Chapter-1

Introduction

1.1 Introduction to Text Summarization

The way we share, store, write and publish text data has been revolutionized with the advances in networks, Internet, PC hardware and software tools, which makes it easier to create content. When you have such a vast material at your hands it gets harder to search and find data relevant to you. While search engines can be used for finding data, they don’t analyze the semantic structure of the text or understand the text. Matches for queries depend mostly on word repetition. Thus, you may end up trying to scan large number of documents. Summaries aid users to evaluate the relevance of a document without reading the full text. Since a summary is not cluttered with detail, a user can quickly recognize its relevance.

As we can see, many Digital Libraries contain huge amount of information and a simple search would yield thousands of pages of content, which is to be submitted to the user in a brief manner. In the same manner, if at all, a user wants to know the content in some ‘Research Papers’ offered in some website, it would be extremely tiresome to go through all the pages and then find out that the information is irrelevant to you, where a brief ‘summary’ of the entire thesis can be presented, which helps a lot. The authors predict that
the summarization tools will play an important role in conquering the vast information universes ahead.

Eduard Hovy [17] formally defines summary as:

‘a summary is a text that is produced from one or more texts, that contains a significant portion of the information in the original text(s), and that is no longer than half of the original text(s).’

This simple definition captures three important aspects that characterize research on automatic summarization:

- Summaries may be produced from a single document or multiple documents,
- Summaries should preserve important information,
- Summaries should be short.

The important topics in the text are abstracts, which are special forms of summaries that are generated. A summary formed of sentences extracted from the original text is called an extract. In the case of keyphrases, extraction of the most important and representative phrases is called keyphrase extraction which constrains the output to phrases, that appear in the document. If the target key phrases contain phrases that do not appear in the original text, it is called keyphrase generation.

Content to be summarized can be any data representable in text. Long documents such as journals, novels and books or short documents
such as emails, news articles and dialogue scripts are some examples of attacked text genres. Each genre has different structures and its own difficulties. Summarizing longer documents with more topics and weaker co-relations is a harder problem, because it is difficult to determine the main topic in the document. In a short document, repetition of terms is lower and interpreting the document requires more prior knowledge.

Summaries could be displayed in search results as an informative tool for the user. For example Google search engine provides summaries in search results. Digital libraries and journals could make use of summaries, so that users of the library or readers of the journal could benefit from summaries and find relevant text easily. News portals could provide precise summaries about news merged from multiple source articles. Web browsing with web site summaries could change our browsing habits and enable us to filter irrelevant web pages.

Search engines could index summaries instead of the whole document, lowering the resources needed by indexing algorithms. A summary capable of highlighting the most important aspects of the text could improve the performance of search engines, in terms of the relevance of the results.

Unfortunately, most of the data available today does not have summaries. Even for a human, summarizing a text is an exhaustive and difficult work. For this reason, automating the task of creating
indicative and informative summaries has been issued by many researchers.

*Automated text summarization* is the process of automatically constructing summaries for a text. ‘*Automatic text summarization aims at providing a condensed representation of the content according to the information that the user wants to get.*’

Automatic Summarization also plays an important role in Digital Libraries. As there are many books related to a particular subject, the user wants to read the most apt one’s only. Summarization provides the users the resources to read the gist of the book/document (Summary /Abstract). It also helps him to get rid of going through the redundant information, by way of duplicate detection and deletion. Summarization also helps in quick searching. It is optimal to search for the key words in the Summary/Abstract rather than searching in the whole text because the Summary is supposed to contain all the important key words.

Systems summarizing a single document are called *single document summarization* systems, while systems summarizing a set of documents to form a single summary are called *multi document summarization* systems. Single document summarization is a difficult task by itself, but multi document summarization has additional difficulties. *Query relevant summarization* systems provide a summary for document(s) based on a query or a question. Query relevant summarization is very
similar to question answering. The generated summary is shaped by the user's interest.

