CHAPTER 1

INTRODUCTION

1.1 Aim and Objective

The concept of automatic summarization is gaining a momentum and is being explored by many researchers and developers. Our project focuses on the extractive part of automatic summarization. Our aim is to enable the users to quickly go through the IEEE papers and grasping maximum important data. Often reading through IEEE papers about a particular topic is time consuming as it contains a minimum of 5000- 7000 words. What if a user could go through around 2000-3000 words and retain the important information. The goal of automatic text summarization is condensing the source text into a shorter version preserving its information content and overall meaning. A summary can be employed in an indicative way as a pointer to some parts of the original document, or in an informative way to cover all relevant information of the text. In both cases the most important advantage of using a summary is its reduced reading time. A good summary system should reflect the diverse topics of the document while keeping redundancy to a minimum. Summarization tools may also search for headings and other markers of subtopics in order to identify the key points of a document.

- Summaries reduce reading time.
- When researching documents, summaries make the selection process easier.
- Automatic summarization improves the effectiveness of indexing.
- Automatic summarization algorithms are less biased than human summarizers.
- Personalized summaries are useful in question-answering systems as they provide personalized information.
- Using automatic or semi-automatic summarization systems enables commercial abstract services to increase the number of texts they are able to process.

1.2 Problem Statement

Textual information in the form of digital documents quickly accumulates to huge amounts of data. Most of this large volume of documents is unstructured: it is unrestricted and has not been organized into traditional databases. Processing documents is therefore a perfunctory task, mostly due to the lack of standards. We cannot possibly create summaries of all of the text manually; there is a great need for automatic methods.

Going through long IEEE papers can be difficult and time consuming. If you need to go through many papers– good luck. What if you want to know everything the document talks about. Here the abstract fails. A summary which gives you detailed information about the contents of the paper but is way shorter will take less time. For this purpose, we need a text summarizer which will summarize the entire paper for the user.

Drawbacks of existing system:

- There is no such system which only summarizes IEEE papers.
- Existing systems take documents in text or doc format but not in pdf format.
- There is a limit on the number of characters the summarizer takes
- The summarizers don't take the format of the document into consideration or the images or tables in the document.

1.3 Definition, Abbreviation and Acronym

The definitions of the terms, which are used in this SRS document, are shown below:

GUI Graphical User Interfaces

API Application Program Interface

SRS Software Requirement Specification

PHP Hypertext Preprocessor

1.4 Methodology

The complete development of this system can be divided into the following stages:

Problem definition stage: This is the very first stage to develop any project. It actually defines the aim and the concept of the project. In this section the purpose i.e. why, this particular project is required, is explained.

Designing block diagram: At this stage we have categorized the whole system into different parts. These block diagrams will be helpful in understanding the concept and working of the integrated system.

Developing Flowchart for software: To get the logical flow of the software, the development of flowchart is having a prominent role. So, we have to analyze the complete system and organized the flowchart in such a manner that one can understand the complete working of the software.

Writing the actual code: After the development of the block diagrams and flowchart we started writing the actual code with the help of algorithm and flowcharts. The most crucial part of our project involved implementing the data flow using Jason. Thus, with the help of stated documents, we were successful in writing the actual code.

Compiling the code: After the actual code was written, then it was being complied. At the time of compilation, many compilation errors such as syntax errors were also being corrected, if any.

Testing: This time we tested our project for actual working, using different types of inputs to check if they are properly encrypted.

Documentation: After the above all procedures, the documentation of the project was being done, which covered the various topics such as project overview, implementation, designing, etc. along with all the necessary figures and tables.

1.5 Features

- Converting the IEEE papers into required format.
- Extracting images and tables from the document.
- Analyzing the template of the document.
- Preprocessing the data.
- Separating the document into segments according to the IEEE paper.

CHAPTER 2

LITERATURE SURVEY

2.1 Text Rank & PageRank Algorithm

Text Rank is a graph-based ranking model for text processing which can be used in order to find the most relevant sentences in text. In order to find the most relevant sentences in text, a graph is constructed where the vertices of the graph represent each sentence in a document and the edges between sentences are based on content overlap, namely by calculating the number of words that 2 sentences have in common. Based on this network of sentences, the sentences are fed into the Page rank algorithm which identifies the most important sentences. When we want to extract a summary of the text, we can now take only the most important sentences.

Following are the steps:

1. Separate the text into sentences
2. Build a sparse matrix of words and the count it appears in each sentence
3. Normalize each word with tf-idf
4. Construct the similarity matrix between sentences
5. Calculate page rank for sentences.

Following is the example of how Text Rank and PageRank work:

Document 1: The game of life is a game of everlasting learning
Document 2: The unexamined life is not worth living
Document 3: Never stop learning

Step 1: Term Frequency (TF)

Term Frequency also known as TF measures the number of times a term (word) occurs in a document. Given below are the terms and their frequency on each of the document.

TF for Document 1

Document1	the	game	of	life	is	a	everlasting	learning
Term Frequency	1	2	2	1	1	1	1	1

TF for Document 2

Document2	the	unexamined	life	is	not	worth	living
Term Frequency	1	1	1	1	1	1	1

TF for Document 3

Document3	never	stop	learning
Term Frequency	1	1	1

In reality each document will be of different size. On a large document the frequency of the terms will be much higher than the smaller ones. Hence we need to normalize the document based on its size. A simple trick is to divide the term frequency by the total number of terms. For example in Document 1 the term game occurs two times. The total number of terms in the document is 10. Hence the normalized term frequency is 2 / 10 = 0.2. Given below are the normalized term frequency for all the documents.

Normalized TF for Document 1

Document1	the	game	of	life	is	a	everlasting	learning
Normalized TF	0.1	0.2	0.2	0.1	0.1	0.1	0.1	0.1

Normalized TF for Document 2

Document2	the	unexamined	life	is	not
Normalized TF	0.142857	0.142857	0.142857	0.142857	0.142857

	worth	living
	0.142857	0.142857

Normalized TF for Document 3

Document3	never	stop	learning
Normalized TF	0.333333	0.333333	0.333333

Step 2: Inverse Document Frequency (IDF)

The main purpose of doing a search is to find out relevant documents matching the query. In the first step all terms are considered equally important. In fact certain terms that occur too frequently have little power in determining the relevance. We need a way to weigh down the effects of too frequently occurring terms. Also the terms that occur less in the document can be more relevant. We need a way to weigh up the effects of less frequently occurring terms. Logarithms help us to solve this problem.

Let us compute IDF for the term **game**

IDF(game) = 1 + loge(Total Number Of Documents / Number Of Documents with term game in it)

There are 3 documents in all = Document1, Document2, Document3

The term game appears in Document1

IDF(game) = 1 + loge(3 / 1)

= 1 + 1.098726209

= 2.098726209

TERMS	**IDF**
the	1.405507153
game	2.098726209
of	2.098726209
life	1.405507153
Is	1.405507153
A	2.098726209
everlasting	2.098726209
learning	1.405507153
unexamined	2.098726209
not	2.098726209
worth	2.098726209
living	2.098726209
never	2.098726209
stop	2.098726209

Step 3: TF*IDF

TF*IDF of Document1

TERM	TF	IDF	TF*IDF
the	0.1	1.405507153	0.1405507153
game	0.2	2.098726209	0.4197452418
Of	0.2	2.098726209	0.4197452418
Life	0.1	1.405507153	0.1405507153
Is	0.1	1.405507153	0.1405507153
A	0.1	2.098726209	0.2098726209
everlasting	0.1	2.098726209	0.2098726209
learning	0.1	1.405507153	0.1405507153

TF*IDF of Document2

TERM	TF	IDF	TF*IDF
the	0.142857	1.405507153	0.200786535356121
unexamined	0.142857	2.098726209	0.299817730039113
Life	0.142857	1.405507153	0.200786535356121
is	0.142857	1.405507153	0.200786535356121
not	0.142857	2.098726209	0.299817730039113
worth	0.142857	2.098726209	0.299817730039113
living	0.142857	2.098726209	0.299817730039113

TF*IDF of Document3

TERM	TF	IDF	TF*IDF
never	0.333333	2.098726209	0.699574703424597
stop	0.333333	2.098726209	0.699574703424597
learning	0.333333	1.405507153	0.468501915830949

Step 4: Vector Space Model – Cosine Similarity

D1=document1 D2=document2

Cosine Similarity (d1, d2) = Dot product(d1, d2) / ||d1|| * ||d2||

Dot product (d1,d2) = d1[0] * d2[0] + d1[1] * d2[1] * … * d1[n] * d2[n]

||d1|| = square root(d1[0]^2 + d1[1]^2 + ... + d1[n]^2)

||d2|| = square root(d2[0]^2 + d2[1]^2 + ... + d2[n]^2)

Cosine similarity for Document1,Document2 & Document3:

CS(D1,D2)=(0+(0.1405507153)*(0.200786535356121))/0.4899

=0.057

CS(D1,D3)=(0+(0.1405507153)*(0.468501915830949))/1.1431

=0.057

CS(D2,D3)=0

Similarity Matrix

	D1	D2	D3
D1	1	0.057	0.057
D2	0.057	1	0
D3	0.057	0	1

Step 5: Calculating page ranks for sentences

PR(D1) =1-d/N * {CS(D2)/1+CS(D3)/1}

PR(D1) =1-0.85/3 *{0.057/1+0.057/1}

=0.0057

PR(D2) =1-d/N * {CS(D1)/1+CS(D3)/1}

PR(D2) =1-0.85/3 *{0.057/1+0}

=0.0029

PR(D3) =1-d/N * {CS(D1)/1+CS(D2)/1}

PR(D3) =1-0.85/3 *{0.057/1+0}

=0.0029

Hence, as the words present in D2 and D3 are present in D1, D1 has higher page rank.

2.2 Technical Papers

2.2.1 A Frequent Term and Semantic Similarity based Single Document Text Summarization Algorithm[7]

In this paper a frequent term-based text summarization algorithm is designed and implemented in java. The designed algorithm works in three steps. In the first step the document which is required to be summarized is processed by eliminating the stop word and by applying the stemmers. In the second step term-frequent data is calculated from the document and frequent terms are selected, for these selected words the semantic equivalent terms are also generated. Finally, in the third step all the sentences in the document, which are containing the frequent and semantic equivalent terms, are filtered for summarization. The system is divided into three major parts, an input text document, a summarizer algorithm and a summarized text document as output. The summarizer algorithm is further divided into the three parts – the text pre-processing module, frequent terms generation module along with the semantically similar terms and sentence filtering module for summarization. The overall methodology of semantic similarity bases single document summarization can be expressed in terms of an algorithm. The algorithm takes two input parameters – the input text document and number of frequent terms. As the output it generates a summarized text document along with the two measures compression ratio and retention ratio. The proposed algorithm is implemented using open source technologies and is verified over the standard text mining corpus. The discovered results are interesting and meaning of the summarized document is also preserved. The future direction for the proposed work is to apply the similar concept in multi text document summarization.

2.2.1 A Review Paper on Extractive Techniques of Text Summarization[2]

In this paper, we have learned about extractive summarization techniques. Automatic text summarization can be classified into two categories based on their approach: summarization based on abstraction and summarization based on extraction. Systems for extractive summarization are typically based on technique for sentence extraction and attempt to identify the set of sentences that are most important for the overall understanding of a given document. Extractive summaries are formulated by extracting key text segments (sentences or passages) from the text, based on statistical analysis of individual or mixed surface level features such as word/phrase

frequency, location or cue words to locate the sentences to be extracted. The “most important” content is treated as the “most frequent” or the “most favourably positioned” content.

Following are some of the extractive methods used:

A. Term frequency - Inverse document frequency(TF-IDF)

We combine the definitions of term frequency and inverse document frequency, to produce a composite weight for each term in each document. The TF-IDF weighting scheme assigns to term t a weight in document given by

tf-idft,d = tft,d × idft.

