Int. J. Advanced Networking and Applications
Volume: 11 Issue: 06 Pages: 4458-4467 (2020) ISSN: 0975-0290

Content Modelling Intelligence System Based on Automatic Text Summarization

Sanjan S Malagi, Rachana Radhakrishnan, Monisha R, Keerthana S
Department of Information Science & Engineering, JSS ACADEMY OF TECHNICAL EDUCATION, Bangalore-560060
Email: sanjansmalagi38@gmail.com, rachu.radhi@gmail.com, monishamonu1998@gmail.com, keeru1498@gmail.com
Dr D V Ashoka
Professor, Department of Information Science & Engineering, JSS ACADEMY OF TECHNICAL EDUCATION, Bangalore-560060
Email: dr.dvashoka@gmail.com

ABSTRACT

Nowadays, within the period of having huge information, literary information is rapidly developing and is accessible in numerous diverse languages. Often due to time limitations, we are not able to devour all the information that is accessible. With the fast-paced world, it is troublesome to peruse all the textual content. Therefore, the necessity for content summarization comes to the spotlight. It is in this manner we are able to summarize the content so that it gets easier to ingest the data, keeping up the substance, and understanding the data. A few content summarization approaches have been presented in the past for a long time for English and some other European languages but there are startlingly few methods that can be found for the local languages of India. This paper presents a study of extractive content summarization methods for multiple Indian and international languages like Hindi, Kannada, Telugu, Marathi, German, French, etc. This paper proposes a system of Optical Character Recognition (OCR) which extracts the content from the uploaded picture. The main motive of the OCR is the creation of editable records from documents that already exist or picture files. The Optical Character Recognition also works on sentence discovery to protect a document's structure. The paper also presents a strategy for programmed sentence extraction utilizing the Text-rank algorithm. This approach relegates scores to the sentences by weighting the highlights like term frequency, word events, and noun weight and expressions. The outcome of this work demonstrates that our approach gives more accuracy and also provides text-to-speech with the interpretation of one language to another while maintaining coherence and accomplishes superior results when compared with existing methods.

Keywords – **Natural Language Processing, Optical-Character Recognition, Summarization, Text-rank algorithm, Text-to-speech.**

Date of Submission: June 15, 2020 Date of Acceptance: July 14, 2020

I. INTRODUCTION

As the quantity of data on the internet is increasing by the minute, in numerous groups like content, video and images, it has gotten to be troublesome for the person to discover relevant information about his intrigue [1]. Assume client questions for data on the web, user may get a huge number of result records which may not be essential to his/her concern. To find relevant data, a client must look through the complete records, this causes data over-burden issues which lead to a waste of time and unnecessary hard work. To bargain with this predicament, programmed content summarization plays an imperative role.

Content Modelling Intelligence is utilized to condense the huge sums of literary information. This model accomplishes the following benefits:

- Expelling some redundancies. The user does not squander time perusing tedious data.
- Summarization permits users for the evacuation of information that is not fundamental to the acknowledgement of the document.
- Choosing the foremost pertinent and vital sentences from the textual content.
- The extraction of sentences from the report or any text document has to keep coherence in the mind thereby the summary maintaining the essence of the initial record.
- Text to Speech availability is one of the key features utilized to listen to the outlined content report which can be spared for the advance reference.
- Single Document Summarization allows users to upload and summarize the document.

Content Modelling is one of the foremost challenging and serious issues within the field of Natural language processing (NLP). It could be prepared by producing a brief and important outline of content from different content assets such as raw text, news API, images, and document summarization. This model has applications in several areas such as news artefacts, emails, blog posts, research papers and text documents to receive a summary of the results obtained.

In our model we deal with four types of input data. They are categorized as:

- News API Content.
- Image Content.
- Raw text Content.
- Text Document Content.

So, when a client takes a picture of the text that he or she needs to print, uploads the image in an OCR program, the model creates an editable content record which is feasible for the client. This record can be utilized to print or distribute the required text. Sometimes, raw content we are managing with is fair as well long to peruse, to analyze, and as well long to devour. Summarization permits you to extract the significant points and points from a bit of the actual content, making it simpler to devour and analyze. The method of recognizing and demonstrating the most critical components of a record or a collection of records is summarization. These distinguished components can be related in a much smaller volume in comparison with the initial context. Through such a strategy, we get an outline of the original content that can best convey the meaning of the original content. This paper emphasizes the study and performance analysis of programmed content summarizers for different languages. The contents of each section may be provided to understand easily about the paper.

II. RELATED WORK

The earliest study of text summarization started in 1958, which included the recurrence of a particular word, positioning of the sentence, and the presence of keywords.

Automatic Text Summarizer was a paper written by Dr. Annapurna P Patil [2]. The authors describe three main processes-
1) Preprocessing: Text is separated based on punctuations, stop words and the remaining words are stored in a list along with the corresponding sentence.
2) Text-Rank algorithm is implemented.
3) Then the authors assign a weighted score to every sentence in the input document. The authors then simply choose the sentences with the largest weights and show them as per their beginning arrangement within the document.

An Overview of Text Summarization Techniques was a paper written by Narendra Andhale [3]. In this paper the authors discuss the different methods used to summarize text. The methods identified are –

1) Extractive Methods: a. Term Frequency-Inverse Document Frequency Method b. Cluster-Based Method c. Text Summarization with Neural Network d. Text Summarization with Fuzzy Logic

2) Abstractive Methods: a. Structured Based Approach b. Tree-Based Method c. Template Based Method d. Ontology-Based Method.

Automatic Text Summarization of News Articles was a paper written by Prakhar Sethi et al [4]. In this paper the authors tweaked methods that had already been researched to leverage the fact that only news articles were being dealt with. The following methods were used-
A. Sentence Tokenization
B. Part of speech tagging for tokenized words.
C. Pronoun Resolution.
D. Lexical Chain formation.
E. Scoring Mechanisms.
F. Summary Extraction.
The authors were able to automatically summarize news articles and compare them and analyze which scoring parameter yields superior results.

A Survey of Automatic Text Summarization Techniques for Indian and Foreign Languages was a paper written by Prachi Shah et al [5]. The authors discuss the various summarization techniques that work with multiple languages- Indian and foreign. Pre-processing, processing and extraction are the three major steps described in the system. The first phase of preprocessing involves segmentation, tokenization, and stop word removal. Sentence position, sentence length, numerical data, presence of inverted comma, and keywords in the sentence are some of the features to be considered. Finally, a machine learning model is implemented to decide if a particular sentence should be appended to the final summary or not. The sentence is then given a rank based on the sentence score. Sentences with higher ranks are included in the summary.

Automatic text summarization based on sentence clustering and extraction was a paper written by Zhang Pei-Ying et al [6]. Data mining is a process of extraction of usable data from a larger set of raw data in simpler words.[7] It is the collection of raw data, analyzation of data patterns and processing of the larger data for an effective output [8]. This paper proposes a sentence similarity computing method based on the three features of the sentences - analyzing the word form feature, the word order feature, and the semantic feature. The approach consists of three steps: 1. Clusters the sentences based on the semantic distance. 2. Each cluster calculates the cumulative sentence similarity (multi-features combination method.) 3. Chooses the topic sentences by some extraction rules. Weights are used to describe the contribution of each sentence feature. The proposed system uses the weight to describe the contribution of each feature of the sentence, describing the sentence similarity more precisely.