Cohesion and coherence are two natural phenomena, seen in sensible texts. Coherence is the semantic structure of the text that gives the feeling that the text is interpretable. The coherence structure is hard to model. Modeling the coherence structure requires prior knowledge and requires some level of understanding. Marcu [62] presents a good summarization system which takes advantage of coherence. Marcu uses discourse structure and more specifically rhetorical parsing to model coherence. His model depends on cue phrases called discourse markers. Research on coherence based summarization systems, is challenged by the difficulties in modeling coherence.

Cohesion is defined as sticking together. In text, text units stick to each other with relations. Relations between word meanings in a text form lexical cohesion, which is a type of cohesion. Lexical chains are computational models for lexical cohesion.

Evans [46] addresses the task of summarizing documents written in multiple languages, this had already been sketched by Hovy and Lin [49]. Multilingual summarization is still at an early stage, but this framework looks quite useful for newswire applications that need to combine information from foreign news agencies.

Evans [46] considered the scenario where there is a preferred language in which the summary is to be written, and multiple documents in the
preferred and in foreign languages are available. In their experiments, the preferred language was English and the documents are news articles in English and Arabic. The rationale is to summarize the English articles without discarding the information contained in the Arabic documents. The IBM’s statistical machine translation system is first applied to translate the Arabic documents to English. Then a search is made, for each translated text unit, to see whether there is a similar sentence or not in the English documents. If the sentence is found relevant enough to be included in the summary, the similar English sentence is included instead of the Arabic-to-English translation. This way, the final summary is more likely to be grammatical, since machine translation is known to be far from perfect. On the other hand, the result is also expected to have higher coverage than using just the English documents, since the information contained in the Arabic documents can help to decide about the relevance of each sentence. In order to measure similarity between sentences, a tool named SimFinder was employed, this is a tool for clustering text based on similarity over a variety of lexical and syntactic features using a log-linear regression model.

A crucial issue that will certainly drive future research on summarization is evaluation. During the last fifteen years, many system evaluation competitions like TREC, DUC, and MUC have created sets of training material and have established baselines for
performance levels. Evaluating a summary is a difficult task because there doesn’t exist an ideal summary for a given document or set of documents. From literature survey, it has been found that agreement between human summarizers is quite low, both for evaluating and generating summaries. More than the form of the summary, it is difficult to evaluate the summary content.

Another important problem in summary evaluation is the widespread use of disparate metrics. The absence of a standard human or automatic evaluation metric makes it very hard to compare different systems and establish a baseline. This problem is not present in other NLP problems, like parsing. Besides this, manual evaluation is too expensive, as large scale manual evaluation of summaries, as in the DUC conferences would require over 3000 hours of human efforts. Hence, an evaluation metric having high correlation with human scores would obviate the process of manual evaluation [69].

Usually, the flow of information in a given document is not uniform, which means that some parts are more important than others. The major challenge in summarization lies in distinguishing the more informative parts of a document from the less ones.

Traditionally, Information Retrieval systems rank and present documents based on measuring relevance to the user query. But not all the information retrieved are really useful to the user, and it always takes a lot of time to read and select before the user get what he wants.
Automatic text summarization became an exciting topic in Information Retrieval since it presents the user with summaries of the matching documents which can help the user identify which documents are most relevant to the user’s needs in a very short period of time.

Summarization methods can be roughly grouped into three categories. They are statistical approach, knowledge-based approach and general linguistic approach. Combining them usually can produce much better results for users.

1.1.1 Statistical Approach

The statistical approach summarizes without understanding, and relies on the statistical distribution of certain features [16]. Recent representative works include the classification-based method, the IR-based method and the position-based method among others.

For the classification-based method, it is to classify sentences as either summary worthy or not, depending on the training data. A classifier can be used to perform this task. A training procedure, which is an algorithm, is used to calculate the parameters of the classifier.

For the IR based methods, usually make full use of Information Retrieval. But normally it is hard to give a satisfactory result, because the output summary is not very coherent.

For the position-based method, we should notice that, important sentences are located at positions that are genre-dependent.
1.1.2 Knowledge-Based Approach

The knowledge-based approach stands in direct contrast to the statistical approach. This approach interprets the text using extensive domain knowledge as well as natural language techniques and then summarizes it. This approach is usually used in a specific domain. Compared to the statistical approach, it is more expensive but produces better quality summaries for a specific domain.

