TABLE OF CONTENTS

Chapter 1 : Introduction #

1.1: Project background #

1.2: Existing Projects #

1.3: objective #

Chapter 2 : project management #

2.1: project management #

*2.1.1: App Development Methodology #*

*2.1.2: project planning #*

2.1.2.1: Understanding the requirement #

2.1.2.2: Breaking down the work #

2.1.2.2.1: Work breakdown structure #

2.1.2.2.2: Product breakdown structure #

2.1.2.3: Understanding dependencies #

2.1.2.3.1: Dependency diagram. #

2.1.2.4: Using planning tools. #

2.1.3: Challenges. #

2.2: File management #

Chapter 3 : Design #

3.1:requirements #

3.2: Architecture #

3.2.1: UML Class diagram #

3.3: programming languages #

3.4: Classification of summarization tasks #

3.5: methodology #

3.6: Technical Challenges #

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 #

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 text 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 Intelligent text summarizer(ITS) is a tool that provides Generic summaries of a given document. Most Important Thing is this supports a wide variety of languages and this Summarization tool can be used to summarize a one single document on a percentage level and base on a keyword. Generate a summary report can be considered as some extra option that user can prefer.

This will be ideal for any user who is struggle with huge amount of information 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. “It has been more than 50 years since Luhn started his initial investigation on automatic text summarization (Luhn, 1958) [ 9]”, “Hovy defines a summary as “a text that is produced from one or more texts, that convey important information in the original text(s) and that is no longer than half of the original text (s) and usually significantly less than that” (Hovy, 2005).” 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. According to several literatures could find, to Realize this, from the documents texts mining is necessary [1]. 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 [1]. 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]. Also in [2] it says “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].

1. Automatic keyword extraction

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.

|  |  |
| --- | --- |
| Methodology | Algorithms |
| 1. **Simple Statistics** | Inverse document frequency(IDF) |
| Relative frequency ratio(RF) |
| Position of a key word(POK) |
| Word-co-occurrences(WO) |
| Term Frequency(TF) |
| Term weighting(TW) |
| PAT-tree |
| 1. **Linguistic** | Electronic dictionary(EDR) |
| NP chunks(Parsing) |
| Discourse analysis |
| Tree Tagger |
| WordNet |
| n-Grams |
| POS pattern |
| 1. **Machine Learning** | Naive Bayes |
| Bagging |
| HMM |
| SVM |
| 1. **Hybrid** | Combination of Simple statistics, Linguistic and Machine Learning. |

**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]

2. Text summarization

1. 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.

1. The different dimensions of text summarization can be generally categorized based on its input type. [19]

**Single Document summarization** -Single document summarization produces summary of single input document. [19]

**Multi Document Summarization** – Multi document summarization produces summary of multiple input document. [19]

1. Base on purpose summarization can be divided in to different categories.

**Generic Summarization** - Generic summarization purpose is to summarize all texts regardless of its topic or domain. Generic summaries make no assumptions about the domain of its source information and view all documents as homogenous texts. [19]

**Domain Specific Summarization** - this type of summarization requires domain specific. [19] knowledge bases to assist its sentence selection process.

**Query focused summarization** - Query-based summary contains only information which are queried by the user. [19]

1. Also Summaries can be divided in to Different categories base on output type.

**Extractive** - extracts are produced by identifying important sentences which are directly selected from the document. [19] There are several approaches to sentence extraction. namely, frequency based approach, feature based approach and machine learning based approach.

1. Frequency Based Approach

Two techniques that use frequency as a basic form of measure in text summarization are: word probability and term frequency-inverse document frequency. [19]

1. Feature Based Approach

Here Identify the features that reflects the relevance of that sentence.

Some common features are Title/Headline Word, sentence position, presence of title word and cue words, Sentence Length,

term Weight, Proper Noun. [19]

**Abstractive** -In abstractive summarization, the selected document sentences are combined coherently and compressed to exclude unimportant sections of the sentences. [19]

1. 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 containing these languages.

**Cross-lingual summarization** -Here input document in one language and output document to be in another language.

In this project 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. Frequency Based Approach, Feature Based Approach Both approaches are used extract sentences. Also this follows Simple Statistical Approach. Since this summarizer text in queried by the user this is a Query focused summarization. Mono-lingual summarization, Cross-lingual summarizations are two type of summarizations that produce which are based on language.

