Hybrid Approach for Text Summarization with Topic Modelling and Entity Extraction

Tahir Ahmed Shaik[1], A.Vikas[2], G.V.N Pradyumna[3]

[1]Tahir Ahmed Shaik Student, Dept. of Computer Science and Engineering, Anurag Group of Institutions Ghatkesar, Hyderabad.
[2]A. Vikas, Student, Dept. of Computer Science and Engineering, Anurag Group of Institutions Ghatkesar, Hyderabad.
[3]G.V.N. Pradyumna, Student, Dept. of Computer Science and Engineering, Anurag Group of Institutions Ghatkesar, Hyderabad.

---***---

Abstract—*Information contains several important context and points that it conveys to the readers, however as the volume of information grows, understanding the context and getting the points becomes a difficult task. Moreover, it is difficult for human beings to manually extract the summary of large documents of text without tools. Hence, a system is required that condenses such information and eases out the effort for the user to understand. In this paper, we propose a system that extracts the context information and involving entities also at the same time is able to model the topic of the input textual matter. This system combines the capabilities of both extractive and abstractive based approaches to achieve the task and also includes models for topic and entity extraction.*

Keywords- information, automatic summarization, meaningful, shorter-version,

1. INTRODUCTION

Text summarization is a process of extracting or collecting important information from original text and presents that information in the form of summary. In recent years, need for summarization can be seen in various purposes and in many domains such as news articles summary, email summary, short message of news on mobile etc. The automatic summarization of text is a well- known task in the field of natural language processing (NLP). Topic modeling is a type of statistical modeling for discovering the abstract "topics" that occur in the text. The Entity Extraction process refers to extracting the involved real time entities, objects such as (people, places, objects etc.) from the text.

The Information extraction task is well known task in the field of Natural Language Processing (NLP), several research advancements have been made and contributed for the task. The approaches for text summarization can be classified into two categories

1. Extractive based approaches
2. Abstractive based approaches

1.1. Extractive based approaches

These approaches focuses on presenting the most important information out of the source text by statistical techniques such as weighting or scoring strategies applied to the texts. The approach associates weights to the sentences of the texts on the basis of the tokens and using these attached weights identifies the important sentences. This can be observed in the following figure 1.1.

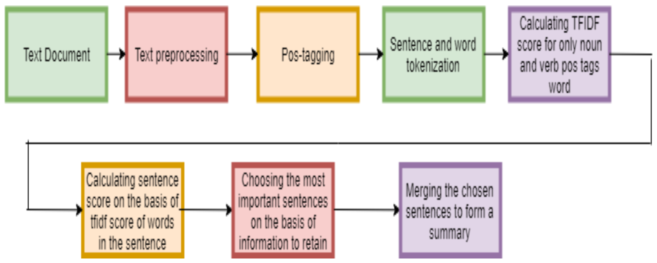

Text Document
Text preprocessing
Pos-tagging
Sentence and word tokenization
Calculating TFIDF score for only noun and verb pos tags word
Calculating sentence score on the basis of tfidf score of words in the sentence
Choosing the most important sentences on the basis of information to retain
Merging the chosen sentences to form a summary

Fig 1.1 Extractive processes for summarization.

1.2. Abstractive based approaches

These approaches involve a much deeper learning approaches for producing the context-full summaries. These approaches make use of Deep learning AI models such as Neural Networks for their capabilities of producing summaries that are more machine generated than being just copied from the source text. Below figure is an example of how an abstractive system works. The outputs generated from these approaches are more human like and dynamic rather than just being copied from the source text. This can be observed in the following figure 1.2.