Implementation of an Optical Character Reader (OCR) for the Bengali Language was a paper written by Muhammed Tawfiq Chowdhury et al [9]. In the proposed model, the authors consider two different sets of inputs for OCR. The two sets of inputs contain images converted from test and scanned images of printed documents. Basically, it works in two phases - character and word detection. OCR also works on sentence detection that helps maintain a document's structure. The system is based on the Tesseract OCR Engine with the user interface developed in the Java Graphical User Interface platform. Unique identification of each and every character in the input file must be possible to enable OCR recognition [10]. Spaces among words are not detected by the OCR in many cases.

III. METHODOLOGY

Content summarization is the condensation of the source document into a smaller and meaningful summary reflecting the important information of the document without altering the meaning. Content summarization is widely categorized into extractive summarization and abstractive summarization.

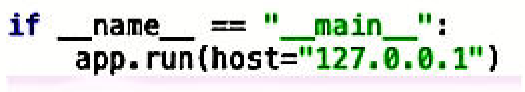
Abstractive Summarization - Abstractive content summarization is the reformulation of the sentences of the source text where it understands the fundamental concepts in the document and then conveys the most relevant information from the original text documents [2].

Extractive Summarization - Extractive content summarization extracts informative sentences or expressions from the source document and groups them to generate a summary maintaining the essence of the original document [2].

The CMI model uses an Extractive Summarization technique where the informative sentences are selected for a summary mainly based on specific criteria. The main task of extractive summarization is to choose which sentences from the input document are relevant and can be used in the summary.
For our model, we use the following software and packages:

3.1 Flask -

In our model we use Flask as a web framework that uses localhost port 5000 when it is running on command (python.py). Flask is a small-scale web framework composed in Python. It is categorized as a microframework since it does not require specific tools, devices, or libraries. It does not have a database abstraction layer, validation of the form, or other modules where pre-existing third-party libraries give common purpose. The Flask also supports extensions where an application could be added, features as if they were going to be executed in Flask itself.



```
if __name__ == "__main__":
    app.run(host="127.0.0.1")
```

The above code snippet is used in our model as a web framework that uses the localhost port 5000 providing the user to input their data on the web page and summarize it on their requirements.

Flask is designed to make getting started quick and easy, with the ability to scale up to complex applications. Flask is used as "Back End" as well as Django. Flask is additionally simple to get started with as an apprentice since there's small boilerplate code for getting a straightforward app up and running. Extensions exist for object-relational mappers, form approval, taking care of uploads, several different open authentication technologies, and a few common system-related micro-framework tools. Extensions are upgraded far more regularly than the main Flask program. It can be utilized to develop a custom computer program system on the best of it and supports Python 2.7 and 3.5 and later.

Some of the features of Flask are:

1. Support for secure cookies (client-side sessions).
2. Development of the server and debugger.
3. Compatibility of Google App Engine
4. Desired features can be enhanced by the extensions that are available.

3.2 Text-rank Algorithm

This is an extractive and unsupervised ranking algorithm for content summarization [2]. Simple representation of the working of the text rank algorithm is shown in Figure 1 below. Text-rank mainly has two NLP tasks- keyword extraction and sentence extraction. The flow of the text rank algorithm is as follows:

- The first step will be to concatenate the entire text of the articles.
- Then split the text into single sentences.
- We'll consider vector representation (word embedding) for each and every sentence in the next step.
- Similarities are then measured between the sentence vectors and stored in a matrix.
- This matrix of similarities is then converted into a graph, with the sentences considered as vertices and similarities as edges.
- Finally, sentences with the most noteworthy score are included in the final summary.

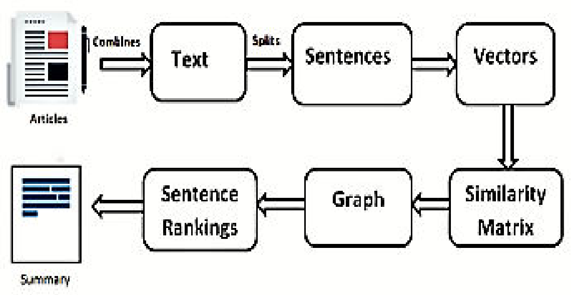


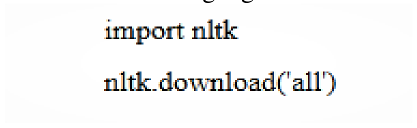
Fig 1. Main phases in the text rank algorithm.

First, the articles are broken down into text and split into sentences. These sentences are given by vectors and similarity matrices and are then represented as Graph and the sentences are ranked accordingly. The Fig 1. represents the main phases in the text-rank algorithm carried out. Finally, the summary is given to the user.

3.3 NLTK (Natural Language toolkit)

Our architecture, which is a mixture of symbolic and statistical natural language processing (NLP) [11] libraries and programs written in the Python programming language for English, uses Natural Language Toolkit. It is a leading network for the development of Python programs for human language data. It supports Windows, Mac OS X, and Linux. Best of all, the Natural language toolkit is a community-driven project, which is free and open-source. A perfect tool for teaching computational linguistics and working with Python language, and a stunning library to work on Natural language.

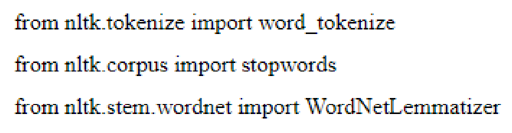
Natural language toolkit is one of the most important toolkits used in Content Modelling Intelligence. It is easy to import all the natural language toolkit packages in the python. Our model uses an import natural language toolkit to download all-natural language toolkits.



```
import nltk

nltk.download('all')
```

Fig 2. Natural language tool kit.

The above code snippet in Fig 2. imports all the natural language toolkits which also use several other models such as corpus which is a collection of text documents wherein text mining is applied in the natural language processing to obtain results.



```
from nltk.tokenize import word_tokenize

from nltk.corpus import stopwords

from nltk.stem.wordnet import WordNetLemmatizer
```

Fig 3. Tokenization.

Word_tokenize package is imported as shown in Fig 3. from the natural language toolkit under the concept of tokenizing in the Natural Language Toolkit to tokenize i.e., to split a large number of sentences or a number of texts into words. Stopwords package is imported under the corpus which is a collection of text documents in the Natural language toolkit to remove the different stopwords in the pre-processed text document. WordNetLemmatizer is imported from wordnet under nltk.stem.wordnet where in lemmatizer is used to group the different influenced forms of a word together so that they're being evaluated as one single object.

3.4 OCR

Optical Character Recognition (OCR) is the automatic conversion of text-typed images, manually written or printed content into machine-encoded material, whether from a scanned image, a report photograph, a scene photograph, etc. It is the innovation that changes over pictures to content so that computers can extricate content information from picture records [7].

OCR innovation classifies optical designs in computerized pictures, based on how they compare to alphanumeric characters. OCR can be a gigantic efficient alternate route for students, analysts, and business visionaries who deal with different parts of records and documents. Once you handle a report with OCR innovation, you'll be able effortlessly to retrieve the content data.

The optical character recognition is further carried out by the tesseract software wherein strings are recognized and further removal of stopwords and checking the grammar takes place to output the summarized text.

