awarded a research project by the International Development Research Centre (IDRC) Canada for Shahmukhi to Gurmukhi Transliteration Solution for Networking.