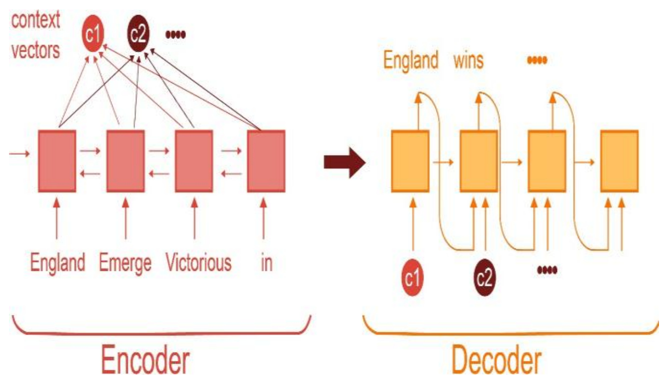


Fig 1.2 Abstractive processes for summarization.

2. APPLICATIONS OF SYSTEM

2.1 Application Areas

A few not all, application areas where there is a scope for summarization and information extraction.

2.1.2 Newsletters

Many weekly newsletters take the form of an introduction followed by a curated selection of relevant articles. Summarization would allow organizations to further enrich newsletters with a stream of summaries (versus a list of links), which can be a particularly convenient format in mobile.

2.1.3. Search marketing and SEO

When evaluating search queries for SEO, it is critical to have a well-rounded understanding of what your competitors are talking about in their content. This has become particularly important since Google updated its algorithm and shifted focus towards topical authority (versus keywords). Multi-document summarization can be a powerful tool to quickly analyze dozens of search results, understand shared themes and skim the most important points.

2.1.4. Internal document workflow

Large companies are constantly producing internal knowledge, which frequently gets stored and under-used in databases as unstructured data. These companies should embrace tools that let them re-use already existing knowledge. Summarization can enable analysts to quickly understand everything the company has already done in a given subject, and quickly assemble reports that incorporate different points of view.

2.1.5. Legal contract analysis

Related to point 4 (internal document workflow), more specific summarization systems could be developed to analyze legal documents. In this case, a summarizer might add value by condensing a contract to the riskier clauses, or help you compare agreements.

2.1.6. Social media marketing

Companies producing long-form content, like whitepapers, e-books and blogs, might be able to leverage summarization to break down this content and make it sharable on social media sites like Twitter or Facebook. This would allow companies to further re-use existing content.

2.1.7. Email overload

Companies like Slack were born to keep us away from constant emailing. Summarization could surface the most important content within email and let us skim emails faster.

2.1.8. Science and R&D

Academic papers typically include a human-made abstract that acts as a summary. However, when you are tasked with monitoring trends and innovation in a given sector, it can become overwhelming to read every abstract. Systems that can group papers and further compress abstracts can become useful for this task.

2.1.9. Help desk and customer support

Knowledge bases have been around for a while, and they are critical for SAAS platforms to provide customer support at scale. Still, users can sometimes feel overwhelmed when browsing help docs. Could multi-document summarization provide key points from across help articles and give the user a well-rounded understanding of the issue?

2.1.10. Helping disabled people

As voice-to-text technology continues to improve, people with hearing disabilities could benefit from summarization to keep up with content in a more efficient way.

3. PROPOSED METHODOLOGY

The existing approaches do provide a solution in generating summary to the text documents, but there are few limitations when dealing with the approaches.

Extractive:

- The information extracted is purely copied from the original text and hence it does not intend to produce the context or any new inferences.
- The results of these approaches are static in nature and lack dynamism.

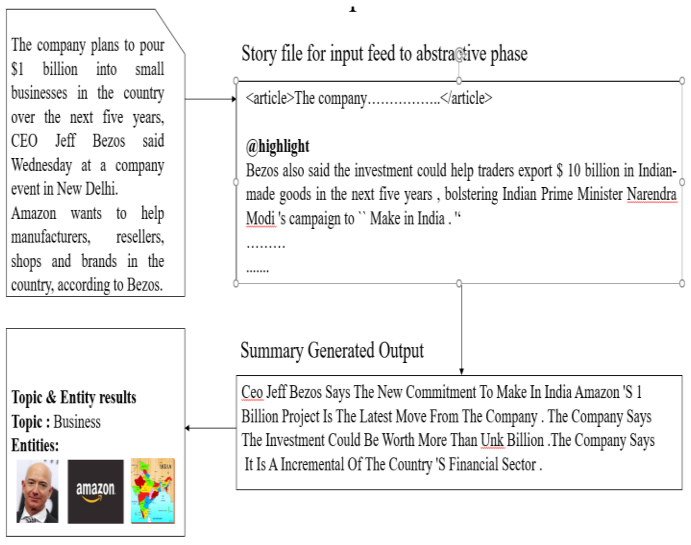
Abstractive:

- These models suffer generating Out of Vocabulary terms (OOV) and generate unknown tokens.
- Also certain models suffer from generating repeated tokens in the output.