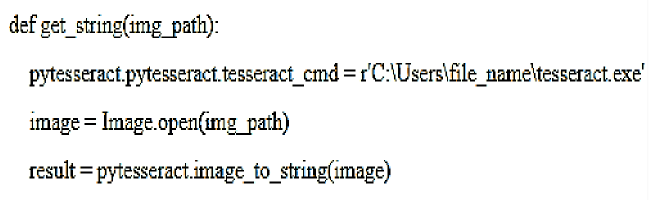
3.5 Tesseract

This model uses the Tesseract software 64-bit version tesseract - OCR - w64 - setup - v5.0.0 - alpha.20200223 which is both available in 32 bit and 64 bit. The tesseract is open-source software that is an engine for Optical Character Recognition for several different operating systems which is powered by the Apache license. Tesseract is recognized as one of the foremost precise open-source OCR engines accessible at this point. It is available for Linux, Windows, and Mac OS X.

Tesseract can distinguish whether the content is monospaced or proportionally dispersed. Tesseract is reasonable for use as a backend and utilized for more complicated OCR errands including layout examination. The line finding algorithm is one of the few parts of Tesseract software.

3.5.1 import pytesseract

Importing the pytesseract in the backend program is used to run the Tesseract application installed on your desktop.



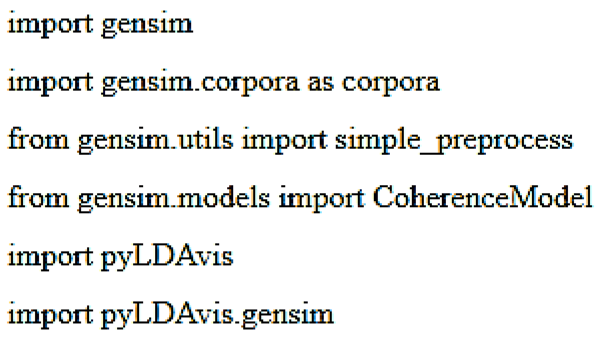
```
def get_string(img_path):

    pytesseract.pytesseract.tesseract_cmd = r'C:\Users\file_name\tesseract.exe'

    image = Image.open(img_path)

    result = pytesseract.image_to_string(image)
```

Fig 4. Tesseract path for optical character recognition.

This code snippet represented in Fig 4. is used to run the tesseract software by specifying the path of the tesseract software installed and carries out the optical character recognition process and converts each sentence into strings for the summarization process.

3.6 Gensim

Gensim is a Python library for topic modeling, indexing of documents, and retrieval of similarities. This module automatically summarizes the given text, by extracting one or more important sentences from the text document. In a similar way, it can also extract keywords.



```
import gensim

import gensim.corpora as corpora

from gensim.utils import simple_preprocess

from gensim.models import CoherenceModel

import pyLDAvis

import pyLDAvis.gensim
```

Fig 5. Gensim to pre-process the data.

Import gensim is used to import the gensim package as shown in Fig 5. which is used to check the semantics of the sentences and preprocess the data and the coherence model is imported from gensim.models which enable the topic modeling in the news API sector. Gensim lets you peruse the content and overhaul the word reference, one line at a time, without stacking the complete content record into the system memory. Corpora imported from gensim.corpora is a dictionary feature enabled in gensim that checks the meanings to form a grammatically correct sentence. Gensim provides algorithms for topic modeling to extract the underlying topics from the huge textual content.

3.7 gTTS

Text-to-Speech Powered by Google may be a screen peruser application created by Google for its Android working frameworks. It powers applications to examine out loud (talk) the content on the screen with back for numerous dialects. The Text to speech API supports a few languages counting English, Hindi, Tamil, French, German, and numerous more.

```
from gtts import gTTS
```

Import gTTS from gtts (Google text to speech) which is used to import packages where the summarized text is played as an audio file (mp3 file format).

gTTS which works flawlessly in python3 but it needs an internet connection to work since it depends on Google to urge the audio information. The audio can be saved in an mp3 file. Google text to speech helps clients to listen to different types of languages through the voice output of the summarized output. Clients are also introduced with the download button which enables clients to download and listen to the summarized data in the Content Modeling Intelligence.

3.8 Google Translator

Google translator is used in Content Modeling Intelligence to translate the summarized text into other different languages of the user's need. Google Translate is a free multilingual measurable and neural machine interpretation benefit created by Google, to translate content and websites from one language into another. Google Translate supports 109 languages at different levels but in this model, we offer 9 different language translations such as Hindi, Kannada, Marathi, French, etc. It offers a website interface to translate the given text.

```
from googletrans import Translator
```

Import Translator from googletrans which provides a feature to translate from the English language to any language available in the translation of their need. We convert the summarized text from English to user-specified language.

IV. IMPLEMENTATION AND RESULTS

The below Fig 2. illustrates the architectural design of the content modeling intelligence model. The CMI (Content Modeling Intelligence) model is designed to accept 4 types of input data as follows:

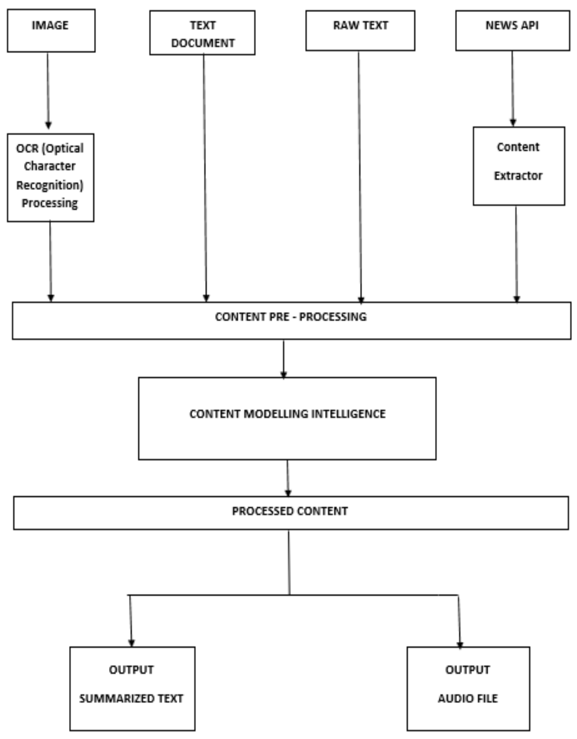


Fig 6. The system architecture of the CMI (Content modeling intelligence).

The CMI (Content Modelling Intelligence) model is designed to accept 4 types of input data as shown in Fig 6 as follows:

- Raw text

- Image
- News API
- Text document

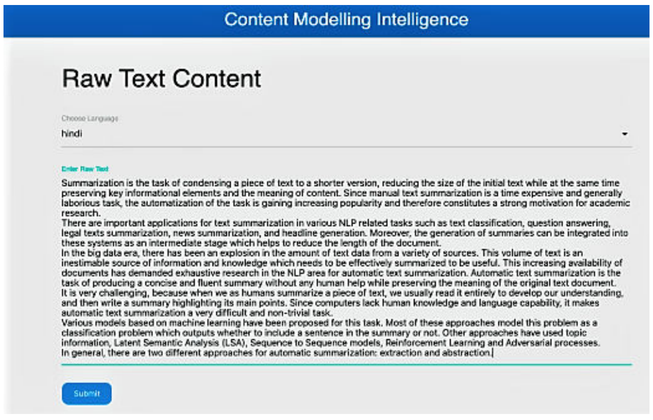
4.1 Implementation of CMI for the Raw Data.

