Chapter 1: Introduction

Abstract

Recently, data is growing fast in every domain such as news, social media, banking, education. With the increasing amount of information, it has become diﬃcult to take out concise information. There is a need of automatic summarizer which will be capable to present human quality summaries and summarize the data especially textual data in original document without losing any incisive purposes.

This Automatic text summarization is a tool that provides summaries of a given document. This Tool can be use on news stories, blog articles and social media posts. Most Important Thing is this supports a wide variety of languages. This Summarization tool can be used to summarize one single document on a percentage. Summarization base on a keyword and generate a summary report can be considered as some extra options that user can select.

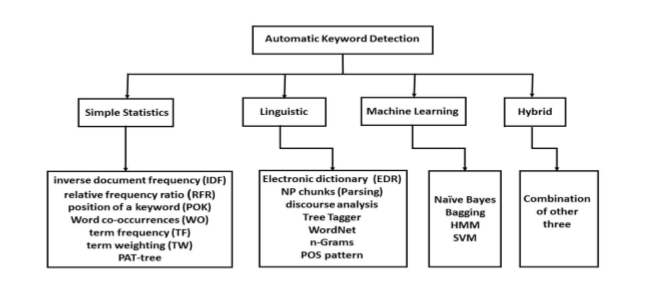
This will be ideal for any user as this produce short length text which comprises all the key information of the document. This is also appropriate for users who need to summarize the entire document based on one specific keyword. This will reduce using several applications such as word and google translator because this itself can generate a summary report and also can summarize documents with own native languages.

Text summarization is emerged as an important research area in recent past. In this regard, study about existing work on text summarization process is useful for build a new solution. It was also a necessity to effort to examine the needs of the solution addressed problems would enable a wider range of people to be aware of important information.

1.1: Project Background

The goal of text summarization Tool is to Publish the contents of a document in a concise form of meeting a user's needs. Online information is available for readers in the form of News information, journal articles, technical reports, biographical information it isn't possible to read everything and There are vast number of documents available in these digital media and extracting only relevant information from all these media is a dull job for the individuals in specify time. so some form of information condensation is needed that can extract only pertinent information from these data sources. To Realize this, from the documents texts mining is necessary The purpose of Text Mining is to process unstructured (textual) information, extract meaningful numeric indices from the text, and, thus, make the information contained in the text accessible to the various data mining (statistical and machine learning) algorithms. It includes tasks like automatic keyword extraction and text summarization. Automatic keyword extraction is the process of selecting words and phrases from the text document that can at best project the core sentiment of the document without any human intervention depending on the model. [1]. Summarization is a process where the most salient features of a text are extracted and compiled into a short abstract of the original document [2]. The important keywords extraction is the primary phase of text summarization. In keyword extraction as in the literature different methodologies used for keyword extraction process and different algorithms used under each methodology as shown in Figure 1.

Automatic Keyword Extraction for Text Summarization: A Survey. [5]



**Figure 1**: Classification of automatic keyword extraction on the basis of approaches used in existing literature

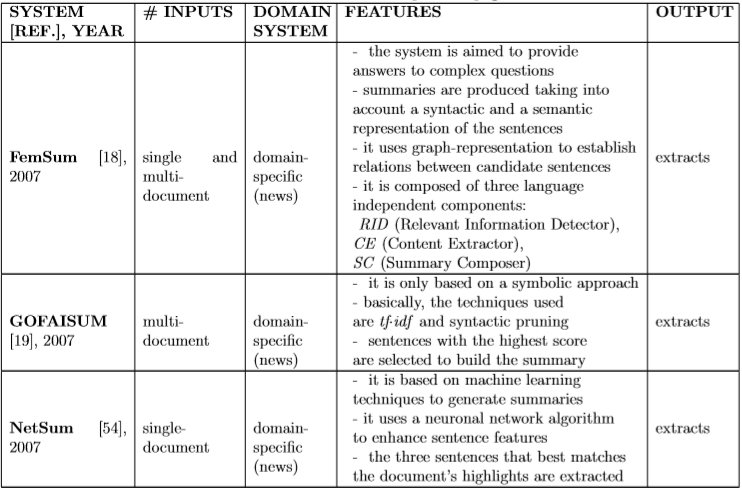
**Simple Statistical Approach -** These strategies are rough, simplistic and have a tendency to have no training sets. They concentrate on statistics got from non-linguistic features of the document, for example, the position of a word inside the document, the term frequency, and inverse document frequency. These insights are later used to build up a list of keywords. Cohen [6]

**Linguistics Approach -** This entry uses linguistic features to identify and extract the key words of the documents in the text. It incorporates the lexical analysis [7], syntactic analysis [8], discourse analysis [9], etc.