1.1.3 General linguistic Approach

The general linguistic approach, works in a general domain and doesn’t rely on domain knowledge, but relies on natural language processing techniques. So, what it differs from the knowledge-based approach is in the aspect of dependence on domain knowledge. From [16], we can see that, nowadays the understanding of text is still shallow. It usually relies on lexical information or discourse markers. To understand the document at a semantic level remains a challenge at the current stage.

1.2 Ontology Based Automated Text Summarization

Ontology is defined as an explicit specification of a conceptualization. Conceptualization consists of the existents and their characteristics. Ontology is interpreted as the formal representation of the conceptualization [25].

As an analogy, one can describe the ontology of a domain as the relational schema of a database. Relational schema represents both the
entities (concepts) and the dependency relations between the entities whereas the ontology consists of concepts and semantic relations between the concepts. By considering ontology attributes we are able to improve the semantic representation of a sentence’s information content.

Each ontology node is populated by a bag-of-words constructed from a web search. Sentences are represented by sub trees in the ontology space, which allows to apply similarity measures in the ontology-space and to compute relations between sentences based on graph properties of the sub trees. Furthermore, node confidence weights computed by the classifier enable us to identify the main topics of a document. The classifier maps a sentence to the taxonomy by choosing the sub trees which best represent the sentence. If the maximum similarity of a child is lower than the current node’s similarity to the sentence, or if a leaf node has been reached, the algorithm stops. A sentence is therefore not necessarily classified to a leaf node, but may be assigned to an internal node. For ontology-based summarization we compute a set of features for each sentence based on the output of the hierarchical classifier. If a sentence is mapped to multiple sub trees in the taxonomy, we include all nodes from every sub tree. If a sentence is classified as a leaf node of a certain depth, it is assumed to contain more specific information than a sentence that is classified to a higher-ranked internal node. We create a bag-of-words for each sentence by removing stop words and applying stemming.
1.3 Description of the Research Work

1.3.1 Motivation

With huge amount of information available on the World Wide Web, there is a pressing need to have Information Access systems that would help users, in providing the relevant information in a concise, pertinent format. People keep abreast of world affairs by listening to news bites. They take investment decisions based on the stock market updates. They go to movies largely on the basis of reviews. With summaries, they can make effective decisions in less time. Summaries save readers time by giving an overview or outline of the content.

We perceived many writers to be having a common inclination in describing their texts. Most of the writers were observed to be having a tendency to use longer sentences, whenever they are at the crux of explaining a critical theme, perhaps due to an inability to elucidate the complex feature in simpler sentences. It is also observed that the words that are considered more important are observed to have been repeating more frequently than those that are considered less important.

Most research on summary generation techniques still relies on extraction of important sentences from the original document to form a summary. Although many summarizing tools are available, it is becoming very difficult to generate meaningful and timely summaries, with the increasing volume of online information.
We also explored the problems facing Indian language information access and quantified the extent of the problem. Our experiments on text summarization tasks using state-of-the-art algorithms used for English like languages showed low accuracy when applied to Indian languages like Telugu.

### 1.3.2 Our Approach

A novel approach is proposed in this thesis for 'Automatic English Summarization' with the following postulates:

- The longest sentence in the text file will be an important sentence in the text.
- The frequently repeated words in the text are certainly the important words in the text.
- The association of the frequently repeated words with long sentences is a key factor.

With these three factors in mind, a well-defined procedure is followed to prepare a ‘summary’ of the given text. Any duplicate sentences generated, if at all, are removed by a Similarity-Search algorithm, and the resultant summary is then compared with the following, for the purpose of calculating efficiency:

- Summary generated by two jurors, who went through the whole text and prepared their own summary, extracting only the original sentences from the source text.
Summary generated over the net by the ‘Digital Library of India’s (DLI) Text Summarization Utility.

It is concluded that the procedure not only gave good summaries, but also is efficient in summarizing ‘technical’ content and the performance is satisfactory in ‘general’ content also.