B. Cluster based method

Clustering is a process of grouping set of objects in such a way that objects in same group are similar to each other. Organization of documents is done in such a way that they address different topics in some sequence. This is also applicable to summaries. Sentence selection is based on cluster Ci. Another factor for selection is location of sentence Li. The last factor that increases the score of a sentence is its similarity to the first sentence in the document to which it belongs (Fi). The overall score (Si) of a sentence i is a weighted sum of the above three factors: Si =W1 *Ci + W2 *Fi+ W3 *Li where Si is the score of sentence Ci, Fi and Li.

C. Graph theoretic approach

Graph theoretic representation of text provides a method to identify these themes. After the pre-processing steps, stop word removal and stemming, sentences in the documents are represented as nodes in an undirected graph. Two nodes are connected with edges; if they have common words here every node is sentence.

D. Machine Learning approach

Given a set of training document and their extractive summaries, the summarization process is modelled as a

classification problem: sentences are classified as summary sentences and non-summary sentences based on the features that they possess. P (s∈<S | F1, F2,.., FN) = P (F1, F2, ..., FN | s∈S) * P (s∈S) / P (F1, F2,..., FN) where s is a sentence from the document collection, F1, F2…FN are features used in classification. S is the summary to be generated, and P (s∈< S | F1, F2,.., FN) is the probability that sentence s will be chosen to form the summary given that it possesses features F1,F2…FN.

E. Query based extractive text summarization

In query based text summarization [10], the scoring of the sentences of a given document is based on the frequency counts of words or phrases. Higher scores are given to the sentences containing the query phrases rather than the ones with single query words. The sentences with highest scores are then extracted for the output summary together with their structural context. Portions of text may be extracted from different sections or subsections. The resulting summary is the union of such extracts. In the sentence extraction algorithm, whenever a sentence is selected for the inclusion in the summary, some of the headings in that context are also selected.

Various methods of extractive approach have emerged in the past. But it is hard to say how much greater interpretive sophistication, at sentence or text level contributes to performance. Without the use of Natural Language Processing, the generated summaries may not be much accurate in terms of semantics.

2.2.1 Automatic Summarization of News Articles using TextRank[6]

This paper discusses implementation of Textrank algorithm**.** TextRank is a graph-based, surface-level algorithm. In this algorithm, the similarity values of the edges are used to weight the vertices. The basic premise of a graph-based ranking model is similar to voting or recommendation. The greater the number of links or “votes” to a vertex, the higher the importance or relevance of that vertex. The score associated with a vertex is determined based on the votes that are cast for it, and the score of the vertices casting these votes. TextRank is an approach inspired by the PageRank algorithm used at Google for ranking web-pages [4]. We first build a graph associated with the text, where the graph vertices represent the units to be ranked. With respect to sentence extraction, the goal is to rank entire sentences, and therefore a vertex is added to the graph for each sentence in the text. Links between sentences are determined by a “similarity” relation between them, wherein “similarity” refers to sentences having some content in common. Similarity measures such as Word overlap, TF-IDF (Term Frequency - Inverse Document Frequency), cosine similarity etc. can be used for ranking. After the ranking algorithm is run with the graph as input, sentences are sorted in descending order of their score, and the highest ranked sentences are selected to construct the summary. TextRank succeeds in identifying the most important sentences in a text based on information exclusively drawn from the text itself. Thus, in this paper we have seen an explanation of the TextRank algorithm used for extractive summarization.

2.3 Existing Systems

2.3.1 Sumplify

It does summarization of text according to the requirements of the user. User can upload or choose file and select length of the summary for summarization

2.3.2 Tools4Noobs

It is also an online summarizing tool. Here the user can select a threshold or select number of lines and select minimum sentence length as well as minimum word length and summarize the content.

2.3.3 FreeSummarizer

It is a free online tool to automatically summarize any text in just few clicks. There is no hassle in using this tool because you don’t have to install any tool on your PC with it. Just enter the text onto it and let it start with the summarization without any hassles.

2.3.4 TextCompacter

This free online summarization tool was created to help struggling readers process overwhelming amounts of information. However, the general approach will help any busy student, teacher, or professional.After text is placed on the page, the web app calculates the frequency of each word in the passage. Then, a score is calculated for each sentence based on the frequency count associated with the words it contains. The most important sentence is deemed to be the sentence with the highest frequency count.

2.3.5 WikiSummarizer

WikiSummarizer is a Web-based application specializing in automatic visualization and summarization of Wikipedia articles. WikiSummarizer automatically summarizes the Wikipedia articles. The program identifies the most important keywords and ranks them by relevancy. For each keyword the most significant sentences in the original text are presented to the reader. You instantly get the headlines with the most important sentences and keywords.The blending of visualization with summarization, knowledge browsing, mind mapping provides you with a wide range of means to explore relevant content. At a glance, without much reading, you immediately spot the key information chunks.

CHAPTER 3

REQUIREMENT

3.1 Functional Requirement

- The system should convert the pdf document into txt document.
- The system should extract tables and images from the document.
- The system should analyze the template of the document and properly segment the document.
- The system should combine the summaries and display it with respective images and tables.
- The system should identify and remove all the noisy data from the document.
- The system should take the individual segments and generate appropriate summaries.

3.2 Non-Functional Requirement

- The system should accept only IEEE papers.
- The system should take only one document at a time.

Other Non-functional Requirements are:

- Reliability
- Maintainability
- Portability
- Reusability
- Resource Utilization

3.3 Hardware Requirement & Software Requirement

- System: i3 2.4 GHz
- Hard Disk: 100 GB
- Monitor: 14’ Color Monitor
- Ram: 4 GB
- Mouse: Optical Mouse
- A reliable internet connection
- Python 3.6.4 and xampp

3.4 Tools & Packages

- **Sklearn**

  It is a free software machine learning library for the Python programming language.
- **NLTK**

  The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in the Python programming language.
- **Punkt Sentence Tokenizer**

  This tokenizer divides a text into a list of sentences, by using an unsupervised algorithm to build a model for abbreviation words, collocations, and words that start sentences. It must be trained on a large collection of plaintext in the target language before it can be used.
- **TfidfTransformer()**

  Transform a count matrix to a normalized tf or tf-idf representation. The goal of using tf-idf instead of the raw frequencies of occurrence of a token in a given document is to scale down the impact of tokens that occur very frequently in a given corpus and that are hence empirically less informative than features that occur in a small fraction of the training corpus.
- **pagerank(nx graph)**

  Return the PageRank of the nodes in the graph. A NetworkX graph. Undirected graphs will be converted to a directed graph with two directed edges for each undirected edge**.**
- **CountVectorizer()**

  Convert a collection of text documents to a matrix of token counts.
- **PyPDF2**

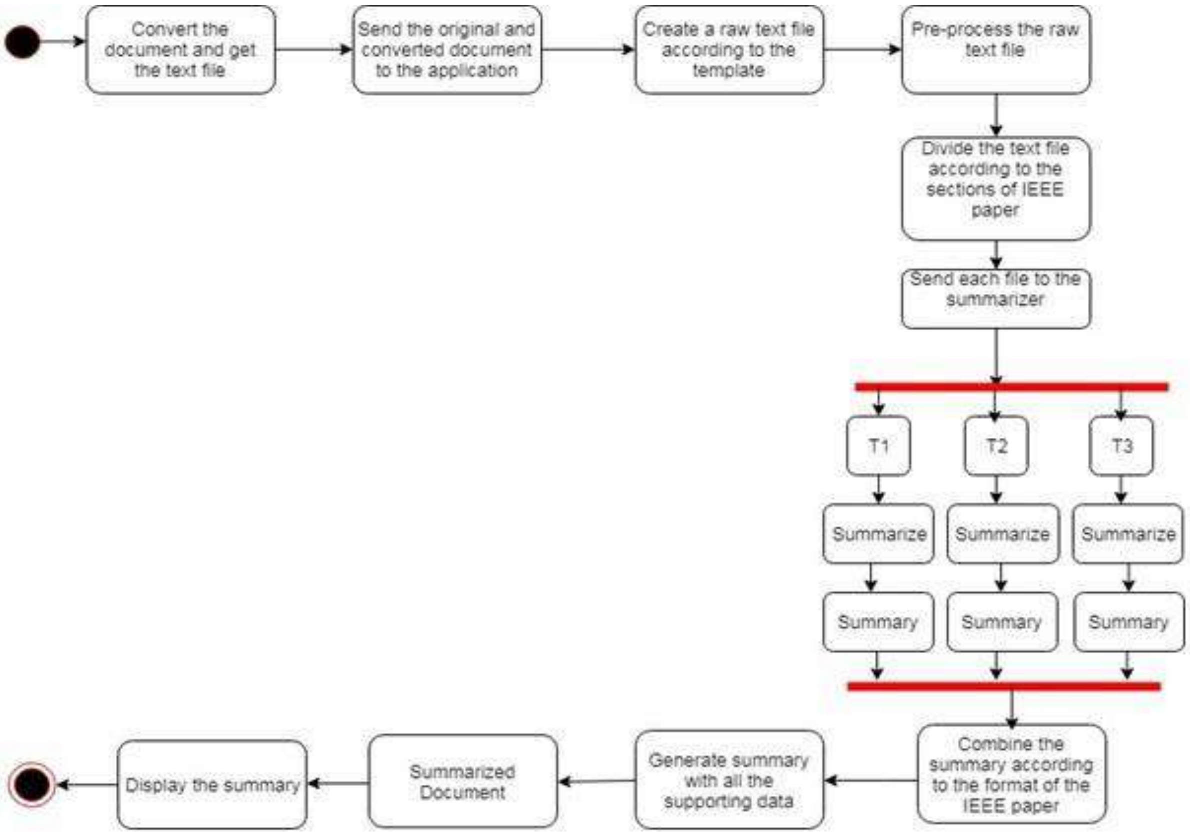
  Used to extract images and tables from the pdf.
- **Pre-processing**

  We use regular expressions to identify noise and remove the noise. We use stop words from nltk to identify stop words.