**Machine Learning Approach -** This approach requires manually annotated training data and training models. Hidden Markov model [10], support vector machine (SVM) [11], naive Bayes (NB) [12], bagging [8], etc. are commonly used training models in these approaches

**Hybrid Approach -** These approaches combine the above two methods or use heuristics, such as position, length, layout feature of the words. [13]

There are current uses that can be summarized as shown in Figure 2. Following is brief description about former Text Summarization Systems.



**Figure 2:** Text Summarization Systems [14]

Automatically summarized procedures can be divided as follows,

**Interpret -** This is where a representation of the document to be summarized is produced.

**Transform** - This is produced from where the representation of the document is turned into one of a summary of the document.

**Generate** - Here, summary text is the summary representation. Also known as synthesis.

Also Summaries can be divided in to Different categories.

**Extractive -** text summarization involves the selection of phrases and sentences from the source document to make up the new summary

**Abstractive -** text summarization involves generating entirely new phrases and sentences to capture the meaning of the source document.

Within and across these two summaries there are two sub categories of summarization Based on function and target reader.

**Indicative summary**

This type of summary categorizes the topics of the document and characteristics such as length, writing style, etc. This sort of summary is required for writing an abstract for a less structured document like an essay, editorial, or book. An indicative abstract is generally made up of three parts. [17]

**Informative summary**

This summary is for writing an abstract for a strictly structured document like an experiment, investigation, or survey etc. An informative abstract is made up of four parts. [17]

1. Purpose
2. Methodology
3. Results
4. Conclusions

**Query focused summarization**

This summarize text in the input document that is relevant to a specific user query.

Also there are categories of summarization Based on language.

**Mono-lingual summarization**

Here input and Target document in the same language.

**Multi-lingual summarization**

Here if the input document in multiple languages summary is also contain these languages.

**Cross-lingual summarization**

Here input document in one language and output document to be in another language.

In this project automatic text summarization is subject to a single document summarization with multiple languages. Also, the summaries produced are largely extracts of the document being summarized, rather than newly generated abstracts there for this produce summaries base on extractive method. Also this follows Simple Statistical Approach. Since this summarizer text in the input document this is a Query focused summarization. Mono-lingual summarization, Cross-lingual summarizations are two type of summarizations which are based on language.

1.2: Problem Landscape

1.3: Objective

Chapter 2: Project Management Tools

2.1: Project Management

The Project Management approach selected should be that which will deliver the best value for money in terms of getting the job done and ensuring adequate control. So clearly, with a small project, it is both practical and sensible to adjust the project management approach to the size of the project. project management method offers a convenient and effective structure for the management of IS projects. [18]. The Automatic Text summarization project is a result that comes from a recent strategic review. The analysis showed that automatic text summarizer has a good reputation in its market place. From the start of the project to the implementation level and testing level variety of project management tools dynamically adapted to manage schedule, scope, risk, Resources, Quality. Below are the main tools that helped to carry out the project to the success of the software project.

* GitHub
* Google Drive
* NetBeans
* TODO plugin
* GitKraken

2.1.1: File Management

2.2: Project Planning

Project planning is very important for the successful completion of the project. Those involved in the project need to know exactly what their role is, and what they expect to produce and, when it is wanted. This information will inform by The project plan. This gives the opportunity to think about what the project is about and how it is to be achieved, deliverables and when are they wanted, what skills we need and where do we get them, the problems we are likely to encounter and how shall we tackle them, the risks involved and how to overcome them. Since this automatic text summarizer is an individual project only one person need to think about above mentioned criteria. Following is the sequence of the process of project planning to automatic text summarizer.

**Understanding the requirement**

This ITS project starts with a requirements speciﬁcation. The main achievement of this project is build a prototype of automatic text summarizer which provide accurate and concise and ﬂuent summaries of longer text, such as news stories, blog articles and social media posts, using diﬀerent summarization techniques by Retrieval of most important information in response to a user query.

**Breaking down the work**

By looking at objectives of the project it is need to consider what need to be done to achieve that goal. What we're trying to produce and how to archive it.

**Product descriptions**

Purpose-

We have taken a large number of online information through the World Wide Web. For this reason, if someone searches on something online, the response obtained is lots of diﬀerent Web pages with many information which is impossible for person to read completely. There for The purpose of this is to make user easy to retrieve only important information.