So a system that can use both approaches to generate a summary and also to provide a scope for information extraction such as text classification and entity identifications is required for generating more understandable and accurate summary. The combined system aggregates to the powers of such techniques which can be used in the field of text summarization.

3.1. Proposed System

The proposed system provides a hybrid architecture of summarization that combines the power of dual approaches of extractive and the abstractive methodology techniques, apart from it, the proposed system combines the task of text classification for modeling the source text topic and identifying the real time entities and/or objects involved/discussed in text. This Hybrid combinational approach provides with a much higher and deeper level of information to the user that is much simple and leading to be more dynamic for inference. Below fig 3.1 is an example on how the proposed system works. There are many aspects and research regarding automatic document summarization and information extraction that, apart from their importance, cannot be investigated. Our system work is to a wider scope to build a system that enables for information condensation, classification and extraction through the combined approaches.



The company plans to pour $1 billion into small businesses in the country over the next five years, CEO Jeff Bezos said Wednesday at a company event in New Delhi.
Amazon wants to help manufacturers, resellers, shops and brands in the country, according to Bezos.

<article>The company……………..</article>

@highlight
Bezos also said the investment could help traders export $ 10 billion in Indian-made goods in the next five years , bolstering Indian Prime Minister Narendra Modi 's campaign to `` Make in India . ''
………
…….

Ceo Jeff Bezos Says The New Commitment To Make In India Amazon 'S 1 Billion Project Is The Latest Move From The Company . The Company Says The Investment Could Be Worth More Than Unk Billion .The Company Says It Is A Incremental Of The Country 'S Financial Sector .

Fig 3.1 Example for Proposed System

3.2 Scope

Extracting data from sources is a standard process in systematic review (SR) development. However, the data extraction process still relies too much on manual efforts which is slow, costly, and subject to human error. In this study, we develop summarization system aimed at enhancing productivity and reducing errors in the traditional data extraction process. Information extraction is the process of extracting specific (pre-specified) information from textual sources. One of the most trivial examples is when your email extracts only the data from the message for you to add in your Calendar. Other free-flowing textual sources from which information extraction can distill structured information are legal acts, medical records, social media interactions and streams, online news, government documents, corporate reports and more. Gathering detailed structured data from texts, information extraction enables:

- The automation of tasks such as smart content classification, integrated search, management and delivery;
- Data-driven activities such as mining for patterns and trends, uncovering hidden relationships, etc.

Apart from these it has a greater scope and spreads its roots across several applications, few of which are discussed below.

3.3. Purpose

The demand and scope for natural language processing tasks is growing at each passing day. A much deeper and a higher level of context, meaning and brief information must be presented that is simple, concise, and understandable but at the same time it must be highly descriptive and have a high level of information. Such information finds its scope through various applications from general public, information retrieval systems, automation etc. We aim to solve this problem by providing a system that is able to provide reduced context-full information along with descriptive parameters from the text. Our goals are that

- These summaries will be as important as possible in the aspect of the texts intention.
- Supplying the user a smooth and clear and simple interface.
- Maintaining the descriptiveness of information to be presented.
- Keeping a smooth transition of processes.

4. IMPLEMENTATION

The implementation consists of 4 phases, which include:

- Extractive
- Abstractive
- Topic
- Entity

All these phases are implemented using the libraries which are supported by the Python language.