Create a table here.

1.2 Existing projects

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

**Figure 2:** Text Summarization Systems

Also following is some other popular online text summarizers: -

1.3: Objective

Chapter 2: Project Management Tools

The Project Management approach selected should be 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]

2.1: Project Management

The ITS project is a result that comes from a recent strategic review. The analysis showed that automatic text summarizers have a good reputation.

2.1.1: App Development Methodology

There is a considerable effect of app development methodology on each factor of the star model of project management.

**Budget** - Selected methodology determines whether it can assist to complete the project within the budget and to provide a good return on investment. Since this ITS is a prototype this is free of budget.

**Schedule** - Selected methodology determines whether it can assist to project to deliver on time, to project to get satisfactory requirement, to follow workflow as described in the scheduling. ITS project need to be deliver on time. As there were limited time period to this there should be a proper schedule. There must be greater transparency of the project. Sometimes Due to external factors had to do rescheduling. As the scheduling always provide plan B this was possible.

**Scope** - Selected methodology determines whether it can assist to have the well-defined scope and to have very clear scope of the project. In order to successfully get the final outcome of ITS project which is expected, also had to avoid the project is being way behind the schedule there for a well-defined scope were very important.

**Risk** - Selected methodology determines whether it can assist to manage project risks and opportunities and to meet the business objectives.

**Resources** - Selected methodology determines whether available material resources and whether it can assist to achieve maximum utilization of available resources.

**Quality** - Selected methodology determines whether it can assist to meet quality requirements and to meet client satisfaction to successful the project. [20]

collecting requirements is more signiﬁcant in light weight and in heavy weight methodologies. Also that in-time delivery is equal in both methodologies. The project scope is well deﬁned and clear compared to the heavyweight methodologies. Lightweight methodologies and heavyweight methodologies are of greater signiﬁcance for the availability and better utilization of resources. Since ITS project is a prototype which is individual I used agile development methodologies to continue the development of the project.

2.1.2: 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 need and where do they get, the problems likely to encounter and how will tackle them and the risks involved and how to overcome them. Since this automatic text summarizer is an individual project I thought about above all mentioned criteria. Following is the sequence of the process of project planning to ITS.

2.1.2.1: **Understanding the requirement**

After going through several current use of automatic text summarizers, it is observed that till now most of the summarization tools have developed with less features. Most of them only available for summarize a given document at a predefine level (System preferred way). There for user does not have a chance to give a summarization level which user is preferred. User need to strike to a one specific summarization. Also some of current uses could not able to summarize multiple language documents. When go through these several current uses I could found no current summarization tool has the feature of summarizing a given document based on user preferred key word. To overcome these, I proposed a prototype of new intelligent automatic text summarizer which provide accurate and concise and ﬂuent summaries of longer text using diﬀerent summarization techniques by Retrieval of most important information with more usability in response to a user document. Following are the business objectives in this Proposed ITS product.

1. Summarize a given document based on a user preferred key word.
2. Summarize a given document according to many summarization level.
3. Summarize documents with multiple languages.
4. Generate a summary report.

2.1.2.2: **Breaking down the work**

By looking at business objectives of the project it is need to consider what need to be done to achieve that goal. What is trying to produce and how to archive it. The work breakdown structure and product break down structure are two basic approaches to do this. Both structures will be illustrated in Figure 3 and Figure 4.

2.1.2.2.1: Work breakdown structure.

**Figure 3.**

2.1.2.2.2: Product breakdown structure.

**Figure 4.**

2.1.2.3: **Understanding dependencies**

Dependencies are fundamental to planning a project and, later, in understanding the effects of any problems encountered. [18]. The dependencies between the product/deliverable are illustrated in Figure 5.

2.1.2.3.1: Dependency diagram.

**Figure 5.**

2.1.2.4: **Using planning tools.**

It may take time to create various diagrams and charts for a project. As plans are requiring adjustments and revision during project progresses they cannot be produce completely at the start of the project. Use of computerized tools make replanting less of a chore. Many project planning tools are there on the market which are created for use on computers. Advantages of planning tools helped in following ways to ITS.