When raw data is provided to the model as input by the user as shown in Fig 3, the system pre-processes the data in the steps mentioned below:

- The provided content is extracted from the textual content and divided into sentences based on the punctuation symbols like periods(.), exclamation marks(!) and question marks(?).
- The sentences are divided into words on blank spaces.
- The words with less significance and the articles such as 'a', 'an 'the', 'of', 'I' are removed from the text.
- The rest of the words in the text are changed into their original form. Example: 'replacing', 'replacement', 'replaced' are all converted back to 'replace'.
- The resulting words are listed concurring to their sentences.

Now, each language that is being offered in the model is assigned a language code (Example: Marathi as 'me', Hindi as 'hi', etc). Then the text-rank algorithm is implemented.

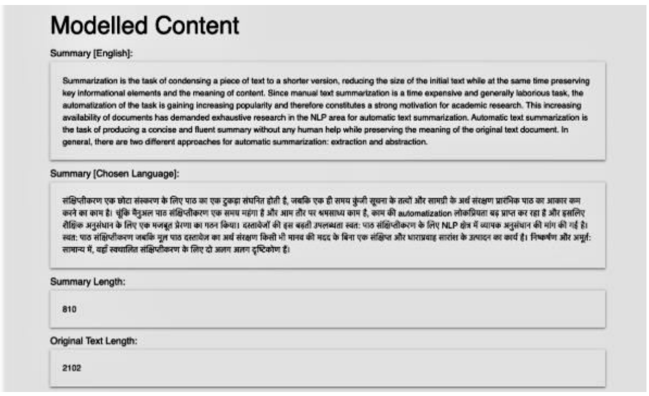
The keywords are extracted from the given raw text and the sentences are split into individual words that are being treated as tokens. The raw text is provided as shown in Fig 7. The punctuation symbols such as ('[', '\\', ',') are being replaced according to the semantics of the sentences. The same procedure continues for each sentence of the textual content and the grammatical errors are being removed by using the GingerIt package. The total length of the initial raw text and the summarized text is being evaluated and displayed as output. As shown in Fig 8. This summarized text can also be translated to any available language options. The audio conversion feature is also available that can be accessible for download.



Choose Language
hindi

Enter Raw Text
Summarization is the task of condensing a piece of text to a shorter version, reducing the size of the initial text while at the same time preserving key informational elements and the meaning of content. Since manual text summarization is a time expensive and generally laborious task, the automatization of the task is gaining increasing popularity and therefore constitutes a strong motivation for academic research.
There are important applications for text summarization in various NLP related tasks such as text classification, question answering, legal texts summarization, news summarization, and headline generation. Moreover, the generation of summaries can be integrated into these systems as an intermediate stage which helps to reduce the length of the document.
In the big data era, there has been an explosion in the amount of text data from a variety of sources. This volume of text is an inestimable source of information and knowledge which needs to be effectively summarized to be useful. This increasing availability of documents has demanded exhaustive research in the NLP area for automatic text summarization. Automatic text summarization is the task of producing a concise and fluent summary without any human help while preserving the meaning of the original text document.
It is very challenging, because when we as humans summarize a piece of text, we usually read it entirely to develop our understanding, and then write a summary highlighting its main points. Since computers lack human knowledge and language capability, it makes automatic text summarization a very difficult and non-trivial task.
Various models based on machine learning have been proposed for this task. Most of these approaches model this problem as a classification problem which outputs whether to include a sentence in the summary or not. Other approaches have used topic information, Latent Semantic Analysis (LSA), Sequence to Sequence models, Reinforcement Learning and Adversarial processes.
In general, there are two different approaches for automatic summarization: extraction and abstraction.

Fig 7. Input for Raw Text.



Summary [English]:

Summarization is the task of condensing a piece of text to a shorter version, reducing the size of the initial text while at the same time preserving key informational elements and the meaning of content. Since manual text summarization is a time expensive and generally laborious task, the automatization of the task is gaining increasing popularity and therefore constitutes a strong motivation for academic research. This increasing availability of documents has demanded exhaustive research in the NLP area for automatic text summarization. Automatic text summarization is the task of producing a concise and fluent summary without any human help while preserving the meaning of the original text document. In general, there are two different approaches for automatic summarization: extraction and abstraction.

Summary [Chosen Language]:

संक्षिप्तीकरण एक छोटा संस्करण के लिए पाठ का एक टुकड़ा संघनित होती है, जबकि एक ही समय कुंजी सूचना के तत्वों और सामग्री के अर्थ संरक्षण प्रारंभिक पाठ का आकार कम करने का काम है। चूंकि मैनुअल पाठ संक्षिप्तीकरण एक समय महंगा है और आम तौर पर श्रमसाध्य काम है, काम की automatization लोकप्रियता बढ़ प्राप्त कर रहा है और इसलिए शैक्षिक अनुसंधान के लिए एक मजबूत प्रेरणा का गठन किया। दस्तावेजों की इस बढ़ती उपलब्धता स्वत: पाठ संक्षिप्तीकरण के लिए NLP क्षेत्र में व्यापक अनुसंधान की मांग की गई है। स्वत: पाठ संक्षिप्तीकरण जबकि मूल पाठ दस्तावेज का अर्थ संरक्षण किसी भी मानव की मदद के बिना एक संक्षिप्त और धाराप्रवाह सारांश के उत्पादन का कार्य है। निष्कर्षण और अमूर्त: सामान्य में, वहाँ स्वचालित संक्षिप्तीकरण के लिए दो अलग अलग दृष्टिकोण है।

Summary Length:

810

Original Text Length:

2102

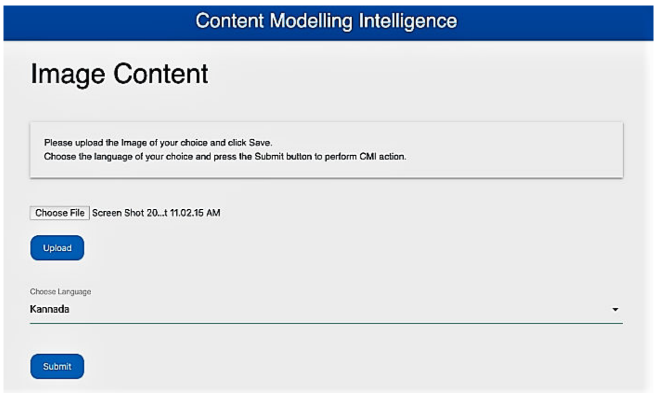
Fig 8. Output for Raw Text.

4.2 Implementation of CMI for Image Content

When an image is scanned and uploaded by the user in the Content modeling intelligence model, the OCR works by analyzing the image and interpreting the characters into codes that can be utilized for the data processing. The basic steps in the OCR process are image acquisition, pre-processing, segregation, feature extraction, classification, and summarization. The user's image is processed by the Tesseract software for Optical Character Recognition. The layout of the given image is considered and the scanned image is analyzed for light and dark areas.

