Abstract

The need for text summarization is crucial as we enter the era of information overload. In this paper we present an automatic summarization system, which generates a summary for a given input document. Our system is based on identification and extraction of important sentences in the input document. We listed a set of features that we collect as part of summary generation process. These features were stored using vector representation model. We defined a ranking function which ranks each sentence as a linear combination of the sentence features. We also discussed about discourse coherence in summaries and techniques to achieve coherent and readable summaries. The experiments showed that the summary generated is coherent the selected features are really helpful in extracting the important information in the document. 1 Introduction A huge amount of on-line information is available on the web, and is still growing. While search engines were developed to deal with this huge volume of documents, even they output a large number of documents for a given user's query. Under these circumstances it became very difficult for the user to find the document he actually needs, because most of the naive users are reluctant to make the cumbersome effort of going through each of the documents. Therefore systems that can automatically summarize one or more documents are becoming increasingly desirable. A summary can be loosely defined as a short version of text that is produced from one or more texts. Automatic summarization is to use automatic mechanism to produce a finer version for a given documents. SparkJones (1999) discussed several ways to classify summaries. The following three factors are considered to be important for text summarization. • Input factors : text length, genre, number of documents • Purpose factors: audience, purpose of summarization. • Output factors: running text or headed text etc. Summaries can be classified into different types based on dimensions, genre, and context. • Dimensions Single vs. Multi-document summarization • Genre Headlines, outlines, minutes, chronologies, etc. • Context Generic, Query specific summaries As pointed out in Mani and Maybury [1999] summaries can be classified in to extracts (most relevant sentences

are selected from the text), and abstracts (text is analyzed, a conceptual representation is provided which in turn is used to generate sentences that form summary). Conventional text summarization systems produce summaries by using sentences or paragraphs as basic unit, giving them degree of importance (Edmundson [1969]), sorting them based on the importance, and gathering the important sentences. In this paper we have presented an extract type summary generation system. 2 Background Most of the summarization work done till date is based on extraction of sentences from the original document. The sentence extraction techniques compute score for each sentence based on features such as position of sentence in the document[Baxendale 1958; Edmundson 1969], word or phrase frequency[Luhn 1958], key phrases (terms which indicate the importance of the sentence towards summary e.g. “this article talks about”)[Edmundson 1969]. There were some attempts to use machine learning (to identify important features), use natural language processing (to identify key passages or to use relationship between words rather than bag of words). The application of machine learning to summarization was pioneered by Kupiec, Pedersen, and Chen [1995], who developed a summarizer for scientific articles using a Bayesian classifier. For the generation of a coherent and readable summary, one has to do significant amount of text analysis to generating good feature vector, handling

discourse connectors , and refining the sentences. This system is an attempt in that direction. 3 System Description The architecture of our summarization system is shown in Fig 1. The system has both text analysis component and summary generation component. The text analysis component is based on syntactic analysis, followed by a component which identifies the features associated with each sentence. Text normalization is applied before syntactic analysis of the text which include extracting the text from the document (format conversion, if needed), removing floating objects like figures, tables, identification of titles and subtitles, and dividing the text into sentences. After text normalization the normalized text is a passed through a feature extraction module. Feature extraction include extracting features associated with the sentence (such as sentence number, number of words in that sentence and so on) and the features associated with words (such as the named entities, the term frequency and so on). The summary generation component calculates the score for each sentence based on the features that were identified

by the feature extraction module. Sentence refinement is done on the sentences with high score, and the resulting sentences are selected for the summary in the same order as they were found in the input text document.