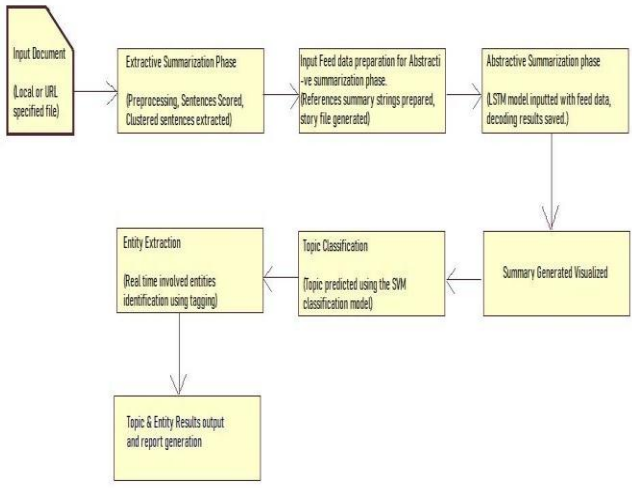

Input Document
(Local or URL specified file)
Extractive Summarization Phase
(Preprocessing, Sentences Scored, Clustered sentences extracted)
Input Feed data preparation for Abstracti-ve summarization phase.
(References summary strings prepared, story file generated)
Abstractive Summarization phase
(LSTM model inputted with feed data, decoding results saved.)
Summary Generated Visualized
Topic Classification
(Topic predicted using the SVM classification model)
Entity Extraction
(Real time involved entities identification using tagging)
Topic & Entity Results output and report generation

Fig 4.1: Architecture of the proposed system.

The Architecture of the proposed system is as shown in the following fig 4.1. As seen in the fig the entire system is categorized into a set of phases, where in each phase is responsible for handling a particular task. The input is text document .doc or a plain text file .txt or it could even be a web document specified through the URL. Each phase is explained into the following subsections.

4.1 Extractive Method:

The system uses the NLTK python module for achieving the NLP (Natural Language Processing) tasks. The NLTK provides tokenization for sentences and words, stop words and stemming of words to root forms. The Library also provides POS (Parts of Speech) tagger for identification of parts of speech in the data. Beautiful Soup module is used for HTML parsing. The pyFPDF is used for flushing the output to the pdf document. The communication with the HTML source document on web is provided by the urllib module. It consists of 3 modules

Pre-process Module:

This module has the duty of filtering the input data through the processes such as removal of stop words, reduction of words to root forms, formatting of spaces and unrecognized symbols through the pre-process class. Then this module consists of the Tokenization class that performs the splitting of the data into sentences and words.

Feature Extraction Module:

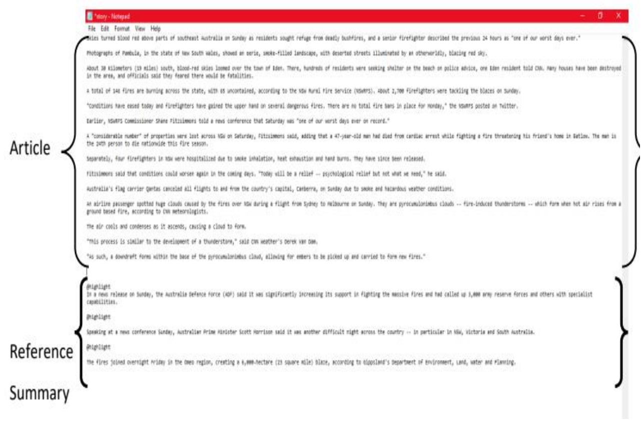
When the input data to an algorithm is too large to be processed and it is suspected to be redundant then it can be transformed into a reduced set of features (also named a feature vector). Determining a subset of the initial features is called feature selection. The core module for the application is calculating the sentence score and assigning the weights to the sentences. This process is done in Sentence Scoring class.

Clustering Module

K-means clustering is one of the simplest and popular unsupervised machine learning algorithms. Typically, unsupervised algorithms make inferences from datasets using only input vectors without referring to known, or labeled, outcomes. This module clusters the weighted sentences to form a cluster of major sentences to be chosen for the summary. After extractive process is completed, the output document is used for generating input-feed data.