The lighter areas would then be recognized as background and darker areas as written characters. Then the computer processes the dark areas recognized as characters to find alphabetic letters, numeric digits, and symbols. Then by using the line finding algorithm, each sentence is recognized as a string. These strings are further split into individual words by using a tokenizer and lemmatizer from the natural language toolkit. The words are then weighted using the text rank algorithm and processed by the GingerIt module for a semantical and error free summary of the original image. In this process summarized text can be translated into any desired language viable on the model as well as converted into an audio file in mp3 format. This audio file can be played as well as accessed for download.

The below Fig 9. Represents the input data i.e., the image to be provided in the website and the Fig 10. Represents the output data provided to the user from the image.

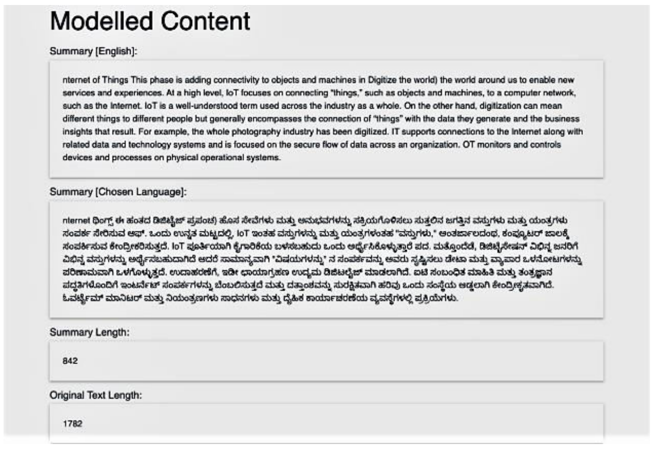


Please upload the Image of your choice and click Save.
Choose the language of your choice and press the Submit button to perform CMI action.

Choose File Screen Shot 20...t 11.02.15 AM

Choose Language
Kannada

Fig 9. Input for Image Data.


Modelled Content

Summary [English]:

nternet of Things This phase is adding connectivity to objects and machines in Digitize the world) the world around us to enable new services and experiences. At a high level, IoT focuses on connecting "things," such as objects and machines, to a computer network, such as the Internet. IoT is a well-understood term used across the industry as a whole. On the other hand, digitization can mean different things to different people but generally encompasses the connection of "things" with the data they generate and the business insights that result. For example, the whole photography industry has been digitized. IT supports connections to the Internet along with related data and technology systems and is focused on the secure flow of data across an organization. OT monitors and controls devices and processes on physical operational systems.

Summary [Chosen Language]:

nternet ಥಿಂಗ್ಸ್ ಈ ಹಂತದ ಡಿಜಿಟೈಜ್ ಪ್ರಪಂಚ) ಹೊಸ ಸೇವೆಗಳು ಮತ್ತು ಅನುಭವಗಳನ್ನು ಸಕ್ರಿಯಗೊಳಿಸಲು ಸುತ್ತಲಿನ ಜಗತ್ತಿನ ವಸ್ತುಗಳು ಮತ್ತು ಯಂತ್ರಗಳು ಸಂಪರ್ಕ ಸೇರಿಸುವ ಆಫ್. ಒಂದು ಉನ್ನತ ಮಟ್ಟದಲ್ಲಿ, IoT ಇಂತಹ ವಸ್ತುಗಳನ್ನು ಮತ್ತು ಯಂತ್ರಗಳಂತಹ "ವಸ್ತುಗಳು," ಅಂತರ್ಜಾಲದಂಥ, ಕಂಪ್ಯೂಟರ್ ಜಾಲಕ್ಕೆ ಸಂಪರ್ಕಿಸುವ ಕೇಂದ್ರೀಕರಿಸುತ್ತದೆ. IoT ಪೂರ್ತಿಯಾಗಿ ಕೈಗಾರಿಕೆಯ ಬಳಸಬಹುದು ಒಂದು ಅರ್ಥೈಸಿಕೊಳ್ಳುತ್ತಾರೆ ಪದ. ಮತ್ತೊಂದೆಡೆ, ಡಿಜಿಟೈಸೇಷನ್ ವಿಭಿನ್ನ ಜನರಿಗೆ ವಿಭಿನ್ನ ವಸ್ತುಗಳನ್ನು ಅರ್ಥೈಸಬಹುದಾಗಿದೆ ಆದರೆ ಸಾಮಾನ್ಯವಾಗಿ "ವಿಷಯಗಳನ್ನು" ನ ಸಂಪರ್ಕವನ್ನು ಅವರು ಸೃಷ್ಟಿಸಲು ಡೇಟಾ ಮತ್ತು ವ್ಯಾಪಾರ ಒಳನೋಟಗಳನ್ನು ಪರಿಣಾಮವಾಗಿ ಒಳಗೊಳ್ಳುತ್ತದೆ. ಉದಾಹರಣೆಗೆ, ಇಡೀ ಛಾಯಾಗ್ರಹಣ ಉದ್ಯಮ ಡಿಜಿಟಲೈಜ್ ಮಾಡಲಾಗಿದೆ. ಐಟಿ ಸಂಬಂಧಿತ ಮಾಹಿತಿ ಮತ್ತು ತಂತ್ರಜ್ಞಾನ ಪದ್ಧತಿಗಳೊಂದಿಗೆ ಇಂಟರ್ನೆಟ್ ಸಂಪರ್ಕಗಳನ್ನು ಬೆಂಬಲಿಸುತ್ತದೆ ಮತ್ತು ದತ್ತಾಂಶವನ್ನು ಸುರಕ್ಷಿತವಾಗಿ ಹರಿವು ಒಂದು ಸಂಸ್ಥೆಯ ಅಡ್ಡಲಾಗಿ ಕೇಂದ್ರೀಕೃತವಾಗಿದೆ. ಓವರ್ಟೈಮ್ ಮಾನಿಟರ್ ಮತ್ತು ನಿಯಂತ್ರಣಗಳು ಸಾಧನಗಳು ಮತ್ತು ದೈಹಿಕ ಕಾರ್ಯಾಚರಣೆಯ ವ್ಯವಸ್ಥೆಗಳಲ್ಲಿ ಪ್ರಕ್ರಿಯೆಗಳು.

Summary Length:

842

Original Text Length:

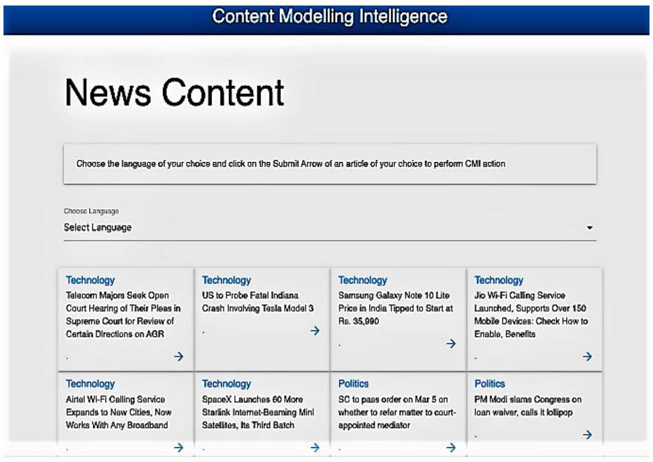
1782

Fig 10. Output for Image Data.