**Make Easy to re-planning** - Normally planning tools produce a project breakdown, create estimates, produce a schedule. With planning tools, I could do some adjustments to my plan in some points due to not working the plan. As planning tools have ability to review estimates and reschedule this was possible.

**Gave high Presentation quality** – Had high quality output in a variety of formats as they have flexible presentation and reporting facilities.

**Keep track with the progress** – This facilitated to track progress on a project and compare actual progress with the plan. Then in some instances I could identify where problems seem to be arising and to investigate them and decide how to respond them.

From the start of the project to the implementation level and testing level variety of project management tools dynamically alter to manage schedule, scope, risk, Resources, Quality. Below are the main project planning tools that helped to carry out the ITS project to the success of the software project.

* Trello
* TODO plugin

Below is a screen shot of Trello boards that demonstrating tasks of the ITS project.

**Figure 6.**

2.2: File Management.

Below are the main file Management tools that help to carry out the project.

* + GitHub
  + GitKraken
  + Google Drive

Around 90% of the files in ITS related with the development code base which were maintained by using a private Git repository on GitHub and used GitKraken. The remaining files were managed through a google drive cloud storage, where were in sync with local drive, as well as Git repository files.

Advantages of using a Git repository to ITS project shown as below:

* View the code changes across versions.
* If an error occurred there were ability to revert the code files/ revert the entire project back to the previous state.
* Due to flexibility it efficiency independent from the scale of the project.

Below is a screen shot of Git repository of the ITS project.

**Figure 7.**

2.1.2: Challenges

During the project planning phase could discover following challenges.

* At the start scope of the project was unclear.
* The planed schedule for some tasks not accurate.

Chapter 3: Design

3.1: Requirements

The main system requirements which is to be created by the software design are as follows.

* Provide an accurate summary of the document using various summarization methods.
* Create my own native algorithm to give comprehensive summaries.
* Select combination of features to assign score / weight to each sentence
* Assign scores to each sentence in accurate way
* If user place a document with different language, language detection must be done properly to give accurate comprehensive summary.
* Make design contains clearly understandable words, phrases and the commands for the users on behalf of using confusing system oriented words and phrases.
* Make interfaces are in a proper standard, understandability of the situation and actions.

3.2: Architecture

The ITS 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 and java-json.jar. The ITS project followed the general summary process and totally there are four steps which come as document segmentation(**Interpret**), feature extraction **(Transform)**, segmentation classification(**Transform**), and abstract generation(**Generate**). As shown in the Figure 3.

Feature Extraction

(Transform)

Document Segmentation.

(Interpret)

Abstract Generation

(Generate)

Segmentation Classification

(Transform)

**Figure 4:** Summarizer architecture

The process of summarization begins with processing of input document using my naive summarization algorithm. This algorithm 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 scores. 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 java.io. FileInputStream used. My native algorithm is such an important subject to talk in more details which is under 3.6 itself later in the report.

The Translation API's-TO DO

3.2.1: UML Class diagram

**Figure 5:**

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.

3.3: Programming Language

ITS project is written in java. There are many java solutions available for summarization of text. Those are,

1.Write a document

Different type of documents can be there which might be available to get summary as shown below.

* + Text File
  + Docks File
  + Doc File
  + PDF File

ITS write all texts in PDF, Doc in to a text file. In Java a text file can be write using java.io packages of j2sdk. For reading a file, File class, FileOutputStream class FileWriter class are available. Following is a screen shot of how ITS writing texts in to a text file. In the native algorithm used The java.io.InputStream.read method as it reads the next byte of the data from the input stream and returns an Integer value in the range of 0 to 255.

**Figure 6:**

2.Reading the document

ITS capable of get summary of a text file. As mention above after write particular document to a text file there is a way of reading that document in java. Using java.io package of j2sdk for reading the file, File class, FileInputStream and FileReader classes are available. Following is a screen shot of how ITS writing a reading a texts from a file.

**Figure 7:**

Further in java there are several ways to read a Doc file, PDF file. To reading a Doc there a many third party APIs available. Using apache POI can read a DOC file. This API has ability of manipulate various file format based upon Office Open XML standards (OOXML) and Microsoft OLE2 compound document format. Also there a set of APIs for reading PDF files. PDF box is one of the APIs which was designed by apache.