The second approach is ‘Ontology Based Automatic Text Summarization’, in which we present an approach to sentence extraction that maps sentences to nodes of a hierarchical ontology. By considering ontology attributes we are able to improve the semantic representation of a sentence’s information content. Our experimental results show that the ontology-based extraction of sentences outperforms baseline classifiers, leading to higher accuracy of summary extracts.

As an analogy, one can describe the ontology of a domain as the relational schema of a database. Relational schema represents both the entities (concepts) and the dependency relations between the entities whereas the ontology consists of concepts and semantic relations between the concepts. Each ontology node is populated by a bag-of-words constructed from a web search. Sentences are represented by sub trees in the ontology space, which allows to apply similarity measures in the ontology-space and to compute relations between sentences based on graph properties of the sub trees. Furthermore, node confidence weights computed by the classifier enable us to
identify the main topics of a document. The classifier maps a sentence to the taxonomy by choosing the sub trees which best represent the sentence. If the maximum similarity of a child is lower than the current node’s similarity to the sentence, or if a leaf node has been reached, the algorithm stops. A sentence is therefore not necessarily classified to a leaf node, but may be assigned to an internal node. For ontology-based summarizer we compute a set of features for each sentence based on the output of the hierarchical classifier. If a sentence is mapped to multiple sub trees in the taxonomy, we include all nodes from every sub tree. If a sentence is classified as a leaf node of a certain depth, it is assumed to contain more specific information than a sentence that is classified to a higher-ranked internal node. We create a bag-of-words for each sentence by removing stop words and applying stemming.

In this thesis we also explored the problems facing Indian language information access and quantified the extent of the problem. Our experiments on text summarization tasks using state-of-the-art algorithms used for English like languages showed low accuracy when applied to Indian language like ‘Telugu’.

So we presented our approach towards ‘Single and Multi-document Telugu Automated Text Summarization’. Our approach attempts to generate a text summary from the article of Telugu news papers, while avoiding the repetition of identical or similar information and presenting the information in such a way that makes sense to the reader. The proposed algorithm work as follows:
➢ Summarize each document
  o The first step of the process of generating a multi-document summary is to create individual single-document summaries for all documents in the set.

➢ Group the summaries in clusters
  o The second step of the multi-document (MD) summarization process is the grouping of the individual summaries into clusters. Documents are ‘clustered’ on the basis of the contents of their summaries, where a cluster consists of summaries that describe a similar topic.
  o For those documents that seem to discuss a similar topic, representative segments are eventually chosen for the MD summary while the other ones are ‘hidden’, i.e., not shown but still accessible to the user.

➢ For each cluster select representative passage(s) that will contribute to the final summary
  o The third step of the multi-document summarization process involves selecting a member of a cluster as a representative summary for the cluster. When the user wants a topical summary, the topic description is used to pick the summary that has most similarity to the topic.
  o In the case of a generic summary, the representative summary chosen is one that best represents the cluster. In this case, the summary that has most occurrences of the common terms across documents in a cluster is chosen. Since clusters can be
overlapping it is possible that the same segment(s) is chosen to represent several clusters.

- Organize these passages in a logical way.
  - Finally, the last step of the multi-document summarization process involves organizing the selected passages in an order that makes sense to the reader. Currently, we organize the selected passages based upon topic similarity.

Finally it is concluded that the proposed procedure not only gave good summaries, but also with higher accuracy.

1.3.3 Scopes and Objectives

- The proposed systems works for the data corpus of English and Telugu documents.
- Ontology Based Automated Text Summarization is limited to a particular domain.

1.4 Organization of the thesis

This thesis is organized as follows:

Chapter 2 discusses the necessary background of Text Summarization and Similarity Search- Methods and Techniques.

Chapter 3 discusses the approach towards Automated English Text Summarization.
Chapter 4 discusses the Ontology Based Approach for Automated Text Summarization.

Chapter 5 discusses the approaches towards Automated Telugu Summarization.

Chapter 6 discusses the approaches towards Corpus Based Multi-document Telugu Text Summarization

Chapter 7 discusses Conclusion