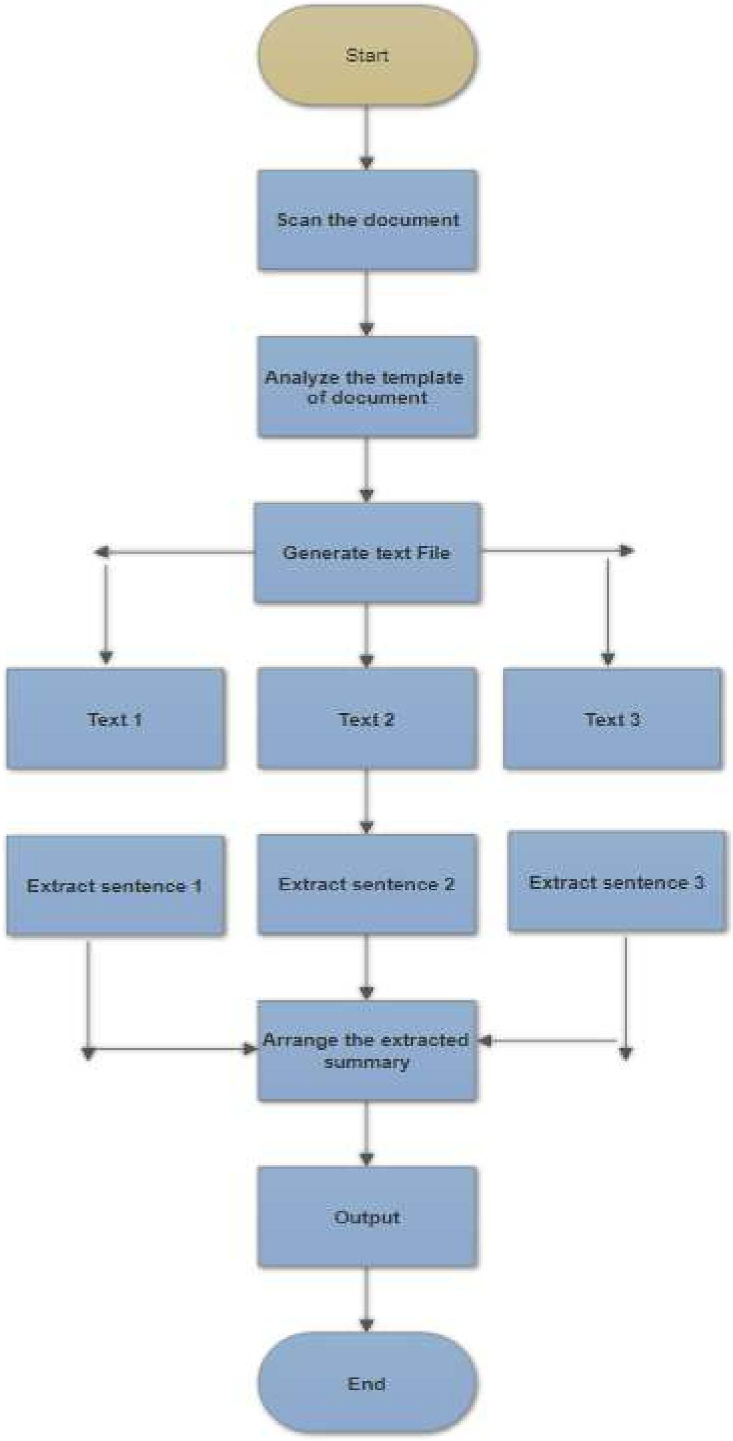
CHAPTER 4

DESIGN

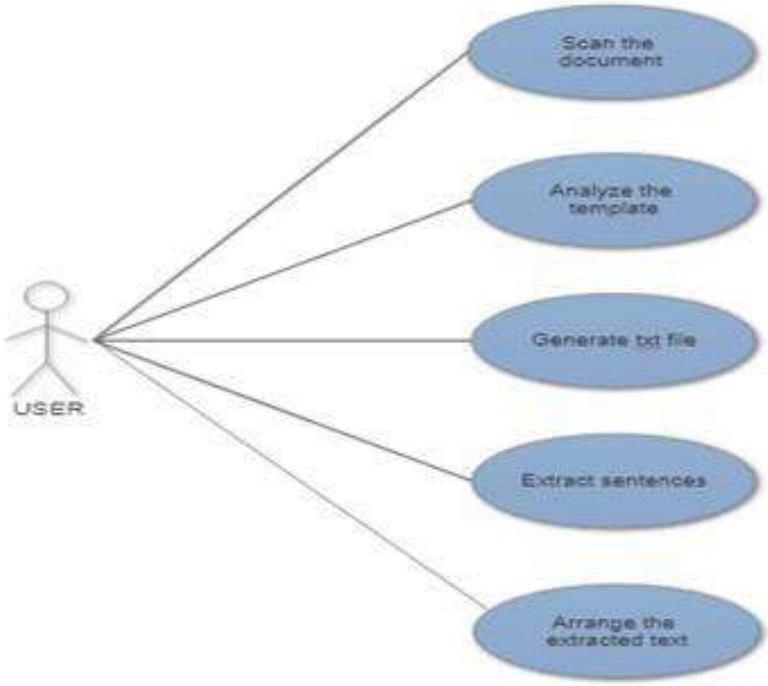
4.1 System Diagram


Convert the document and get the text file
Send the original and converted document to the application
Create a raw text file according to the template
Pre-process the raw text file
Divide the text file according to the sections of IEEE paper
Send each file to the summarizer
T1
T2
T3
Summarize
Summarize
Summarize
Summary
Summary
Summary
Combine the summary according to the format of the IEEE paper
Generate summary with all the supporting data
Summarized Document
Display the summary

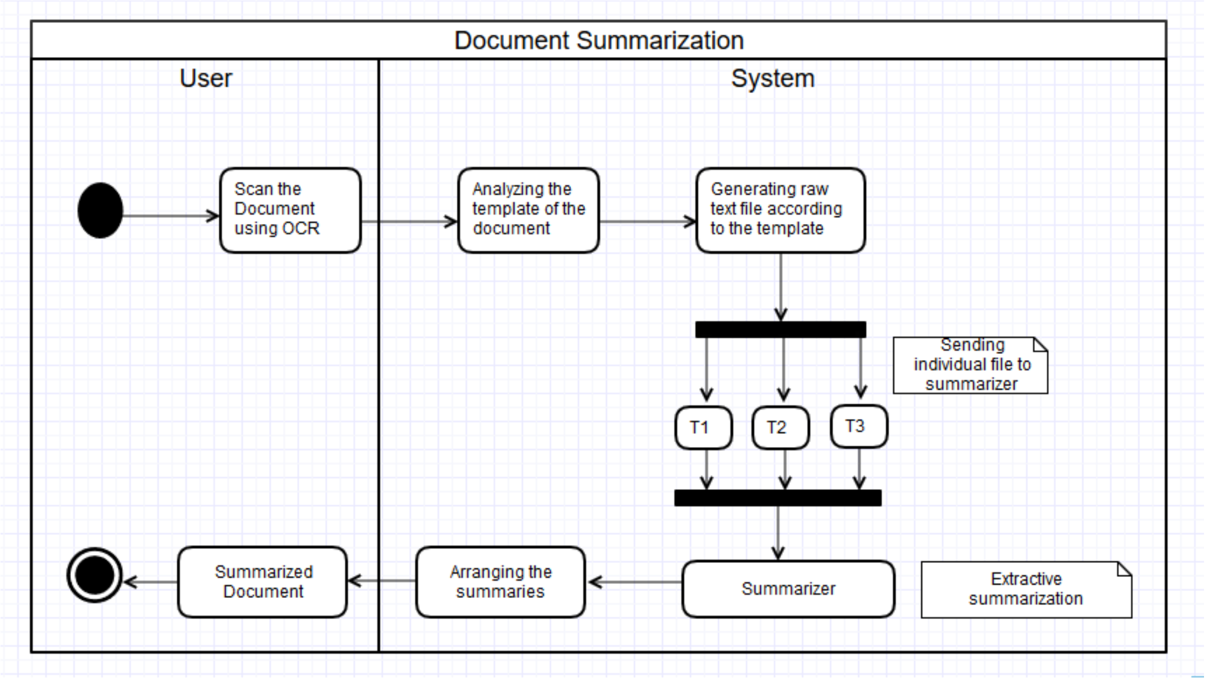
4.2 Flowchart



4.3 Use Case Diagram



4.4 Activity Diagram


Document Summarization
User
System
Scan the Document using OCR
Analyzing the template of the document
Generating raw text file according to the template
Sending individual file to summarizer
T1
T2
T3
Summarizer
Extractive summarization
Arranging the summaries
Summarized Document

CHAPTER 5

SYSTEM DESCRIPTION

We (humans) are generally good at this type of task as it involves first understanding the meaning of the source document and then distilling the meaning and capturing salient details in the new description. As such, the goal of automatically creating summaries of text is to have the resulting summaries as good as those written by humans.

Document Summarization is the process of creating a short and coherent version of a longer document. There is an enormous amount of textual material, and it is only growing every single day. Think of the internet, comprised of web pages, news articles, status updates, blogs and so much more. The data is unstructured and the best that we can do to navigate it is to use search and skim the results. There is a great need to reduce much of this text data to shorter, focused summaries that capture the salient details, both so we can navigate it more effectively as well as check whether the larger documents contain the information that we are looking for.

Our system has taken a different approach to the above problem by performing extractive summarization on IEEE research papers. Extractive text summarization involves the selection of phrases and sentences from the source document to make up the new summary. Techniques involve ranking the relevance of phrases in order to choose only those most relevant to the meaning of the source.

5.1 Preprocessing of Text

Our first task was to identify all the noises in the IEEE papers. Noise is any piece of text which is not relevant to the context of the data and the end-output can be specified as the noise. We identified in most of the IEEE papers has headers, footers, dates, email ID, publishers personal information, references such as [1] and special symbols. Hence preprocessing of the text was needed before sending it to the summarizer. We solved this problem by creating a regular expression for various noises as well as using NLTK package called wordnet and stopwords for identifying Non-English words.

5.2 Splitting of Sections

IEEE research papers have a specific standard format thus to perform information extraction was very much strenuous. At first, we were inputting the whole paper raw text to our summarizer. By doing this we faced major problems, the summary generated was leading to information loss and it was taking a lot of time to output the summary. We considered of splitting the individual sections of IEEE papers in different text files. We accomplished this by doing pattern matching through a regular expression. As IEEE papers have Roman numbering for their titles, we split the sections and subsection on the basis of Roman numbering. After sending split files to summarizers, the summary generated quickly and it didn't lead to loss of information.

5.3 Summary Generation

Our system is generating an extractive summary. The algorithms that we are using to generate summary is Text Rank and Page Rank.

Text Rank – is a graph-based ranking model for text processing which can be used in order to find the most relevant sentences in text and also to find keywords. In order to find the most relevant sentences in text, a graph is constructed where the vertices of the graph represent each sentence in a document and the edges between sentences are based on content overlap, namely by calculating the number of words that 2 sentences have in common. Based on this network of sentences, the sentences are fed into the PageRank algorithm which identifies the most important sentences. When we want to extract a summary of the text, we can now take only the most important sentences.

PageRank – is an algorithm used by Google Search to rank websites in their search engine results. In our summarizer we use sentences instead of links to score. Rank the sentences with underlying assumption that “summary sentences" are similar to most other sentences. The algorithm allows to summarize text by calculating how sentences are related to one another. This is done by looking at overlapping terminology used in sentences in order to set up links between sentences. The resulting sentence network is next plugged into the 'Page rank' algorithm which identifies the most important sentences in your text and ranks them.

5.4 Displaying Summary with Figures and Tables

Our system does not only give the summary of the IEEE paper but we are also extracting all the Figures and Tables that are present in the IEEE paper. The reason behind this is supposing a user encounters a word fig 3 in a summary and following its description, the user must be able to map this information to the actual image of fig 3, this will make the user understand the summary much better.

CHAPTER 6

TESTING

6.1 Test Cases

6.1.1 Noise Removal

Objective

To check if Noise in text is removed

Description

Noise is any piece of text which is not relevant to the context of data

Input

Pass a sample text to the summarizer

Expected Output

The System removes the noise from the text

Sample test

Ryan Heartfie1d*, George Loukas, Diane Gan
Department of Computing and
Information Systems
University of Greenwich, UK
*Email: r.j.heartfie1d@gre.ac.uk
Message Queue Telemetry Transport
(MQTT) [1] and Constrained Application Protocol (COAP) [2]
are most used protocols, and each is designed to be low
overhead and constrained device friendly.

Observed Output

The system removes the noise from the text

Result

Pass

6.1.2 Splitting of text

Objective

To check if IEEE paper is split into text

Description

Splitting the paper topic wise

Input

Pass a sample text to the summarizer

Sample Test

Main.txt

Abstract - The problem of finding an ambulance in an emergency situation is a big issue in India. To solve that we have proposed a system that will reduce the time taken to call an ambulance and also to provide preventive measures that can actually save a life. The System will have OneTouch feature that will initiate ambulance booking and at the side a chatbot which will help the user to get tips and direction to what they must do in a particular emergency.

I. INTRODUCTION

Nowadays thousands of people say 'Good morning' to Alexa every day [6], half a million people have professed their love for it, and more than 250,000 have proposed manage to it. The use of the personalized bot has increased at a drastic pace in the last decade. But the benefit of using such chatbot is just minimum to humans. People spend more time in chatting rather than focusing on the work that they need to be complete. The Chatbot is used to have a conversation with humans with the help of natural language. The chatbot will have to be trained in such a way that it can help us to get a proper guided path. Having chat bot provide a solution to a question is more important than just having a simple greeting 0r pep talk from the bot. As App like Xiaoice in China have average conversation length is 23 conversations per session (CPS) more analytics in getting correct treatment for different cases is possible. In the city like Mumbai, Death rate suffers due to delay in ambulance to reach the patient.

Expected Output

Text1.txt

Abstract - The problem of finding an ambulance in an emergency situation is a big issue in India. To solve that we have proposed a system that will reduce the time taken to call an ambulance and also to provide preventive measures that can actually save a life. The System will have OneTouch feature that will initiate ambulance booking and at the side a chatbot which will help the user to get tips and direction to what they must do in a particular emergency.