3.create summary of the document.

Java provided powerful technologies to build the native summarization algorithm in ITS. Here uses array list in java as it is more sense to model real world more than just arrays. Also compare method in Comparator class in java use to compare objects and collection has sort method to sort. In the algorithm Java Dictionary class is used. It is capable of store data in the form of key-value pairs (any type of objects). Keys should be unique as well as cannot be null. The java String class is representing character strings. In ITS used String class to share immutable strings rather to examine individual characters, to compare strings, to extract substrings, for searching strings all characters with uppercase or lowercase. Also string concatenation operation is provided good support. Also append method in StringBuilder class was helped to finally build the concatenate all the text in final summary. The method tostring used to string conversion purpose. Further parseDouble, parseInteger methods in Double, Integer classes used to conversion purpose of Integers and double values. There are classes StringTokenizer, StreamTokenizer to brake strings in to tokens. Since the tokenization method in StringTokenizer is simpler than the method in StreamTokenizer for ITS project in order to break string in to tokens string tokenizer class helped. This do not differentiate among numbers, identifiers, quoted strings and skip the comments. Set of delimiter specified at the creation time. Also used the trim method in java to checks this Unicode value before and after the string, if it exists then removes the spaces and returns the omitted string. There are many APIs that supports for the java. In order to generate Mono-lingual and Cross-lingual summarization I used google translation API which is support for java.

The Translation API's-TO DO

3.4: Classification of Summarization Tasks

**Figure 8:**

3.5: Methodology

A very native summarization algorithm is introduced in ITS. This Algorithm is capable of cuts around 25 %, 50 %, 75 % words from the original document. Specialty of this algorithm from other summarization algorithm is this uses combination of features to assign scores to particular sentence which are not used by many other summarizations. like word probability (no of common words) and uses intersection matrix, word position, summaries according to a user given key word for that it uses user search result function, sentence count is selected to the summary according to the user preferred amount in the option. Translate multiple languages that user place documents in to general language english and produce summaries base on above mentioned features and then produce final summary in user required language.

As display in the figure 4 the first step in the summarization process is to separate the document into different segment. This algorithm grabbed key sentence from each paragraph from the original document. In order to do that need to follow following sequence.

1. Entire text in the document divided in to paragraphs.

2. Each divided paragraphs break in to sentences.

3. Each sentence break in to words.

Current day there are two main approaches to do summarization.

1. Analyze and re-write the text in a short format.
2. Extract/grabbed key sentence from the original text and put them together in a proper way to preserve context and meaning. (extractive)

In ITS algorithm it follows the second approach which I mentioned above as with first approach could not achieve considerable result.

Following describe each function of the algorithm.

1. **The Sentence group function.**

Here takes sentence with paragraph number and group them with similar paragraph number.

1. **The Probability function.**

Here according to the second step in figure 4 following the feature extraction.

This function receives two sentences then according to Frequency Based Approach Counting each considered word in a one sentence by comparing with other words in second sentence. Probability of a considered word is shown as below,

f(w)=n(w)/N

Where,

n(w) = The frequency counts of considered word w when compare with two sentences.

N= The total number of words in both considered sentences.

Finally, add all probabilities with frequency count then this will be return as number of common words.

Ex: Assume two sentences are,

**Sentence 1 (s1) -** Mohandas Karamchand Gandhi was an eminent freedom activist and he is an influential political leader.

**Sentence 2 (s2)-** Gandhi is known by different names, such as Mahatma, Bapuji, and Father of the Nation as he is an influential political leader.

Total number of words in both sentences = 37

**Figure 9:**

|  |  |
| --- | --- |
| Word | Probability |
| Mohandas | 0/37 |
| Karamchand | 0/37 |
| Gandhi | 1/37 |
| was | 0/37 |
| an | 2/37 |
| eminent | 0/37 |
| freedom | 0/37 |
| activist | 0/37 |
| and | 1/37 |
| he | 1/37 |
| is | 2/37 |
| influential | 1/37 |
| political | 1/37 |
| leader | 1/37 |
| **Total** | **10/37** |

The total Number of Common words between Sentence 1 relative to Sentence 2 = 10

1. **The Intersection matrix function**

This function receives count of common tokens of each sentences relative to other sentences from the probability function. These resulting values then normalize and calculate a value for the intersection between them with average length of the two sentences. The intersection score (normalized values) between two sentences mentioned in the Figure 9:can be shown as below,

f (s1, s2) = | {w | w in s1 and w in s2} | / ((|s1| + |s2|) / 2)

Where,

w = No of common words

|s1| = size of sentence 1(total no of words in sentence 2).

|s2| = size of sentence 2 (total no of words in sentence 1).