4.3 Implementation of CMI for News API Content

News API is a simple and easy-to-use API that returns JSON metadata for headlines and articles that is scattered all over the internet. The news API enables computer applications to request stories and information about their stories (metadata) from them. News API content is one of the important features in Content Modeling Intelligence. News API are not freely available, so we use MongoDB as the database which acts as a server and the sample articles can be accessed from it. There are stored News API articles in MongoDB that are categorized on the basis of their topics into political, business and technology. Once all the news articles are made available in MongoDB, the user can get a summary of any desired article from the available list.

The Fig 11. represents the APIs to be summarized by the user in the website and the Fig 12. represents the output data provided to the user from the image.



News Content

Choose the language of your choice and click on the Submit Arrow of an article of your choice to perform CMI action

Choose Language
Select Language

Technology
Telecom Majors Seek Open Court Hearing of Their Pleas in Supreme Court for Review of Certain Directions on AGR

Technology
US to Probe Fatal Indiana Crash Involving Tesla Model 3

Technology
Samsung Galaxy Note 10 Lite Price in India Tipped to Start at Rs. 35,990

Technology
Jio Wi-Fi Calling Service Launched, Supports Over 150 Mobile Devices: Check How to Enable, Benefits

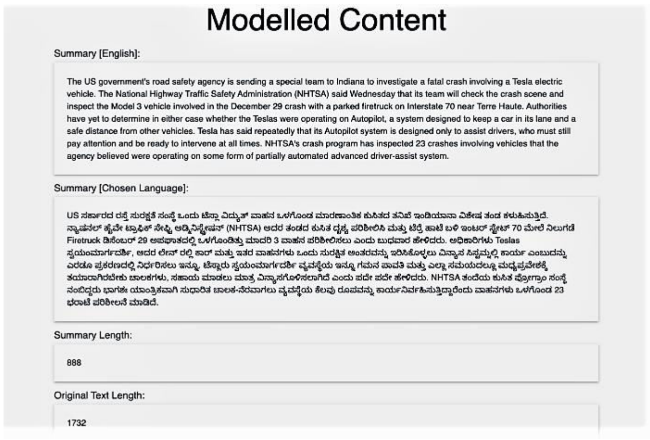
Technology
Airtel Wi-Fi Calling Service Expands to New Cities, Now Works With Any Broadband

Technology
SpaceX Launches 60 More Starlink Internet-Beaming Mini Satellites, Its Third Batch

Politics
SC to pass order on Mar 5 on whether to refer matter to court-appointed mediator

Politics
PM Modi slams Congress on loan waiver, calls it lollipop

Fig 11. Input for News API.


Modelled Content

Summary [English]:

The US government's road safety agency is sending a special team to Indiana to investigate a fatal crash involving a Tesla electric vehicle. The National Highway Traffic Safety Administration (NHTSA) said Wednesday that its team will check the crash scene and inspect the Model 3 vehicle involved in the December 29 crash with a parked firetruck on Interstate 70 near Terre Haute. Authorities have yet to determine in either case whether the Teslas were operating on Autopilot, a system designed to keep a car in its lane and a safe distance from other vehicles. Tesla has said repeatedly that its Autopilot system is designed only to assist drivers, who must still pay attention and be ready to intervene at all times. NHTSA's crash program has inspected 23 crashes involving vehicles that the agency believed were operating on some form of partially automated advanced driver-assist system.

Summary [Chosen Language]:

US ಸರ್ಕಾರದ ರಸ್ತೆ ಸುರಕ್ಷತೆ ಸಂಸ್ಥೆ ಒಂದು ಟೆಸ್ಲಾ ವಿದ್ಯುತ್ ವಾಹನ ಒಳಗೊಂಡ ಮಾರಣಾಂತಿಕ ಕುಸಿತದ ತನಿಖೆ ಇಂಡಿಯಾನಾ ವಿಶೇಷ ತಂಡ ಕಳುಹಿಸುತ್ತಿದೆ. ನ್ಯಾಷನಲ್ ಹೈವೇ ಟ್ರಾಫಿಕ್ ಸೇಫ್ಟಿ ಅಡ್ಮಿನಿಸ್ಟ್ರೇಷನ್ (NHTSA) ಅದರ ತಂಡದ ಕುಸಿತ ದೃಶ್ಯ ಪರಿಶೀಲಿಸಿ ಮತ್ತು ಟೆರ್ರೆ ಹಾಟೆ ಬಳಿ ಇಂಟರ್ ಸ್ಟೇಟ್ 70 ಮೇಲೆ ನಿಲುಗಡೆ Firetruck ಡಿಸೆಂಬರ್ 29 ಅಪಘಾತದಲ್ಲಿ ಒಳಗೊಂಡಿತ್ತು ಮಾದರಿ 3 ವಾಹನ ಪರಿಶೀಲಿಸಲು ಎಂದು ಬುಧವಾರ ಹೇಳಿದರು. ಅಧಿಕಾರಿಗಳು Teslas ಸ್ವಯಂಮಾರ್ಗದರ್ಶಿ, ಅದರ ಲೇನ್ ರಲ್ಲಿ ಕಾರ್ ಮತ್ತು ಇತರ ವಾಹನಗಳು ಒಂದು ಸುರಕ್ಷಿತ ಅಂತರವನ್ನು ಇರಿಸಿಕೊಳ್ಳಲು ವಿನ್ಯಾಸ ಸಿಸ್ಟಮ್ನಲ್ಲಿ ಕಾರ್ಯ ಎಂಬುದನ್ನು ಎರಡೂ ಪ್ರಕರಣದಲ್ಲಿ ನಿರ್ಧರಿಸಲು ಇನ್ನೂ, ಟೆಸ್ಲಾರು ಸ್ವಯಂಮಾರ್ಗದರ್ಶಿ ವ್ಯವಸ್ಥೆಯ ಇನ್ನೂ ಗಮನ ಪಾವತಿ ಮತ್ತು ಎಲ್ಲಾ ಸಮಯದಲ್ಲೂ ಮಧ್ಯಪ್ರವೇಶಕ್ಕೆ ತಯಾರಾಗಿರಬೇಕು ಚಾಲಕಗಳು, ಸಹಾಯ ಮಾಡಲು ಮಾತ್ರ ವಿನ್ಯಾಸಗೊಳಿಸಲಾಗಿದೆ ಎಂದು ಪದೇ ಪದೇ ಹೇಳಿದರು. NHTSA ತಂಡೆಯ ಕುಸಿತ ಪ್ರೋಗ್ರಾಂ ಸಂಸ್ಥೆ ನಂಬಿದ್ದರು ಭಾಗಶಃ ಯಾಂತ್ರಿಕವಾಗಿ ಸುಧಾರಿತ ಚಾಲಕ-ನೆರವಾಗಲು ವ್ಯವಸ್ಥೆಯ ಕೆಲವು ರೂಪವನ್ನು ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತಿದ್ದಾರೆಂದು ವಾಹನಗಳು ಒಳಗೊಂಡ 23 ಭರಾಟೆ ಪರಿಶೀಲನೆ ಮಾಡಿದೆ.