Text2.txt

I. INTRODUCTION

Nowadays thousands of people say 'Good morning' to Alexa every day [6], half a million people have professed their love for it, and more than 250,000 have proposed maniage to it. The use of the personalized bot has increased at a drastic pace in the last decade. But the benefit of using such chatbot is just minimum to humans. People spend more time in chatting rather than focusing on the work that they need to be complete. The Chatbot is used to have a conversation with humans with the help of natural language. The chatbot will have to be trained in such a way that it can help us to get a proper guided path. Having chat bot provide a solution to a question is more important than just having a simple greeting 0r pep talk from the bot. As App like Xiaoice in China have average conversation length is 23 conversations per session (CPS) more analytics in getting correct treatment for different cases is possible. In the city like Mumbai, Death rate suffers due to delay in ambulance to reach the patient.

Result

Pass

6.1.3 Summary

Objective

To check whether the summary is properly generated.

Description

Checking whether the summary is covering the important points and making sense.

Input

Pass a sample text of the summarizer

Expected Output

Summary of sample text

Sample Test

Earth is the third planet from the Sun and the only object in the Universe known to harbor life. According to radiometric dating and other sources of evidence, Earth formed over 4.5 billion years ago. Earth's gravity interacts with other objects in space, especially the Sun and the Moon, Earth's only natural satellite. Earth revolves around the Sun in 365.26 days, a period is known as an Earth year. During this time, Earth rotates about its axis about 366.26 times.

Earth's axis of rotation is tilted with respect to its orbital plane, producing seasons on the Earth.The gravitational interaction between the Earth and Moon causes ocean tides, stabilizes the Earth's orientation on its axis, and gradually slows its rotation. Earth is the densest planet in the Solar System and the largest of the four terrestrial planets.

Earth's lithosphere is divided into several rigid tectonic plates that migrate across the surface over periods of many millions of years. About 71% of Earth's surface is covered with water, mostly by oceans. The remaining 29% is land consisting of continents and islands that together have many lakes, rivers and other sources of water that contribute to the hydrosphere.

Observed Output:

Earth is the third planet from the Sun and the only object in the Universe known to harbor life. According to radiometric dating and other sources of evidence, Earth formed over 4.5 billion years ago. Earth's gravity interacts with other objects in space, especially the Sun and the Moon, Earth's only natural satellite. Earth's axis of rotation is tilted with respect to its orbital plane, producing seasons on the Earth. The gravitational interaction between the Earth and Moon causes ocean tides, stabilizes the Earth's orientation on its axis, and gradually slows its rotation. Earth is the densest planet in the Solar System and the largest of the four terrestrial planets. The remaining 29% is land consisting of continents and islands that together have many lakes, rivers and other sources of water that contribute to the hydrosphere.

Result

Pass

6.1.4Order of sentences in summary

Objective

Order of the sentences in the summary

Description

Ordering the sentences of summary as the source paragraph

Input

Pass a sample text of the summarizer

Expected output

Comparing order of the sentences of input and output. Both should be in the same order.

Sample test

As the location of the patient is not known accurately the Dispatcher call up the ambulance which is approximately near to the user location. The dispatcher will also have a doctor on call to ensure the ambulance with necessary equipment is provided. Based on the location given the ambulance will reach the location and return to the nearest hospital that is present. The call taker will take quick information and forward the call to the Dispatcher. The dispatcher then uses GPS technology to locate the nearest ambulance to the patient

Observed Output

The call taker will take quick information and forward the call to the Dispatcher. The dispatcher then uses GPS technology to locate the nearest ambulance to the patient. The dispatcher will also have a doctor on call to ensure the ambulance with necessary equipment is provided. Based on the location given the ambulance will reach the location and return to the nearest hospital that is present. As the location of the patient is not known accurately the Dispatcher call up the ambulance which is approximately near to the user location.

Result

Pass

6.1.5 Extraction of figures and tables

Objective

Extraction of figures and tables

Description

Exracting all figures and tables from the IEEE paper

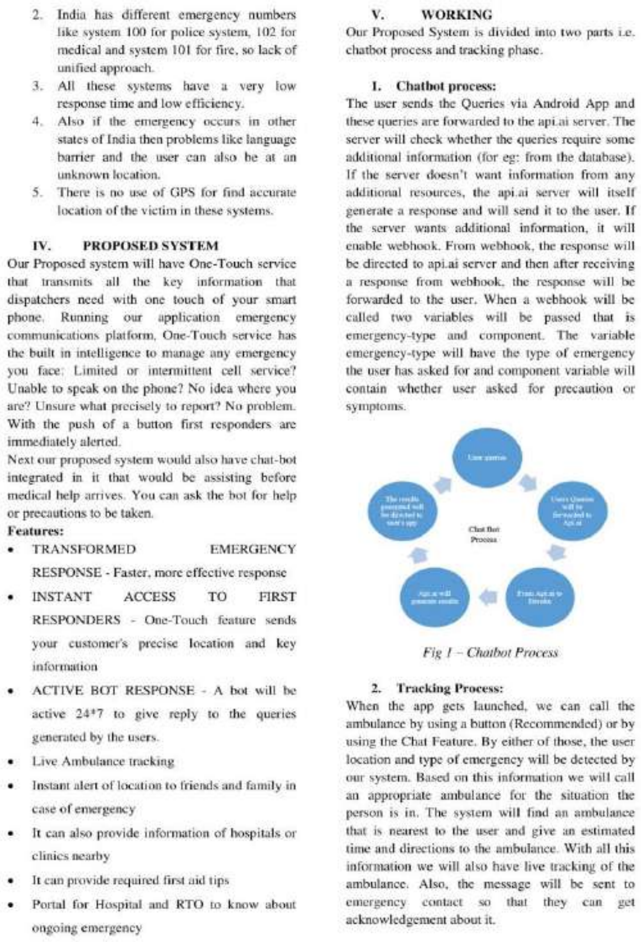
Input

Pass a sample document to the summarizer

Expected output

List of figures and tables on the UI

Sample test



2. India has different emergency numbers like system 100 for police system, 102 for medical and system 101 for fire, so lack of unified approach.
3. All these systems have a very low response time and low efficiency.
4. Also if the emergency occurs in other states of India then problems like language barrier and the user can also be at an unknown location.
5. There is no use of GPS for find accurate location of the victim in these systems.

IV. PROPOSED SYSTEM

Our Proposed system will have One-Touch service that transmits all the key information that dispatchers need with one touch of your smart phone. Running our application emergency communications platform, One-Touch service has the built in intelligence to manage any emergency you face: Limited or intermittent cell service? Unable to speak on the phone? No idea where you are? Unsure what precisely to report? No problem. With the push of a button first responders are immediately alerted.

Next our proposed system would also have chat-bot integrated in it that would be assisting before medical help arrives. You can ask the bot for help or precautions to be taken.

Features:

- TRANSFORMED EMERGENCY RESPONSE - Faster, more effective response
- INSTANT ACCESS TO FIRST RESPONDERS - One-Touch feature sends your customer's precise location and key information
- ACTIVE BOT RESPONSE - A bot will be active 24*7 to give reply to the queries generated by the users.
- Live Ambulance tracking
- Instant alert of location to friends and family in case of emergency
- It can also provide information of hospitals or clinics nearby
- It can provide required first aid tips
- Portal for Hospital and RTO to know about ongoing emergency

V. WORKING

Our Proposed System is divided into two parts i.e. chatbot process and tracking phase.

1. Chatbot process:

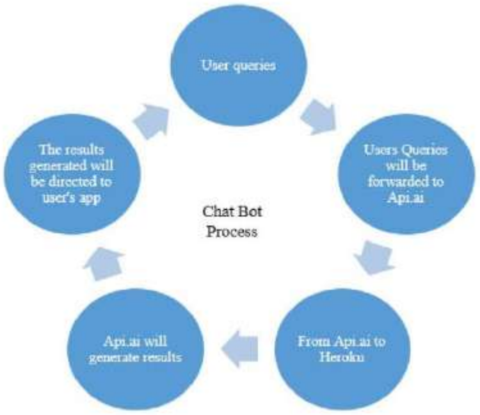
The user sends the Queries via Android App and these queries are forwarded to the api.ai server. The server will check whether the queries require some additional information (for eg: from the database). If the server doesn't want information from any additional resources, the api.ai server will itself generate a response and will send it to the user. If the server wants additional information, it will enable webhook. From webhook, the response will be directed to api.ai server and then after receiving a response from webhook, the response will be forwarded to the user. When a webhook will be called two variables will be passed that is emergency-type and component. The variable emergency-type will have the type of emergency the user has asked for and component variable will contain whether user asked for precaution or symptoms.

Fig 1 – Chatbot Process

2. Tracking Process:

When the app gets launched, we can call the ambulance by using a button (Recommended) or by using the Chat Feature. By either of those, the user location and type of emergency will be detected by our system. Based on this information we will call an appropriate ambulance for the situation the person is in. The system will find an ambulance that is nearest to the user and give an estimated time and directions to the ambulance. With all this information we will also have live tracking of the ambulance. Also, the message will be sent to emergency contact so that they can get acknowledgement about it.

Observed Output


User queries
Users Queries will be forwarded to Api.ai
From Api.ai to Heroku
Api.ai will generate results
The results generated will be directed to user's app
Chat Bot Process