Ex: According to Figure 9**,**

| {w | w in s1 and w in s2} | = 10

(|s1| + |s2|) = 37

((|s1| + |s2|) / 2 = 37/2 = 18.5

There for,

f (s1, s2) = 10 / 18.5 = 0.54

these calculated intersection scores will be store in a matrix. It is a two-dimensional array.

Assume there is an another sentence,

**Sentence 3 (s3)-** Every year, political leader Gandhi’s birthday is celebrated as Gandhi Jayanti, a national holiday in India, and also observed as the International Day of Nonviolence

**Figure 10:**

|  |  |
| --- | --- |
| Word | Probability |
| Mohandas | 0/40 |
| Karamchand | 0/40 |
| Gandhi | 2/40 |
| was | 0/40 |
| an | 0/40 |
| eminent | 0/40 |
| freedom | 0/40 |
| activist | 0/40 |
| and | 1/40 |
| he | 0/40 |
| is | 1/40 |
| influential | 0/40 |
| political | 1/40 |
| leader | 1/40 |
| **Total** | **6/40** |

The total Number of Common words between Sentence 1 relative to Sentence3 = 6

According to f (S1, S3) = | {w | w in S1 and w in S3} | / ((|S1| + |S3|) / 2),

| {w | w in s1 and w in s3} | = 6

(|S1| + |S3|) = 40

(|S1| + |S3|) / 2 = 20

There for,

f (s1, s3) = 6 / 20 = 0.3

Intersection matrix: -

**Figure 11.**

**[0][0] [0][1] [0][2]**

|  |  |  |
| --- | --- | --- |
| F(S1,S1) | 0.4 | 0.3 |

As mention above all the intersection scores will be store in the matrix. [0][1] will hold the intersection score between sentence #1 and sentence #2. [0][2] will hold the intersection score between sentence #1 and sentence #3. [0][0] will hold the intersection of itself.Here I just converted my text into a fully-connected weighted graph. Each sentence is a node in the graph and the two-dimensional array holds the weight of each edge.

1. **The user search result function.**

Following the second step in figure 4 feature extraction, this is one of the most important feature in which importance of sections computed by comparing user search key word (According to the feature base approach). This summarization is performed for each section using a user given key word. With this function find sentences with most number of key word count. If a particular sentence key word count is greater than 0 that sentence will be selected and those will be store in the Array list call key-word array.

Ex: - Conceder following 3 sentences: -

**Sentence 1 (s1) -** Mohandas Karamchand Gandhi was an eminent freedom activist and he is an influential political leader.

**Sentence 2 (s2)-** Gandhi is known by different names, such as Mahatma, Bapuji, and Father of the Nation as he is an influential political leader.

**Sentence 3 (s3)-** Every year, political leader Gandhi’s birthday is celebrated as Gandhi Jayanti, a national holiday in India, and also observed as the International Day of Nonviolence.

If,

1. User given key word - “Mohandas”

Selected Sentences – “Mohandas Karamchand Gandhi was an eminent freedom activist and he is an influential political leader.”

Number of selected sentences - 1

1. User given key word – “Gandhi”

Selected Sentences – **“**Mohandas Karamchand Gandhi was an eminent freedom activist and he is an influential political leader.”

“Gandhi is known by different names, such as Mahatma, Bapuji, and Father of the Nation as he is an influential political leader.”

“Every year, political leader Gandhi’s birthday is celebrated as Gandhi Jayanti, a national holiday in India, and also observed as the International Day of Nonviolence.”

Number of selected sentences - 3

Finally, these selected sentences will also send to the intersection function which is mentioned in 2. In order to get the intersection score (normalized values) between each two sentences.