Summary Length:

888

Original Text Length:

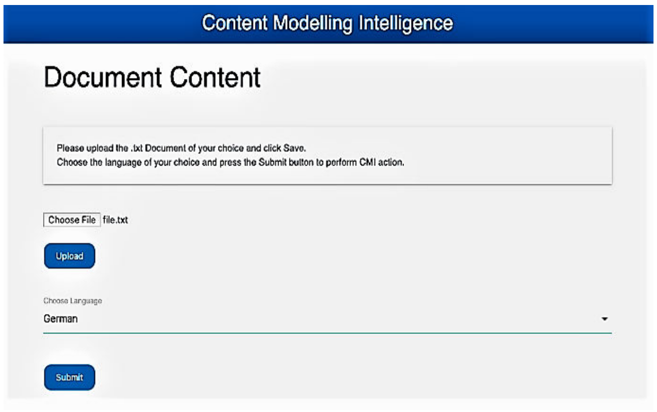
1732

Fig 12. Output for News API.

4.4 Implementation of CMI for Text Document

Implementation of the CMI for text documents works very similar to that of the raw data summarization. A text document is a text file, uploaded by the user for the summarization. An upload feature is provided in the user interface for the user to upload the desired document. This document can then be summarized and translated to any desired language and also be converted into an audio file of the provided languages.

The below Fig 13. represents the input data i.e., the text document to be provided in the website and the Fig 14. Represents the output data provided to the user from the Text document.



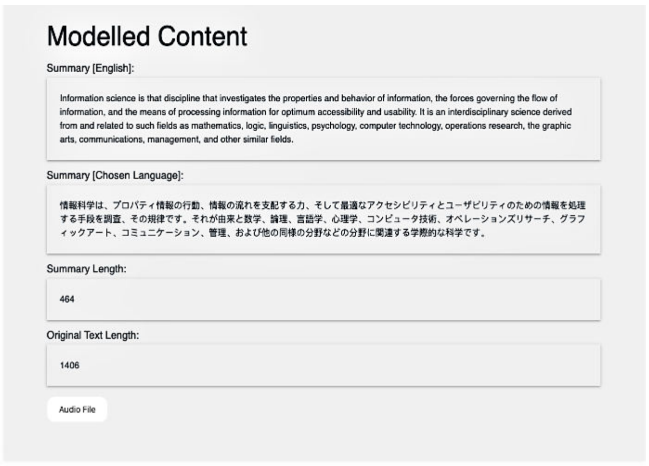
Document Content

Please upload the .txt Document of your choice and click Save.
Choose the language of your choice and press the Submit button to perform CMI action.

Choose File file.txt

Choose Language
German

Fig 13. Input for Text Document.


Modelled Content

Summary [English]:

Information science is that discipline that investigates the properties and behavior of information, the forces governing the flow of information, and the means of processing information for optimum accessibility and usability. It is an interdisciplinary science derived from and related to such fields as mathematics, logic, linguistics, psychology, computer technology, operations research, the graphic arts, communications, management, and other similar fields.

Summary [Chosen Language]:

情報科学は、プロパティ情報の行動、情報の流れを支配する力、そして最適なアクセシビリティとユーザビリティのための情報を処理する手段を調査、その規律です。それが由来と数学、論理、言語学、心理学、コンピュータ技術、オペレーションズリサーチ、グラフィックアート、コミュニケーション、管理、および他の同様の分野などの分野に関連する学際的な科学です。

Summary Length:

464

Original Text Length:

1406

Fig 14. Output for Text Document.

Each method was executed and tested on different lengths of datasets to obtain the speed of the summarizer. The speed of the CMI (Content Modelling Intelligence) with respect to the size of the input data are listed. Tests were run on the sample data collected from the internet. Below are the plots for each of the four input types that have been used where the run time against the number of characters input in the input data can be visualized.

The time complexity of the summarizer algorithm (denoted by Big O) is O(*Nw*).
where *N* is the number of words in the document (correlates to number of characters.)
& *w* is the number of unique words.

Time v/s Size Plots

From the graph, for raw text input data as shown in Fig 15., it appears that the algorithm is quadratic in time. This means that a larger input dataset would take significantly longer to give the result. It is observed that there is reduction in run time for the 8000 characters when compared to the time taken for 7000 characters, indicating that the run time is not solely dependent on the size of the input data. The reason behind the difference in the run time is due to the difference in sentence importance in the vector representation of the sentences. The weights of the edges between the vertices in the graphs varies for each piece of data causing a variation in the run time.

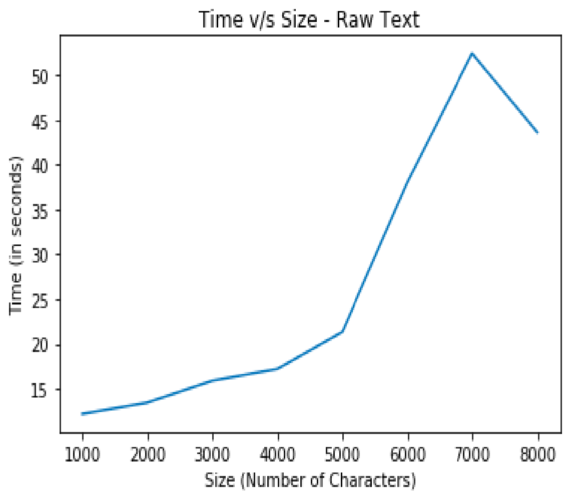


Fig 15. Time v/s Size graph for raw text.

In the plot for image content as shown in Fig 16., the line is observed to be nearly linear. Initially, an increase in runtime is observed after the increase in runtime becomes more slight.

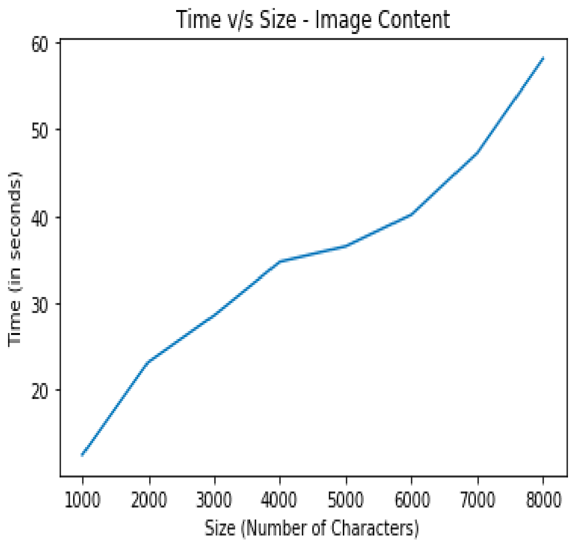


Fig 16. Time v/s Size graph for image content

In the plot for text document and news API as shown in Fig 17. and Fig 18., type input data, we observe that the graph is almost linear. This denotes a linear increase in the run time along with the increase of input data size.