Result

Pass

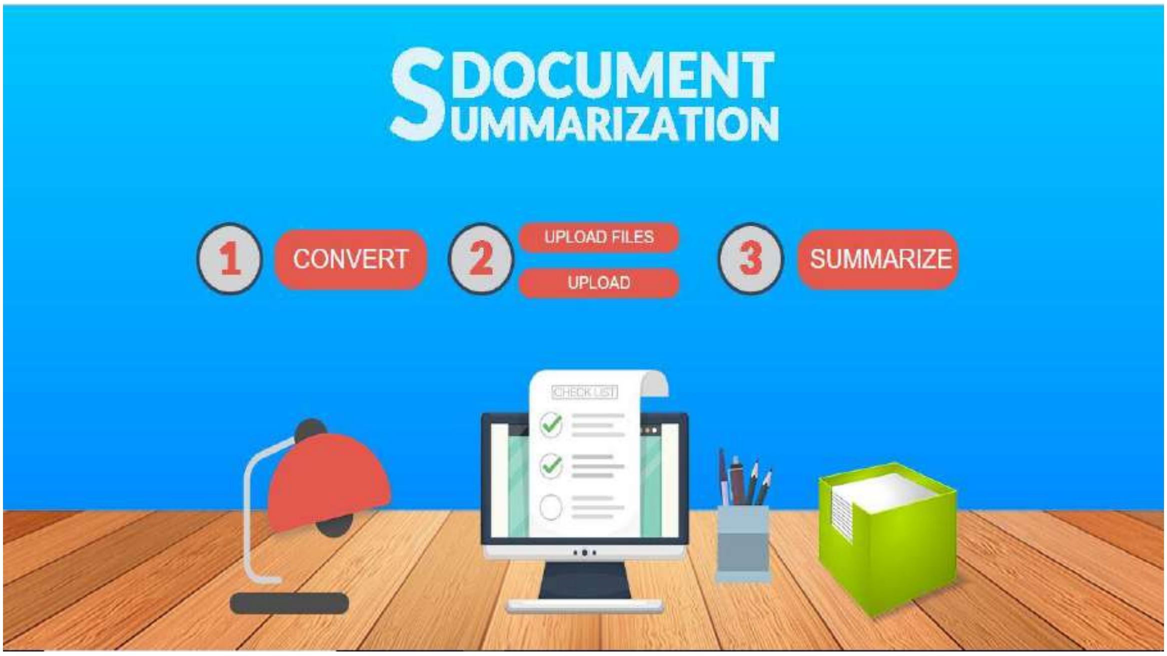
6.2 Results

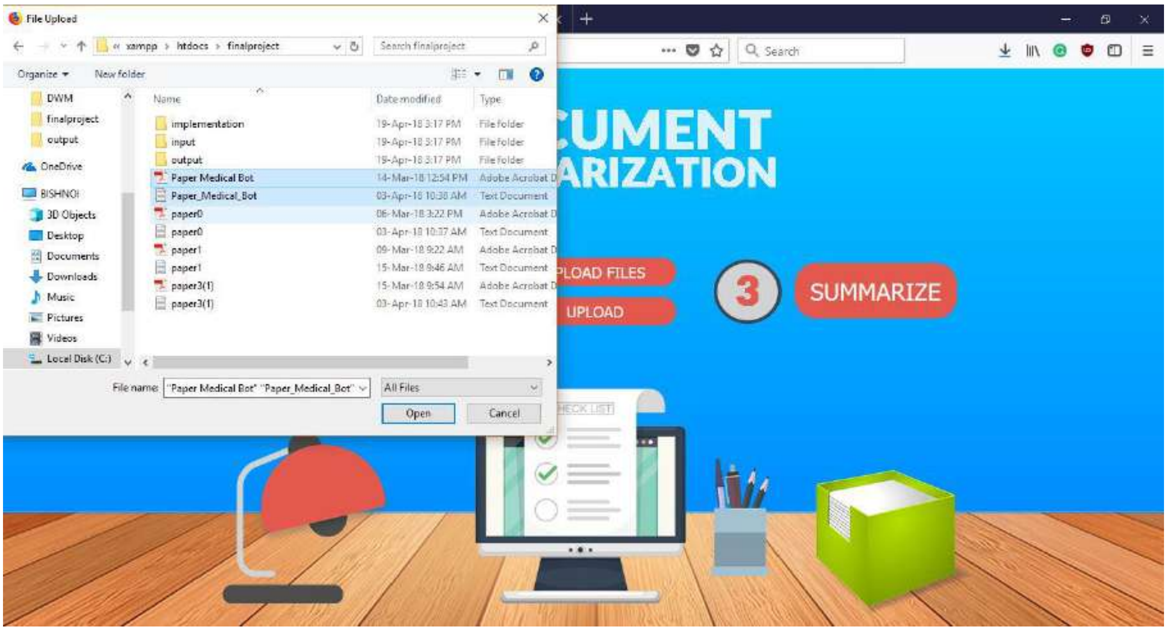
Size of original document	Size of summary	Percentage of compression
2247	1047	47 %

CHAPTER 7

OUTPUT AND SCREENSHOT

7.1 User Interface

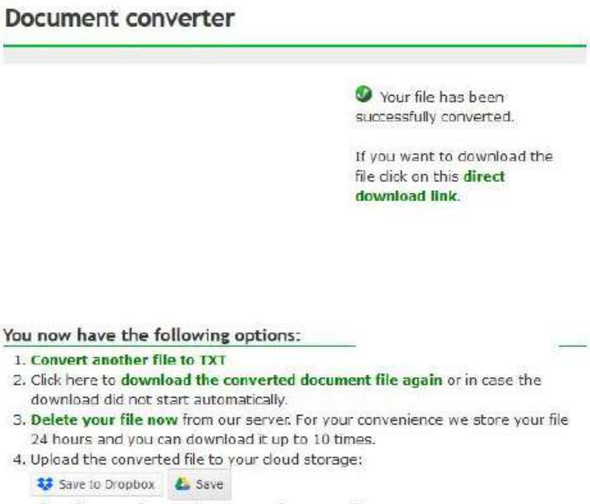




Name	Date modified	Type
implementation	19-Apr-18 3:17 PM	File folder
input	19-Apr-18 3:17 PM	File folder
output	19-Apr-18 3:17 PM	File folder
Paper Medical Bot	14-Mar-18 12:54 PM	Adobe Acrobat D
Paper_Medical_Bot	03-Apr-18 10:30 AM	Text Document
paper0	06-Mar-18 3:22 PM	Adobe Acrobat D
paper0	03-Apr-18 10:37 AM	Text Document
paper1	09-Mar-18 9:22 AM	Adobe Acrobat D
paper1	15-Mar-18 9:46 AM	Text Document
paper3(1)	15-Mar-18 9:54 AM	Adobe Acrobat D
paper3(1)	03-Apr-18 10:43 AM	Text Document

File name: "Paper Medical Bot" "Paper_Medical_Bot"

All Files



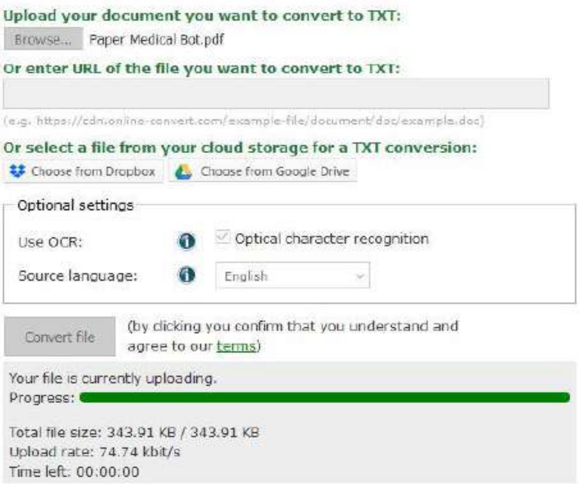
Document converter

✓ Your file has been successfully converted.

If you want to download the file click on this **direct download link**.

You now have the following options:

1. **Convert another file to TXT**
2. Click here to **download the converted document file again** or in case the download did not start automatically.
3. **Delete your file now** from our server. For your convenience we store your file 24 hours and you can download it up to 10 times.
4. Upload the converted file to your cloud storage:



Upload your document you want to convert to TXT:

Browse... Paper Medical Bot.pdf

Or enter URL of the file you want to convert to TXT:

(e.g. https://cdn.online-convert.com/example-file/document/doc/example.doc)

Or select a file from your cloud storage for a TXT conversion:

Optional settings

Use OCR: ☑ Optical character recognition

Source language: English

(by clicking you confirm that you understand and agree to our terms)

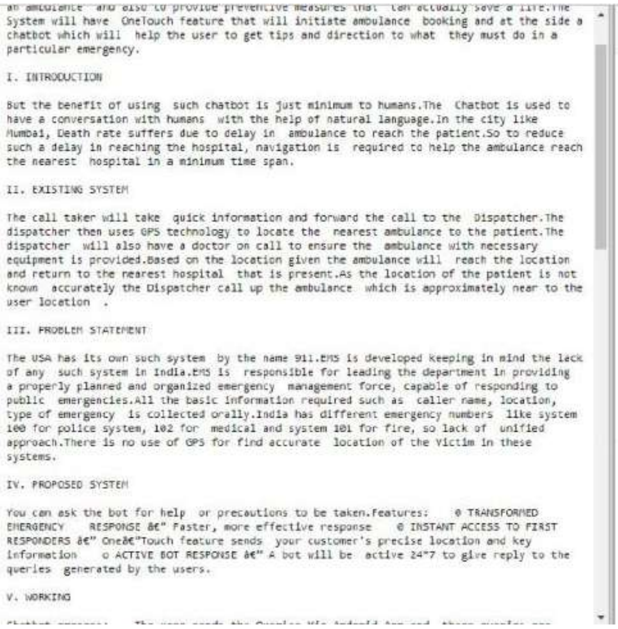
Your file is currently uploading.

Progress:

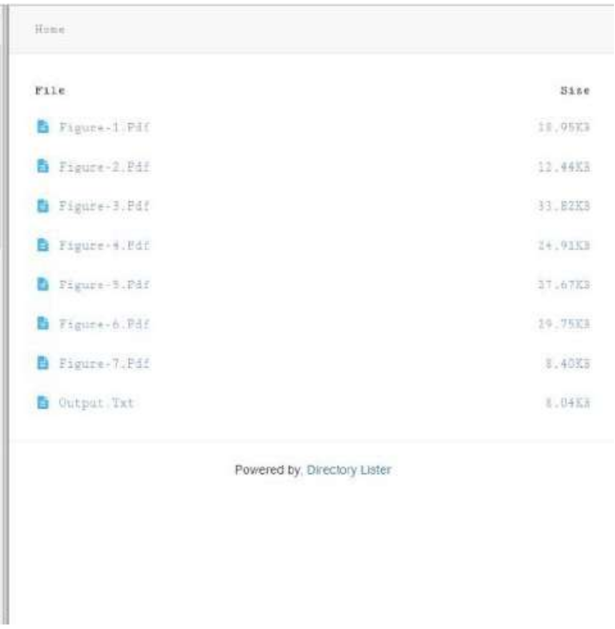
Total file size: 343.91 KB / 343.91 KB

Upload rate: 74.74 kbit/s

Time left: 00:00:00



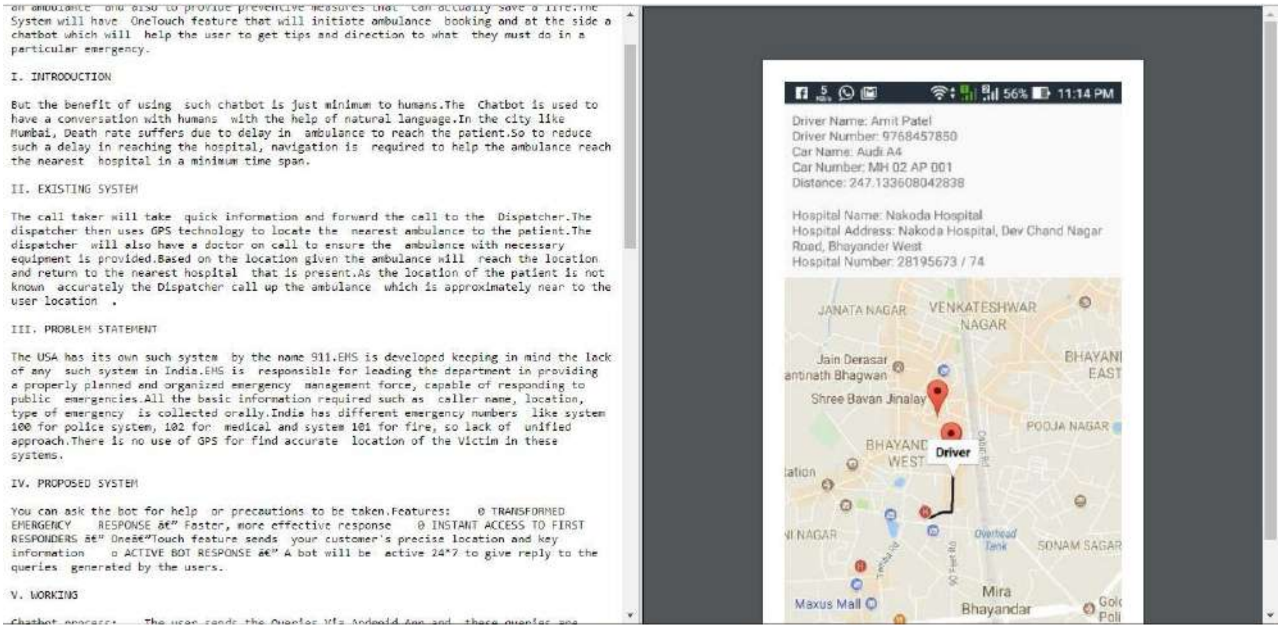
```
an ambulance  and also to provide preventive measures that  can actually save a life.The
System will have  OneTouch feature that will initiate ambulance  booking and at the side a
chatbot which will  help the user to get tips and direction to what  they must do in a
particular emergency.

I. INTRODUCTION

But the benefit of using  such chatbot is just minimum to humans.The  Chatbot is used to
have a conversation with humans  with the help of natural language.In the city like
Mumbai, Death rate suffers due to delay in  ambulance to reach the patient.So to reduce
such a delay in reaching the hospital, navigation is  required to help the ambulance reach
the nearest  hospital in a minimum time span.

II. EXISTING SYSTEM

The call taker will take  quick information and forward the call to the  Dispatcher.The
dispatcher then uses GPS technology to locate the  nearest ambulance to the patient.The
dispatcher  will also have a doctor on call to ensure the  ambulance with necessary
equipment is provided.Based on the location given the ambulance will  reach the location
and return to the nearest hospital  that is present.As the location of the patient is not
known  accurately the Dispatcher call up the ambulance  which is approximately near to the
user location  .

III. PROBLEM STATEMENT

The USA has its own such system  by the name 911.EMS is developed keeping in mind the lack
of any  such system in India.EMS is  responsible for leading the department in providing
a properly planned and organized emergency  management force, capable of responding to
public  emergencies.All the basic information required such as  caller name, location,
type of emergency  is collected orally.India has different emergency numbers  like system
100 for police system, 102 for  medical and system 101 for fire, so lack of  unified
approach.There is no use of GPS for find accurate  location of the Victim in these
systems.

IV. PROPOSED SYSTEM

You can ask the bot for help  or precautions to be taken.Features:    Ø TRANSFORMED
EMERGENCY    RESPONSE â€" Faster, more effective response    Ø INSTANT ACCESS TO FIRST
RESPONDERS â€" Oneâ€"Touch feature sends  your customer's precise location and key
information    o ACTIVE BOT RESPONSE â€" A bot will be  active 24*7 to give reply to the
queries  generated by the users.

V. WORKING
```



Home

File	Size
Figure-1.Pdf	18.95KB
Figure-2.Pdf	12.44KB
Figure-3.Pdf	33.82KB
Figure-4.Pdf	24.91KB
Figure-5.Pdf	37.67KB
Figure-6.Pdf	19.75KB
Figure-7.Pdf	8.40KB
Output.Txt	8.04KB

Powered by: Directory Lister



an ambulance and also to provide preventive measures that can actually save a life.The System will have OneTouch feature that will initiate ambulance booking and at the side a chatbot which will help the user to get tips and direction to what they must do in a particular emergency.