Composition-

Derivation-

Quality/completion criteria-

Work packages

2.2.1: App Development Methodology

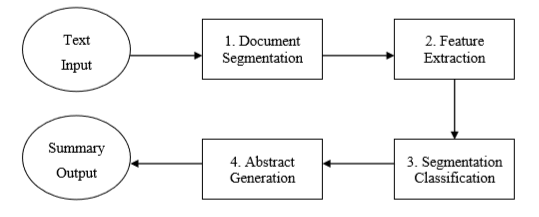
2.2.2: Challenges

Chapter 3: Design

3.1: Requirements

3.2: Architecture

The summarizer is written in Java following the MVC architecture. This uses the following external library itextpdf-5.1.0,commons-logging-1.2.jar, httpclient-4.5.5.jar, httpcore-4.4.9.jar, java-json.jar. The summary generation is a four step process document segmentation(**Interpret**), feature extraction **(Transform)**, segmentation classification(**Transform**), and abstract generation(**Generate**). As shown in the Figure 3.



**Figure 3:** Summarizer architecture [15]

The process of summarization begins with processing of input document which is broken down into paragraphs and subsequently into sentences. The tokenization includes tasks such as paragraph detection base on no of consecutive new line feed characters, extract sentences base on appearance and position of periods in the entire text. This maintains a list of sentences of the document and list of paragraphs in the document Each paragraph is responsible to store the sentences contained in it. To conduct this process, give two numbers which indicated which paragraph the sentence is apart and second which indicated sentence respect to the entire text. The second step in the summary creation process is feature extraction which is features are extracted from the text and intermediate values calculated to them. For example, scores assigned to features such as capitalization, phrases, or the sentence length, words population, word importance (key word). Also Summarizer maintains intersection matrix to keep intermediate values. Thirdly, for each sentence in the context, its score is computed as a weighted sum of feature intermediate values. Last, in step four, the highest scoring sentences are used to generate the document summary. This uses a sentence comparator on score to compare two sentence base on score and Sentence comparator for Summary compare two sentence base on its sentence number. To reads byte stream from a file system FileInputStream. java.io. FileInputStream used.

The Translation API's-TO DO

3.2.1: UML Class diagram

**Figure 4:**

Here design the classes according to the MVC architecture. There for this propose three types of objects in an application the Model, Views and Controllers.

**Model** - Objects hold data and define the logic for manipulating that data [16].

In this summarization application Paragraph, Sentence, GCPTranslator, are model classes as shown in the Figure 4.

**View** - objects represent something visible in the user interface [16].

In this summarization application View is the view class as shown in the Figure 4.

**Controller** - object acts as a Mediator between the Model and View objects. A Controller object communicates data back and forth between the Model objects and the View objects [16].

In this summarization application Algorithm, SentenceComparatorForSummary, SentenceComparatorOnScore, Validator are controller classes as shown in the Figure 4.

3.2.1: Class Diagram

3.3: Challenges

3.3.1 Implementation Challenges need to modify

1.To get the quality summary, quality keywords are required for text summarization.

2. There is no standard to identify quality keywords within or multiple documents. The extracted keywords are varying for applying different approaches of keyword extraction.

3. Multi-lingual text summarization is another challenging task.

ISSUES AND CHALLENGES OCCURS IN TEXT SUMMARIZATION In the area of text summarization, there are following research issues and challenges occurs during implementation

1.In the case of multi-document text summarization, several issues occurs frequently while evaluation of summary such as redundancy, temporal dimension, co-reference or sentence ordering, etc. which makes very difficult to achieve quality summary. Some other issues occurs such as grammaticality, cohesion, coherence which is harmful for summary.

2. The quality of summaries are varying from system to system or person to person. Some person feels some

set of sentences are important for summary, at the same time other person feel the other set of sentences are important for required summary.

3.3.1: Scalability

3.4: Programming Languages

3.5: Classification of Summarization Tasks

3.6: Methodology

3.7: Summarization Algorithm

3.8: Text Translate API

3.9: Unit Testing

Chapter 4: Development

4.1: Tools

4.1.1: IDE

4.1.2: Debugging

4.1.3: Other tools

4.2: Application

4.2.1: UI

4.2.2: Content

4.2.3: Features

Chapter 5: Testing

5.1: Test Plan

5.1.1: Market Fit

5.1.2: Performance and Reliability

5.2: Test Results

5.2.1: Market Fit

5.2.2: Performance and Reliability

Chapter 6: Experimental Results

6.1: Comparison of Summaries Generated with Hand Written Summaries

Chapter 7: Evaluations

7.1: Project Planning

7.2: Design

7.3: Future Development

7.4: Gained Lessons

7.5: Conclusion