**Problem Statement**

**Need For the Project**

With the coming of the information revolution, electronic documents are becoming a principle media of business and academic information. Thousands and thousands of electronic documents are produced and made available on the internet each day. In order to fully utilizing these on-line documents effectively, it is crucial to be able to extract the gist of these documents. Having a Text Summarization system would thus be immensely useful in serving this need.

Text Summarization does the job of making a summary that is produced from one or more documents, that conveys important information in the original text, and it is a shorter form of article. Using this we can help news article writer to create a headline of article, help make a shorter summary of an article which readers may have trouble reading all of them. It can provide as an outline for students to get a gist of an article.

**Survey**

Most early work on text summarization was focused on technical documents and

Early studies on summarization aimed at summarizing from pre-given documents without any other requirements, which is usually known as generic summarization[1]. Luhn [2] proposed that the frequency of a particular word in an article provides a useful measure of its significance. A number of key ideas, such as stemming and stop word filtering, were put forward in this paper that have now been understood as universal preprocessing steps to text analysis. Text Summarization can be done for one document, known as single-document summarization [3], or for multiple documents, known as multi document summarization [4]. On basis of the writing style of the final summary generated, text summarization techniques can be divided into extractive methodology and abstractive methodology [5].

**Work Done**

The current progress has been analyzing various feature metrics to be derived for feeding into a Machine Learning model. Collection and preprocessing of sentences to a proper format so that it is easy to analyze and can be computed. Along with the data manipulation, we have built a web application which will enable a user to input an article either by manually typing or by selecting and uploading a word file which will be read and the summary of the article will be displayed to the user.

**Future Work**

The preprocessed work is used as input features to various Machine Learning algorithm and the model giving maximum accuracy can be chosen. Various new features will have to be incorporated by segregating the type of article entered by a user to give more accuracy. Various formats for file uploading will be supported like PDFs and text file which will open space for variety of documents that can be summarized.

**Design Phase**

Text Summarization process is broken down into three phases:

1. **Pre-processing the data**

Preprocessing is crucial when it comes to processing text. Ambiguities can be caused by various verb forms of a single word, different accepted spellings of a certain word, plural and singular terms of the same things. Moreover, words like a, an, the, is, of etc. are known as stop words. These are certain high frequency words that do not carry any information and don’t serve any purpose towards our goal of summarization.

1. **Feature Extraction**

Once the complexity has been reduced and ambiguities have been removed, the document is structured into a sentence-feature matrix. A feature vector is extracted for each sentence. These feature vectors make up the matrix.

1. **Summary Generation**

The feature vector values are summed to generate a score against each sentence. The sentences are then sorted according to decreasing score value. The most relevant sentence is the \_rst sentence in this sorted list and is chosen as part of the subset of sentences which will form the summary. Then the next sentence we select is the sentence having highest Jaccard similarity with the first sentence, selected strictly from the top half of the sorted list. This process is recursively and incrementally repeated to select more sentences until a user-specified summary limit is reached. The sentences are then re-arranged in the order of appearance in the original text. This produces a coherent summary rather than a set of haywire sentences.

**CHAPTER - 7**

**References**

1. Berger, A., Mittal, V.: Query relevant summarization using FAQs. In: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, pp.294301. Association for Computational Linguistics Stroudsburg, PA, USA (2000).
2. Luhn, H. P.: The Automatic Creation of Literature Abstracts. IBM Journal of Research and Development, vol. 2, issue 2, 159165 (1958).
3. Wan, X., Xiao, J.: Single Document Keyphrase Extraction using Neighborhood Knowledge. In: Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (2008).
4. Shen, D., Sun, J.T., Li, H., Yang, Q., Chen, Z.: Document Summarization using Conditional Random Fields. In: Proceedings of the 20th international joint conference on Artificial Intelligence, pp. 2862-2867. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA (2007).
5. Wong, K.F., Wu, M.J., Li, W.J.: Extractive Summarization Using Supervised and Semi-supervised Learning. In: Proceedings of the 22nd International Conference on Computational Linguistics Volume 1, pp. 985-992. Association for Computational Linguistics Stroudsburg, PA, USA (2008).