I. INTRODUCTION

But the benefit of using such chatbot is just minimum to humans.The Chatbot is used to have a conversation with humans with the help of natural language.In the city like Mumbai, Death rate suffers due to delay in ambulance to reach the patient.So to reduce such a delay in reaching the hospital, navigation is required to help the ambulance reach the nearest hospital in a minimum time span.

II. EXISTING SYSTEM

The call taker will take quick information and forward the call to the Dispatcher.The dispatcher then uses GPS technology to locate the nearest ambulance to the patient.The dispatcher will also have a doctor on call to ensure the ambulance with necessary equipment is provided.Based on the location given the ambulance will reach the location and return to the nearest hospital that is present.As the location of the patient is not known accurately the Dispatcher call up the ambulance which is approximately near to the user location .

III. PROBLEM STATEMENT

The USA has its own such system by the name 911.EMS is developed keeping in mind the lack of any such system in India.EMS is responsible for leading the department in providing a properly planned and organized emergency management force, capable of responding to public emergencies.All the basic information required such as caller name, location, type of emergency is collected orally.India has different emergency numbers like system 100 for police system, 102 for medical and system 101 for fire, so lack of unified approach.There is no use of GPS for find accurate location of the Victim in these systems.

IV. PROPOSED SYSTEM

You can ask the bot for help or precautions to be taken.Features: Θ TRANSFORMED EMERGENCY RESPONSE â€” Faster, more effective response Θ INSTANT ACCESS TO FIRST RESPONDERS â€" Oneâ€"Touch feature sends your customer's precise location and key information ◽ ACTIVE BOT RESPONSE â€" A bot will be active 24*7 to give reply to the queries generated by the users.

V. WORKING

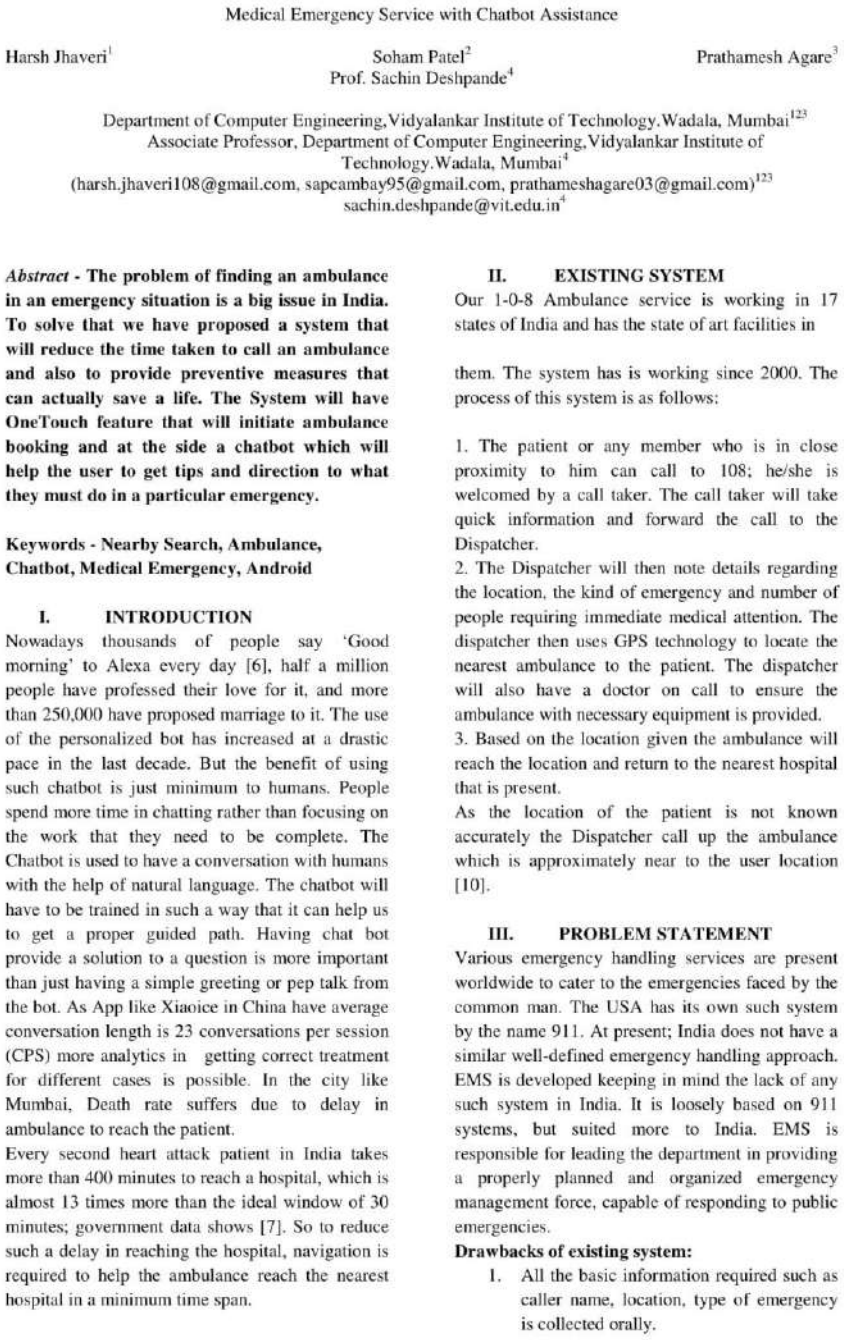
Chatbot process: The user sends the Queries Via Android App and these queries are

Driver Name: Amit Patel
Driver Number: 9768457850
Car Name: Audi A4
Car Number: MH 02 AP 001
Distance: 247.133608042838

Hospital Name: Nakoda Hospital
Hospital Address: Nakoda Hospital, Dev Chand Nagar Road, Bhayander West
Hospital Number: 28195673 / 74

7.2 Example of Document Summarization

This the sample paper we have given as input to our summarizer.



Medical Emergency Service with Chatbot Assistance

Harsh Jhaveri[1] Soham Patel[2] Prathamesh Agare[3]
Prof. Sachin Deshpande[4]

Department of Computer Engineering,Vidyalankar Institute of Technology.Wadala, Mumbai[123]
Associate Professor, Department of Computer Engineering,Vidyalankar Institute of Technology.Wadala, Mumbai[4]
(harsh.jhaveri108@gmail.com, sapcambay95@gmail.com, prathameshagare03@gmail.com)[123]
sachin.deshpande@vit.edu.in[4]

***Abstract* - The problem of finding an ambulance in an emergency situation is a big issue in India. To solve that we have proposed a system that will reduce the time taken to call an ambulance and also to provide preventive measures that can actually save a life. The System will have OneTouch feature that will initiate ambulance booking and at the side a chatbot which will help the user to get tips and direction to what they must do in a particular emergency.**

Keywords - Nearby Search, Ambulance, Chatbot, Medical Emergency, Android

I. INTRODUCTION

Nowadays thousands of people say 'Good morning' to Alexa every day [6], half a million people have professed their love for it, and more than 250,000 have proposed marriage to it. The use of the personalized bot has increased at a drastic pace in the last decade. But the benefit of using such chatbot is just minimum to humans. People spend more time in chatting rather than focusing on the work that they need to be complete. The Chatbot is used to have a conversation with humans with the help of natural language. The chatbot will have to be trained in such a way that it can help us to get a proper guided path. Having chat bot provide a solution to a question is more important than just having a simple greeting or pep talk from the bot. As App like Xiaoice in China have average conversation length is 23 conversations per session (CPS) more analytics in getting correct treatment for different cases is possible. In the city like Mumbai, Death rate suffers due to delay in ambulance to reach the patient.

Every second heart attack patient in India takes more than 400 minutes to reach a hospital, which is almost 13 times more than the ideal window of 30 minutes; government data shows [7]. So to reduce such a delay in reaching the hospital, navigation is required to help the ambulance reach the nearest hospital in a minimum time span.

II. EXISTING SYSTEM

Our 1-0-8 Ambulance service is working in 17 states of India and has the state of art facilities in them. The system has is working since 2000. The process of this system is as follows:

1. The patient or any member who is in close proximity to him can call to 108; he/she is welcomed by a call taker. The call taker will take quick information and forward the call to the Dispatcher.
2. The Dispatcher will then note details regarding the location, the kind of emergency and number of people requiring immediate medical attention. The dispatcher then uses GPS technology to locate the nearest ambulance to the patient. The dispatcher will also have a doctor on call to ensure the ambulance with necessary equipment is provided.
3. Based on the location given the ambulance will reach the location and return to the nearest hospital that is present.

As the location of the patient is not known accurately the Dispatcher call up the ambulance which is approximately near to the user location [10].

III. PROBLEM STATEMENT

Various emergency handling services are present worldwide to cater to the emergencies faced by the common man. The USA has its own such system by the name 911. At present; India does not have a similar well-defined emergency handling approach. EMS is developed keeping in mind the lack of any such system in India. It is loosely based on 911 systems, but suited more to India. EMS is responsible for leading the department in providing a properly planned and organized emergency management force, capable of responding to public emergencies.

Drawbacks of existing system:

1. All the basic information required such as caller name, location, type of emergency is collected orally.

2. India has different emergency numbers like system 100 for police system, 102 for medical and system 101 for fire, so lack of unified approach.
3. All these systems have a very low response time and low efficiency.
4. Also if the emergency occurs in other states of India then problems like language barrier and the user can also be at an unknown location.
5. There is no use of GPS for find accurate location of the victim in these systems.

IV. PROPOSED SYSTEM

Our Proposed system will have One-Touch service that transmits all the key information that dispatchers need with one touch of your smart phone. Running our application emergency communications platform, One-Touch service has the built in intelligence to manage any emergency you face: Limited or intermittent cell service? Unable to speak on the phone? No idea where you are? Unsure what precisely to report? No problem. With the push of a button first responders are immediately alerted.

Next our proposed system would also have chat-bot integrated in it that would be assisting before medical help arrives. You can ask the bot for help or precautions to be taken.

Features:

- TRANSFORMED EMERGENCY RESPONSE - Faster, more effective response
- INSTANT ACCESS TO FIRST RESPONDERS - One-Touch feature sends your customer's precise location and key information
- ACTIVE BOT RESPONSE - A bot will be active 24*7 to give reply to the queries generated by the users.
- Live Ambulance tracking
- Instant alert of location to friends and family in case of emergency
- It can also provide information of hospitals or clinics nearby
- It can provide required first aid tips
- Portal for Hospital and RTO to know about ongoing emergency

V. WORKING

Our Proposed System is divided into two parts i.e. chatbot process and tracking phase.