1. **The sentence Dictionary function**

“Heart” of the algorithm is the sentence dictionary function. Here the third step in figure 4 Segmentation classification goes. Here receives the text as input, and calculate an individual score for each sentence and store it in a key-value dictionary (In a hash map) where sentence is the key and value is the score of the particular sentence. Scores are given to two features.

1.The first feature is the Word Probability/frequency - Here calculate intersection scores between each two sentence in a matrix as described under the intersection matrix function and then calculate an individual score for each sentence base on probability. Assume following is set of intersection scores for S1, S2, S3 sentences:

**Figure 12.**

|  |  |  |
| --- | --- | --- |
| F(S1,S1) | F(S1,S2) | F(S1,S3) |
| F(S2,S1) | F(S2,S2) | F(S2,S3) |
| F(S3,S1) | F(S3,S2) | F(S3,S3) |

S1

S2

S3

Score Base On Probability Si = ∈ f (Si, Sj)

Consider S1,

Score Base On Probability S1 = F(S1,S1) + F(S1,S2)+F(S1,S3)

Consider S2,

Score Base On Probability S2= F(S2,S1)+F(S2,S2)+ F(S2,S3)

Consider S3,

Score Base On Probability S2= F(S3,S1)+F(S3,S2)+ F(S3,S3)

2. The second feature is the Sentence position - Here calculate scores according to the sentence position. Sentences appearing in the beginning of the section are given higher weightage. First get all the sentences in the beginning of the section /document and then calculate score base on position as shown in equation.

Score Base On Position Si = 1/position of the Si in the context

Where,

position of the sentence in the context = sentence number relative to document

Consider S1,

Score Base On Position S1 = 1/ position of the S1 in the context

Consider S2,

Score Base On Position S2 = 1/ position of the S2 in the context

Consider S3,

Score Base On Position S3 = 1/ position of the S3 in the context

Final Individual scores for sentence calculating as follow:

Final Score for Si = Score Base On Probability Si+ Score Base On Position Si

Consider s1,

Final Score for S1= Score Base On Probability S1+ Score Base On Position S1

Consider s2,

Final Score for S2= Score Base On Probability S2+ Score Base On Position S

Consider s3,

Final Score for S3= Score Base On Probability S3+ Score Base On Position S3

Finally calculated individual scores for each sentence store in a key-value dictionary (In a hash map) where sentence is the key and value is the score of the particular sentence.

1. **The percentage calculation**

User has rights to select a percentage level to be summarize a document. Percentage levels that provided to users are 25 %,50 %,75 %,100 % then each percentage levels divided in to 10 and from that amount of sentences will be select 1 sentence as shown in below.

* 25 %

25 / 10 = 2.5

From the answer always consider the integer value. Here for each 2 sentences select a one sentence.

Assume there are 10 sentences. From 10 sentences 5 sentences will be selected.

* 50 %

50 / 10 =5

From the answer always consider the integer value. Here for each 5 sentences select a one sentence.

Assume there are 10 sentences. From 10 sentences 2 sentences will be selected.

* 75 %

75 / 10 =7.5

From the answer always consider the integer value. Here for each 7 sentences select a one sentence.

Assume there are 10 sentences. From 10 sentences 1 sentences will be selected.

* 100 %

100 / 10 = 10

From the answer always consider the integer value. Here for each 10 sentences select a one sentence.

Assume there are 20 sentences. From 20 sentences 2 sentences will be selected.

1. **The Summary Building function**

Here according to the figure 4 generate the most comprehensive accurate shortest summaries of user given documents as user requirement. There are two type of summarizations generate by ITS.

1.General summarization

* According to the user given percentage from every paragraph extract sentences with the highest score using sentence dictionary (hash map).
* Sort the selected sentences in the order of parent text to preserve context and meaning.

2.Key word summarization

* According to the user given key word from every paragraph extract sentences that contain given key word and then store in Array list call key-word array
* For the sentences in key-word array assign scores and finally store total scores for individual sentences in the dictionary.
* According to the user given percentage from every paragraph extract sentences with the highest score using sentence dictionary (hash map).

3.6: Technical challenges.

* Researching about natural language processing, which is how computers can analyze, understand and derive meaning from a human.
* Selecting the right libraries and having an in depth knowledge of those by going through their documentation.
* Evaluate summarizer.

Figure 12.

summarization model

3.7: Methodology

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