Generation of Input-Feed data

This is an intermediate phase where after generation of reference summary is done through the extractive phase, the extracted summary is appended to the original article to form a file called a story file. This file consists of the original article and the reference summary strings separated by a '@highlight' tag. This file is tokenized and converted to a binary (.bin) file to be inputted to the abstractive phase. The input feed data is generated using Stanford core NLP package. In this binary file the article is kept under <article> tag and the reference summary is kept inn<abstract> tag. An example of the generated story file is shown in the following fig 4.2.



```
Skies turned blood red above parts of southeast Australia on Sunday as residents sought refuge from deadly bushfires, and a senior firefighter described the previous 24 hours as "one of our worst days ever."

Photographs of Pambula, in the state of New South Wales, showed an eerie, smoke-filled landscape, with deserted streets illuminated by an otherworldly, blazing red sky.

About 30 kilometers (19 miles) south, blood-red skies loomed over the town of Eden. There, hundreds of residents were seeking shelter on the beach on police advice, one Eden resident told CNN. Many houses have been destroyed in the area, and officials said they feared there would be fatalities.

A total of 146 fires are burning across the state, with 65 uncontained, according to the NSW Rural Fire Service (NSWRFS). About 2,700 firefighters were tackling the blazes on Sunday.

"Conditions have eased today and firefighters have gained the upper hand on several dangerous fires. There are no total fire bans in place for Monday," the NSWRFS posted on Twitter.

Earlier, NSWRFS Commissioner Shane Fitzsimmons told a news conference that Saturday was "one of our worst days ever on record."

A "considerable number" of properties were lost across NSW on Saturday, Fitzsimmons said, adding that a 47-year-old man had died from cardiac arrest while fighting a fire threatening his friend's home in Batlow. The man is the 24th person to die nationwide this fire season.

Separately, four firefighters in NSW were hospitalized due to smoke inhalation, heat exhaustion and hand burns. They have since been released.

Fitzsimmons said that conditions could worsen again in the coming days. "Today will be a relief -- psychological relief but not what we need," he said.

Australia's flag carrier Qantas canceled all flights to and from the country's capital, Canberra, on Sunday due to smoke and hazardous weather conditions.

An airline passenger spotted huge clouds caused by the fires over NSW during a flight from Sydney to Melbourne on Sunday. They are pyrocumulonimbus clouds -- fire-induced thunderstorms -- which form when hot air rises from a ground based fire, according to CNN meteorologists.

The air cools and condenses as it ascends, causing a cloud to form.

"This process is similar to the development of a thunderstorm," said CNN Weather's Derek Van Dam.

"As such, a downdraft forms within the base of the pyrocumulonimbus cloud, allowing for embers to be picked up and carried to form new fires."

@highlight
In a news release on Sunday, the Australia Defence Force (ADF) said it was significantly increasing its support in fighting the massive fires and had called up 3,000 army reserve forces and others with specialist capabilities.

@highlight

Speaking at a news conference Sunday, Australian Prime Minister Scott Morrison said it was another difficult night across the country -- in particular in NSW, Victoria and South Australia.

@highlight

The fires joined overnight Friday in the Omeo region, creating a 6,000-hectare (23 square mile) blaze, according to Gippsland's Department of Environment, Land, Water and Planning.
```

Fig 4.2: generated story file

4.2 Abstractive summarization phase

The generated Input-Feed data is used in the abstractive phase. This phase is responsible for handling the summarization task based on abstractive approach. This approach uses a trained RNN (Recursive Neural Network) that generates the summary tokens forming the final summary. The RNN consists of encoder and decoder sequences which predict the summary tokens. This approach makes use of deep learning i.e. it uses the 'RNN', the Recursive neural networks called LSTM 'Long Short Term Memory' networks sequence to sequence. This consists of two parts

- The Encoder
- The decoder

4.2.1 How it works:

The Encoder part is responsible for accepting the input sequence tokens and for generating hidden states, for generating a hidden state it uses previous saved state information hence it is called Long short-term memory as it saves and uses previous information. Also, in doing so each encoder state must remove unwanted information and only store required tokens for this it uses internal logical gates which throw away unwanted information and keep only the valid ones. These Internal gates are

- Input gate – this gate takes an input sequence token and is transformed or scaled to a fixed value.
- Forget gate - this gate is responsible for deciding which information to throw out.
- Output gate – this gate decides what the next hidden state is.