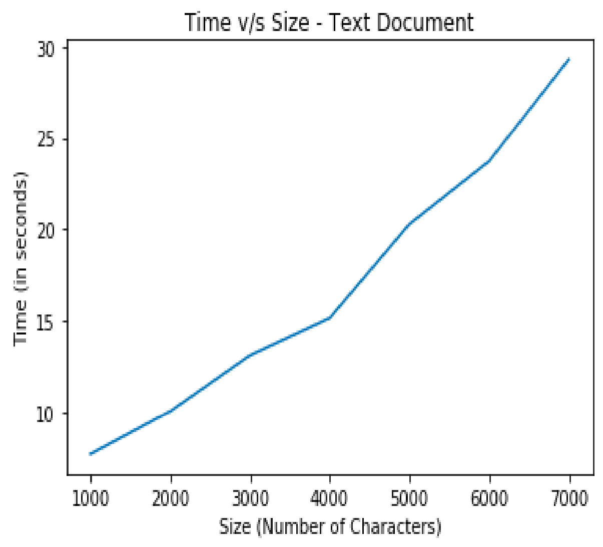


Fig 17. Time v/s Size graph for text document.

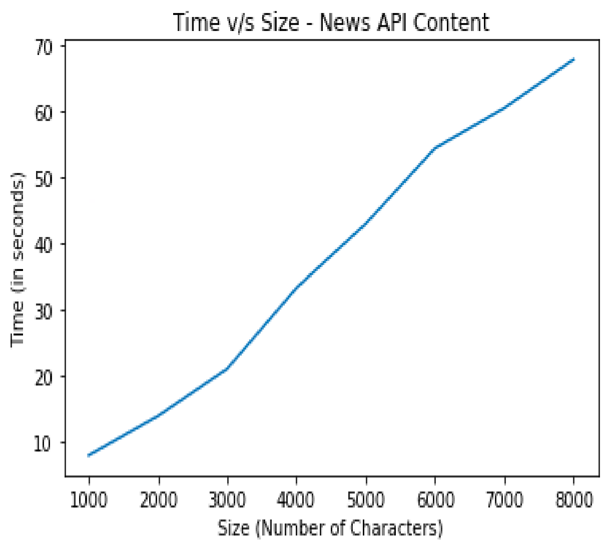


Fig 18. Time v/s Size graph for News API content.

The Content Modelling Intelligence model has the text summarization model based on 4 different input modes i.e., raw text, API content, text document and image content where the user can get the summarized content with the proper semantics which can also be converted to 9 different languages of the user's choice. The use of Text-Rank algorithm which is based on the PageRank algorithm used by Google to rank web pages provides meaningful summaries for all sizes of input. As opposed to most existing systems, this model provides an extra feature of text-to-speech to listen to the summarized text which can both be listened to and downloaded for future use. The model also integrates the feature to obtain a summary of an image containing text-based information. Therefore, a whole package of text summarization and image data summarization, translation of the summarized text to other languages and conversion of the summarized texts to an mp3 audio format is consolidated as a single unit. is provided by the suggested system.

V. CONCLUSION

This paper covers a summarization model to produce an effective summary with the least redundancy and grammatically correct sentences from the source document. This approach has demonstrated good performance for most of the summarization purposes. This CMI model mainly demonstrates 4 modules: raw data, image data, news API and single text document.

First, the raw data that is uploaded by the user is summarized into a simpler and smaller summary in English as well as any chosen language from the 9 language options provided. It also displays the original data length along with the summarized text length. An audio file of the summary in the selected language is also accessible.

Second, when an image is uploaded to the model, the image is processed using the OCR to produce a meaningful summary in English as well as any language provided in the model. This summary can further be converted into an audio file that is accessible for download, indicating the text-to-speech feature.

Third, after choosing the language, the desired news API article is chosen from the given list of articles in the model and submitted for summarization. A short summary is obtained in English as well as in the chosen language. This summary can be converted into an audio file that is accessible for download.

Fourth, any document can be uploaded as per the user's choice and any desired language can be selected for the summarization process. The data in the document is modeled and the output is displayed in English as well as the chosen language. This file can be converted as an audio file that is accessible for download.

The current extractive summarization process is convenient for certain formats of documents. Summarization framework should deliver a compelling summary in a brief time with less redundancy of words and sentences having grammatically error-free sentences. In the future, we are aiming to use features such as summarization of multiple documents in a single go and summarization of the copied link provided by the user. The effectiveness and accuracy of the system can be increased by using advanced machine learning and natural language processing techniques.

VI. REFERENCES

[1] Dr. D V Ashoka, Sanjay B Ankali, "Detection Architecture of Application Layer DDoS Attack for Internet", *International Journal of Advanced Networking and Applications* (2011).

[2] Dr. Annapurna P Patil, Shivam Dalmia, Syed Abu Ayub Ansari, Tanay Aul, Varun Bhatnagar, "Automatic Text Summarizer ", *International Conference on Advances in Computing, Communications and Informatics, IEEE* (2014).

[3] Narendra Andhale, L.A.Bewoor, " An Overview of Text Summarization Techniques " *ICACCI* (2016).

[4] Prakhar Sethi, Sameer Sonawane, Saumitra Khanwalker, R. B. Keskar, "Automatic Text Summarization of News Articles *", International Conference on Big Data, IoT and Data Science (BID) Vishwakarma Institute of Technology*, Pune, Dec 20-22 IEEE (2017).

[5] Prachi Shah, Nikitha P. Desai "A Survey of Automatic Text Summarization Techniques for Indian and Foreign Languages ", *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)* (2016).

[6] ZHANG Pei-Ying, LI Cun-he, "Automatic text summarization based on sentence clustering and extraction ", *Institute of Electrical and Electronics Engineers IEEE* (2009).

[7] R. Chetan and D. V. Ashoka, "Data mining-based network intrusion detection system: A database centric approach", 2012 *International Conference on Computer Communication and Informatics, Coimbatore, pp. 1-6, doi: 10.1109/ICCCI.2012.6158816 (2012)*

[8] D. Jasmine Guna Sundari, D. Sundar, "A Study of Various Text Mining Techniques" *International Journal of Advanced Networking and Applications (IJANA) Volume: 08, Issue: 05 Pages: 82-85 Special Issue, pp:82-85(2017).*

[9] Teddy Montoro, Abdul Muis Sobri, Wendi Usino, "Optical Character Recognition (OCR) Performance in Server-based Mobile Environment ", *International Conference on Advanced Computer Science Applications and Technologies* (2013).

[10] Dr T. Santha, M .Abhayadev "Content Based Image Retrieval Public and Private Search Engines", *Special Issue Published in Int. Jnl. Of Advanced Networking and Applications (IJANA) pp:98-102(2015)*

[11] Akash Shekar, Jeevanantham.P, "Question and Answer Extraction Using NLP", *Special Issue Published in Int. Jnl. Of Advanced Networking and Applications (IJANA)197-199*

BIBLIOGRAPHIES:

Sanjan S Malagi, Rachana Radhakrishnan, Monisha R, and Keerthana S is presently studying in BE final year Information Science &amp; Engineering, JSS ACADEMY OF TECHNICAL EDUCATION, Bangalore. Their area of interest includes Internet of Things (IOT), Natural Language Processing, Text Mining, AI and Bigdata.

Dr D. V. Ashoka is presently working as Professor in the Department of Information Science and Engineering at JSS Academy of Technical Education, Bangalore. He received his M.Tech in computer science &amp; Engineering from VTU and Ph.D degree in Computer Science and Engineering from Dr. MGR, University, Chennai. He has 25 years of academic and research experience. He has published more than 60 research papers in national / international journals and conferences. His fields of interest are Requirement Engineering, Operating System, Computer Organization, Software Architecture, data mining in image processing and Cloud Computing.