1. Chatbot process:

The user sends the Queries via Android App and these queries are forwarded to the api.ai server. The server will check whether the queries require some additional information (for eg: from the database). If the server doesn't want information from any additional resources, the api.ai server will itself generate a response and will send it to the user. If the server wants additional information, it will enable webhook. From webhook, the response will be directed to api.ai server and then after receiving a response from webhook, the response will be forwarded to the user. When a webhook will be called two variables will be passed that is emergency-type and component. The variable emergency-type will have the type of emergency the user has asked for and component variable will contain whether user asked for precaution or symptoms.

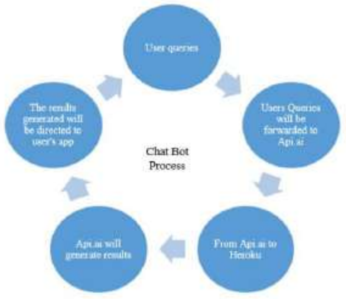

User queries
Users Queries will be forwarded to Api.ai
From Api.ai to Heroku
Api.ai will generate results
The results generated will be directed to user's app
Chat Bot Process

Fig 1 – Chatbot Process

2. Tracking Process:

When the app gets launched, we can call the ambulance by using a button (Recommended) or by using the Chat Feature. By either of those, the user location and type of emergency will be detected by our system. Based on this information we will call an appropriate ambulance for the situation the person is in. The system will find an ambulance that is nearest to the user and give an estimated time and directions to the ambulance. With all this information we will also have live tracking of the ambulance. Also, the message will be sent to emergency contact so that they can get acknowledgement about it.

VI. IMPLEMENTATION DETAILS

As Moore's Law states that the number of transistors used in a densely integrated circuit doubles every two years. The number of the smartphone also has increased and predicted to be 500 million in India by 2018. So we have developed an app for android which tightly coupled with firebase server for an instant response from the server.

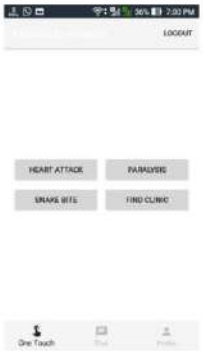


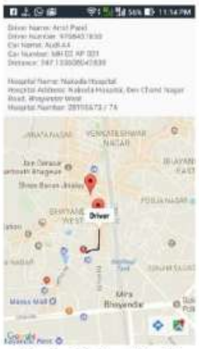
Fig 2 - *Emergency Buttons*

On a click of a button as shown in figure 2, the app will send the location and emergency type to our server which starts a search for an ambulance in the nearby location. Until a particular condition to find an ambulance is not achieved we keep on searching the ambulance with the help of Geofire.

Pseudo code:(Ambulance Searching)

```
begin
    radius=0
    fetch(latitude ,longitude)
    whlie (Driver == 0 and radius < 5)
        begin
            radius is incremented by 1
            findDriver(latitude,longitude,radius)
        end
    if found select the nearest Driver
        initiate navigation
    else show lcation of nearest hospital
end
```

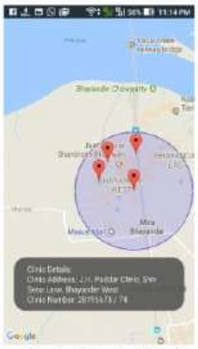
Geofire uses geohash value of the longitude and latitude and stores it in a string format or geohash format. This is very valuable as it is quick and reduces the cost of doing operations on the location in the database [4]. The storage also becomes efficient as the two-dimensional location value is converted to the one-dimensional hash value which is cheap to store [5].



Driver Name: Arsil Patel
Driver Number: 9758451830
Car Name: Audi A4
Car Number: MH 02 AP 001
Distance: 047133808047838

Hospital Name: Nakoda Hospital
Hospital Address: Nakoda Hospital, Dev Chand Nagar Road, Bhayander West
Hospital Number: 28155673 / 74

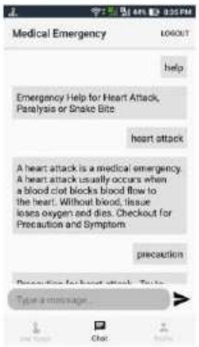
Fig 3 - *Driver Tracking*



Clinic Details
Clinic Address: J.H. Poddar Clinic, Shiv Sena Lane, Bhayander West
Clinic Number: 28155673 / 74

Fig 4 - *Nearby Clinic Search*

As shown in the above figure 3, when a driver is assigned to the user it will plot the location of the driver on the map and gives the user continuous feedback about the location. With this also gives other contact details in case the user needs to interact with the driver. If the user wants to get a location of the nearby clinic then can also be found on the app as shown in figure 4.



Medical Emergency

help

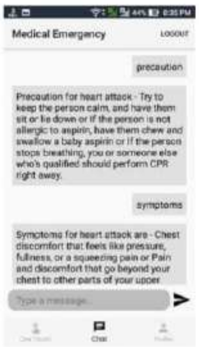
Emergency Help for Heart Attack, Paralysis or Snake Bite

heart attack

A heart attack is a medical emergency. A heart attack usually occurs when a blood clot blocks blood flow to the heart. Without blood, tissue loses oxygen and dies. Checkout for Precaution and Symptom

precaution

Fig 5



Medical Emergency

precaution

Precaution for heart attack - Try to keep the person calm, and have them sit or lie down or if the person is not allergic to aspirin, have them chew and swallow a baby aspirin or if the person stops breathing, you or someone else who's qualified should perform CPR right away.

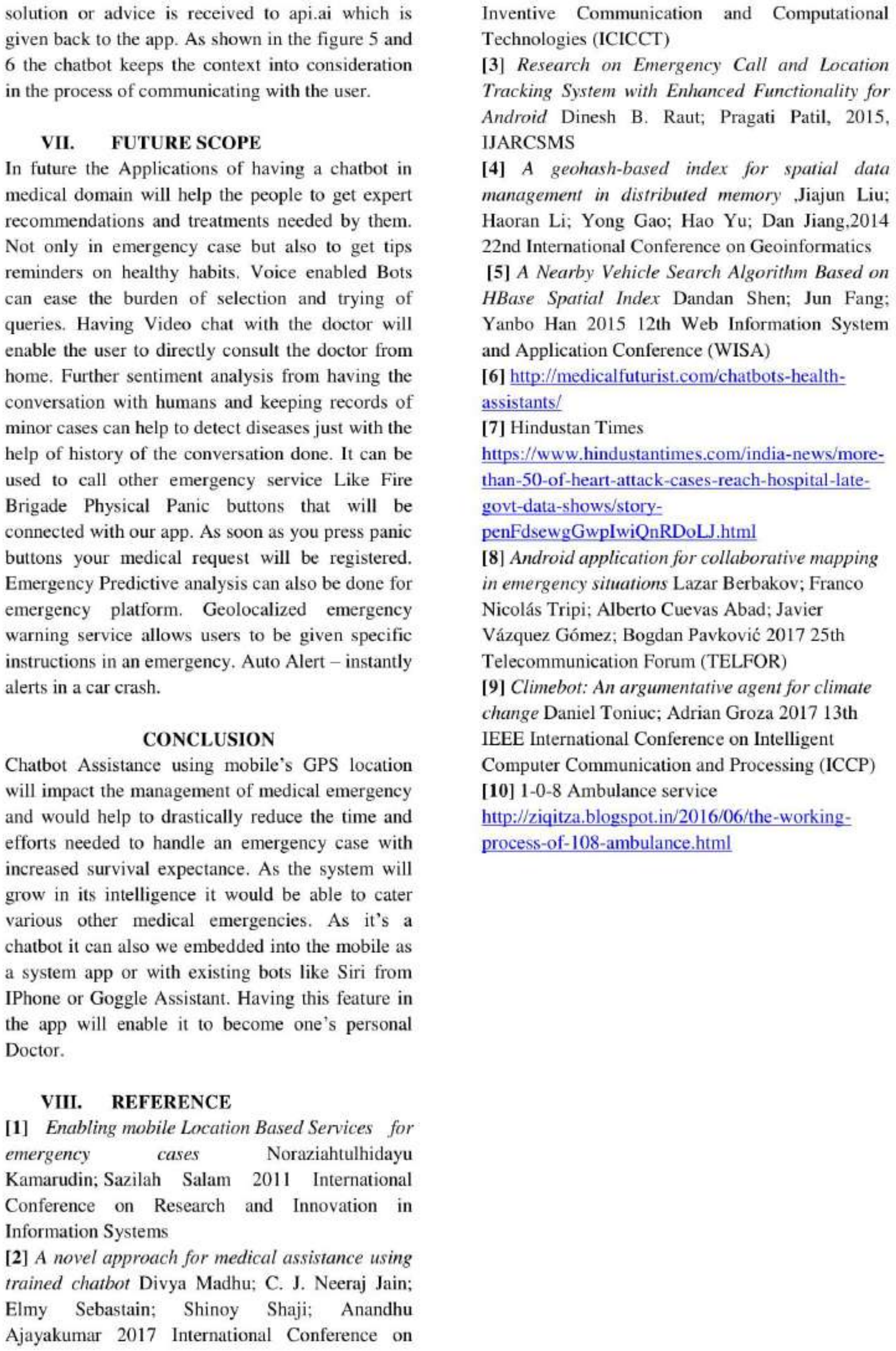
symptoms

Symptoms for heart attack are - Chest discomfort that feels like pressure, fullness, or a squeezing pain or Pain and discomfort that go beyond your chest to other parts of your upper

Fig 6

Interaction with Chatbot on Heart Attack

The Chatbot plays a critical role in the time period between the request for ambulance and ambulance arrival. The chatbot is directly linked to an api.ai server which is intelligent enough to classify the emergency and other questions asked by the user. The Natural language processing is done on that server and based on the query it is classified to a particular intent which calls a python function stored on our Heroku server. From that server, a



solution or advice is received to api.ai which is given back to the app. As shown in the figure 5 and 6 the chatbot keeps the context into consideration in the process of communicating with the user.

VII. FUTURE SCOPE

In future the Applications of having a chatbot in medical domain will help the people to get expert recommendations and treatments needed by them. Not only in emergency case but also to get tips reminders on healthy habits. Voice enabled Bots can ease the burden of selection and trying of queries. Having Video chat with the doctor will enable the user to directly consult the doctor from home. Further sentiment analysis from having the conversation with humans and keeping records of minor cases can help to detect diseases just with the help of history of the conversation done. It can be used to call other emergency service Like Fire Brigade Physical Panic buttons that will be connected with our app. As soon as you press panic buttons your medical request will be registered. Emergency Predictive analysis can also be done for emergency platform. Geolocalized emergency warning service allows users to be given specific instructions in an emergency. Auto Alert – instantly alerts in a car crash.

CONCLUSION

Chatbot Assistance using mobile's GPS location will impact the management of medical emergency and would help to drastically reduce the time and efforts needed to handle an emergency case with increased survival expectance. As the system will grow in its intelligence it would be able to cater various other medical emergencies. As it's a chatbot it can also we embedded into the mobile as a system app or with existing bots like Siri from IPhone or Goggle Assistant. Having this feature in the app will enable it to become one's personal Doctor.

VIII. REFERENCE

[1] *Enabling mobile Location Based Services for emergency cases* Noraziahtulhidayu Kamarudin; Sazilah Salam 2011 International Conference on Research and Innovation in Information Systems

[2] *A novel approach for medical assistance using trained chatbot* Divya Madhu; C. J. Neeraj Jain; Elmy Sebastain; Shinoy Shaji; Anandhu Ajayakumar 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT)

[3] *Research on Emergency Call and Location Tracking System with Enhanced Functionality for Android* Dinesh B. Raut; Pragati Patil, 2015, IJARCSMS

[4] *A geohash-based index for spatial data management in distributed memory* ,Jiajun Liu; Haoran Li; Yong Gao; Hao Yu; Dan Jiang,2014 22nd International Conference on Geoinformatics

[5] *A Nearby Vehicle Search Algorithm Based on HBase Spatial Index* Dandan Shen; Jun Fang; Yanbo Han 2015 12th Web Information System and Application Conference (WISA)

[6] http://medicalfuturist.com/chatbots-health-assistants/

[7] Hindustan Times https://www.hindustantimes.com/india-news/more-than-50-of-heart-attack-cases-reach-hospital-late-govt-data-shows/story-penFdsewgGwpIwiQnRDoLJ.html

[8] *Android application for collaborative mapping in emergency situations* Lazar Berbakov; Franco Nicolás Tripi; Alberto Cuevas Abad; Javier Vázquez Gómez; Bogdan Pavković 2017 25th Telecommunication Forum (TELFOR)

[9] *Climebot: An argumentative agent for climate change* Daniel Toniuc; Adrian Groza 2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)

[10] 1-0-8 Ambulance service http://ziqitza.blogspot.in/2016/06/the-working-process-of-108-ambulance.html

Summary of the above paper

Medical Emergency Service with Chatbot Assistance

Abstract—The problem of finding an ambulance in an emergency situation is a big issue in India. To solve that we have proposed a system that will reduce the time taken to call an ambulance and also to provide preventive measures that can actually save a life. The System will have OneTouch feature that will initiate ambulance booking and at the side a chatbot which will help the user to get tips and direction to what they must do in a particular emergency.

I. INTRODUCTION

But the benefit of using such chatbot is just minimum to humans. The Chatbot is used to have a conversation with humans with the help of natural language. In the city like Mumbai, Death rate suffers due to delay in an ambulance to reach the patient. So to reduce such a delay in reaching the hospital, navigation is required to help the ambulance reach the nearest hospital in a minimum time span.

II. EXISTING SYSTEM

The call taker will take quick information and forward the call to the Dispatcher. The dispatcher then uses GPS technology to locate the nearest ambulance to the patient. The dispatcher will also have a doctor on call to ensure the ambulance with necessary equipment is provided. Based on the location given the ambulance will reach the location and return to the nearest hospital that is present. As the location of the patient is not known accurately the Dispatcher call up the ambulance which is approximately near to the user location.

III. PROBLEM STATEMENT

The USA has its own such system by the name 911.EMS is developed keeping in mind the lack of any such system in India. EMS is responsible for leading the department in providing a properly planned and organized emergency management force, capable of responding to public emergencies. All the basic information required such as caller name, location, type of emergency is collected orally. India has different emergency numbers like system 100 for police system, 102 for medical and system 101 for fire, so lack of unified approach. There is no use of GPS for find accurate location of the Victim in these systems.

IV. PROPOSED SYSTEM

You can ask the bot for help or precautions to be taken. Features: TRANSFORMED EMERGENCY RESPONSE — Faster, more effective response INSTANT ACCESS TO FIRST RESPONDERS — One—Touch feature sends your customer's precise location and key information o ACTIVE BOT RESPONSE — A bot will be active 24*7 to give the reply to the queries generated by the users.

V. WORKING

Chatbot process: The user sends the Queries Via Android App and these queries are forwarded to the api.ai server. The server will check whether the queries require some additional information (for eg: from the database).If the server doesn’t want information from any additional resources, the api.ai server will itself generate a response and will send it to the user. From webhook, the response will be directed to api.ai server and then after receiving a response from webhook, the response will be forwarded to the user. When a webhook will be called two variables will be passed that is emergency-type and component. The variable emergency-type will have the type of emergency the user has asked for and component variable will contain whether user asked for precaution or symptoms .By either of those, the user location and type of emergency will be detected by our system. The system will find an ambulance that is nearest to the user and give an estimated time and directions to the ambulance.

VI. IMPLEMENTATION DETAILS

Fig 2 — Emergency Buttons On a click of a button as shown in figure 2, the app will send the location and emergency type to our server which starts a search for an ambulance in the nearby location. Until a particular condition to find an ambulance is not achieved we keep on searching the ambulance with the help of Geofire. This is very valuable as it is quick and reduces the cost of doing operations on the location in the database. The storage also becomes efficient as the two—dimensional location value is converted to the one— a dimensional hash value which is cheap to store. Fig 3 — Driver Tracking Fig 4 — Nearby Clinic Search

As shown in the above figure 3, when a driver is assigned to the user it will plot the location of the driver on the map and gives the user continuous feedback about the location. If the user wants to get a ocation of the nearby clinic then can also be found on the app as shown in figure 4.A heart attack usually occurs when a blood clot blocks blood flow to the heart. The Chatbot plays a critical role in the time period between the request for ambulance and ambulance arrival. The chatbot is directly linked to an api.ai server which is intelligent enough to classify the emergency and other questions asked by the user. The Natural language processing is done on that server and based on the query it is classified to a particular intent which calls a python function stored on our Heroku server. From that server, a solution or advice is received to api.ai which is given back to the app. As shown in figure 5 and 6 the chatbot keeps the context into consideration in the process of communicating with the user.

VII. FUTURE SCOPE

In future, the Applications of having a chatbot in the medical domain will help the people to get expert recommendations and treatments needed by them. Further sentiment analysis from having the conversation with humans and keeping records of minor cases can help to detect diseases just with the help of history of the conversion done. It can be used to call another emergency service Like Fire Brigade Physical Panic buttons that will be connected to our app. Emergency Predictive analysis can also be done for the emergency platform.

VIII. CONCLUSION

Chatbot Assistance using mobile’s GPS location will impact the management of medical emergency and would help to drastically reduce the time and efforts needed to handle an emergency case with increased survival expectancy. As the system will grow in its intelligence it would be able to cater various other medical emergencies. As it’s a chatbot it can also we embedded into the mobile as a system app or with existing bots like Siri from iPhone or Google Assistant. Having this feature in the app will enable it to become ones, personal Doctor.

CHAPTER 8

CONCLUSION &
FUTURE SCOPE

Conclusion

The increasing growth of the Internet has made a huge amount of information available. It is difficult for humans to summarize large amounts of text. Thus, there is an immense need for automatic summarization tools in this age of information overload.

Document summarizer will help students and other respective users to quickly go through IEEE papers. Using the above-mentioned methodologies, we compressed the whole document to 40% of the whole document along with the images and tables. This system can be implemented for other types of documents by making variations in the templates. Thus, generating a summary with all the supporting data.

Future Scope

- **Multi-document summarization**

  Multi-document Summarization is an automatic summarization approach which uses extractive summarization methodology to extract information from multiple documents written about the same topic. The resulting summary enables the users to quickly familiarize themselves with information contained in a large cluster of documents. It creates summary reports that are both concise and compressed. While the goal of a brief summary is to simplify information search and cut the time by pointing to the most relevant source documents, comprehensive multi-document summary should itself contain the required information, hence limiting the need for accessing original files to cases when refinement is required. Automatic summaries present information extracted from multiple sources algorithmically, without any editorial touch or subjective human intervention, thus making it completely unbiased. The multi-document summarization task is more complex than summarizing a single document, even a long one. The difficulty arises from thematic diversity within a large set of documents. A good summarization technology aims to combine the main themes with completeness, readability, and concision. An ideal multi-document summarization system not only shortens the source texts, but also presents information organized around the key aspects to represent diverse views.

- **Abstractive summarization**

  Extraction techniques merely copy the information deemed most important by the system to the summary (for example, key clauses, sentences or paragraphs), while abstraction involves paraphrasing sections of the source document. In general, abstraction can condense a text more strongly than extraction, but the programs that can do this are harder to develop as they require use of natural language generation technology, which itself is a growing field. Some work has been done in abstractive summarization (creating an abstract synopsis like that of a human), the majority of summarization systems are extractive (selecting a subset of sentences to place in a summary). Abstractive summarization techniques can again be classified two categories- structured based and semantic based methods.

Structured Based Approach

Structured based approach encodes most important information from the document(s) through cognitive schemas [6] such as templates, extraction rules and other structures such as tree, ontology, lead and body phrase structure. Different methods used this approach are discussed as follows.

Tree based method

This technique uses a dependency tree to represent the text/contents of a document. Different algorithms are used for content selection for summary e.g. theme intersection algorithm or algorithm that uses local alignment across pair of parsed sentences. The technique uses either a language generator or an algorithm for generation of summary.

Template based method

This technique uses a template to represent a whole document. Linguistic patterns or extraction rules are matched to identify text snippets that will be mapped into template slots.

Ontology based method

News summarization is done by news agent based on fuzzy ontology. The benefit of this approach is that it exploits fuzzy ontology to handle uncertain data that simple domain ontology cannot.

Lead and body phrase method

This method is based on the operations of phrases (insertion and substitution) that have same syntactic

head chunk in the lead and body sentences in order to rewrite the lead sentence.

Semantic Based Approach

In Semantic based method, semantic representation of document(s) is used to feed into natural language

generation (NLG) system. This method focuses on identifying noun phrases and verb phrases by processing linguistic data. Different methods using this approach are discussed here.

Multimodal semantic model

In this method, a semantic model, which captures concepts and relationship among concepts, is built to represent the contents (text and images) of multimodal documents. The important concepts are rated based on some measure and finally the selected concepts are expressed as sentences to form summary.

Information item based method

In this method**,** the contents of summary are generated from abstract representation of source documents, rather than from sentences of source documents. The abstract representation is Information Item, which is the smallest element of coherent information in a text.

Semantic Graph Based Method

This method aims to summarize a document by creating a semantic graph called Rich Semantic Graph (RSG) for the original document, reducing the generated semantic graph, and then generating the final abstractive summary from the reduced semantic graph.

CHAPTER 9

REFERENCES

Research Papers

[1] Wesley T. Chuang , and Jihoon Yang, "Extracting Sentence Segments for Text Summarization: A Machine Learning Approach" Computer Science Department, UCLA, Los Angeles, CA 90095, USA and HRL Laboratories, LLC, 3011 Malibu Canyon Road, Malibu, CA 90265, USA

[2] SurajitKarmakar, Tanvi Lad, HitenChothani, "A Review Paper on Extractive Techniques of Text Summarization" International Research Journal of Computer Science (IRJCS) Issue 1, Volume 2 (January 2015)

[3] A TEMPLATE BASED ALGORITHM FOR AUTOMATIC SUMMARIZATION AND DIALOGUE MANAGEMENT FOR TEXT DOCUMENTS Prashant G. Desai1 , Sarojadevi H2 , Niranjan N. Chiplunkar3 1 Lecturer, Department of Computer Science & Engineering, N.R.A.M.P., Nitte, Karnataka, India 2Professor & Head, Department of Computer Science, N.M.A.M.I.T., Nitte, Karnataka, India 3Principal, N.M.A.M.I.T., Nitte, Karnataka, India

[4] In Proceedings of the ACL Workshop on Text Summarization, Barcelona, July 2004. "Template-Filtered Headline Summarization" Liang Zhou and Eduard Hovy USC Information Sciences Institute 4676 Admiralty Way Marina del Rey, CA 90292-6695

[5] Corpus-trained text generation for summarization Min-Yen Kan and Kathleen R. McKeown Department of Computer Science Columbia University New York, NY 10027, USA

[6] Automatic Summarization of News Articles using TextRank 1Ajinkya Zadbuke, 2Sahil Pimenta, 3Deepen Padwal, 4Varsha Wangikar 1, 2, 3 IT, University of Mumbai, Maharashtra, India 4 Asst. Professor, KCCEMSR, Maharashtra, India

[7] A Frequent Term and Semantic Similarity based Single Document Text Summarization Algorithm Naresh Kumar Nagwani Assistant Professor, Department of CS&E, National Institute of Technology Raipur Dr. Shrish Verma Associate Professor, Department of E&TC, National Institute of Technology Raipur

Links:

http://nlpforhackers.io/textrank-text-summarization/

https://www.geeksforgeeks.org/page-rank-algorithm-implementation/

https://www.nltk.org/

https://janav.wordpress.com/2013/10/27/tf-idf-and-cosine-similarity/

https://machinelearningmastery.com/gentle-introduction-text-summarization/

https://www.cs.cmu.edu/~jgc/publication/MultiDocument_Summarization_Sentence_ANLP_2000.pdf

https://www.analyticsvidhya.com/blog/2015/04/pagerank-explained-simple/

CHAPTER 10

CERTIFICATE