To transform the input sequences to a scaled value we use two functions tanh() that scales between -1 to 1 and sigmoid() that scales between 0 to 1. The Decoder works in opposite way to encoder, producing the outputs from hidden states.

4.2.2 How Text summarization occurs in LSTM sequence to sequence model:

We use a pointer generator with attention mechanism with the basic LSTM model because the basic model suffers from producing OOV (Out of vocabulary words) and also fails to provide any facts accurately.The tokens of the article W_i are fed one-by-one into the encoder (a single-layer bidirectional LSTM), producing a sequence of encoder hidden states hi. On each step t, the decoder (a single-layer unidirectional LSTM) receives the word embedding of the previous word (while training, this is the previous word of the reference summary; at test time it is the previous word emitted by the decoder), and has decoder state st .Next an attention mechanism is used to produce the attention distribution of the input tokens; the attention distribution can be viewed as a probability distribution over the source words, that tells the decoder where to look to produce the next word. Next, the attention distribution is used to produce a weighted sum of the encoder hidden states, known as the context vector. The context vector is the added to the decoder states to produce the final probability distribution of vocabulary P_{vocab} during the training process. Then in testing process this probability distribution is used to predict words.Next The generation probability $P_{gen} \in [0,1]$ for timestep t is calculated from the context vector, the decoder state s and the decoder input x. This P_{gen} acts as switch which tells whether to choose the word from the P_{vocab} probability distribution or to copy the word from the source text itself. To prevent repetition coverage mechanism is used where a coverage vector is used that is a sum of all attention distributions of previous decoder time steps which tells the degree of usage a word has received thereby preventing using that word again and again. The Entire process is depicted in the following fig 4.3.

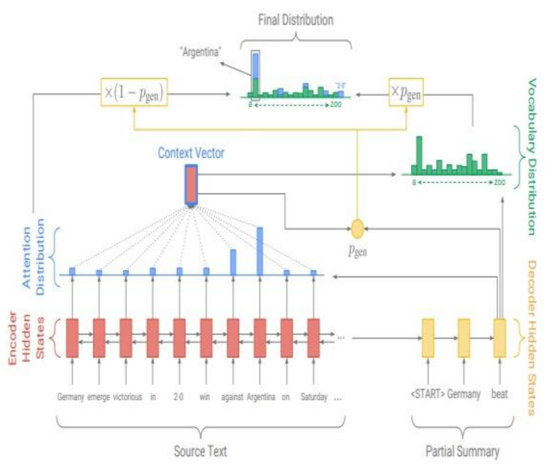


Fig 4.3: Abstractive Summarization Process

4.2.3 Data Set for abstractive summarization and training model:

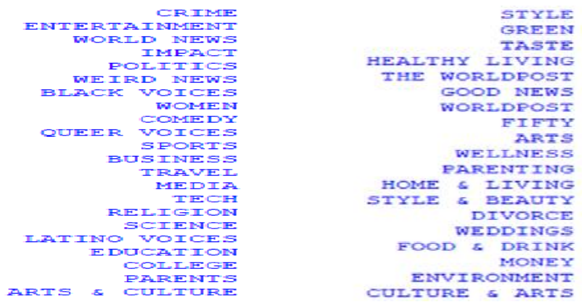
The abstractive phase of the system uses a RNN (Recursive Neural Network) based LSTM (Long short term memory) network model. This model is trained on the 'CNN/Dailymail News article dataset that is procured from the open source repository. The CNN/Dailymail dataset consists of raw news articles and summaries for about 50,000 articles. To processes the raw files and tokenize them an open source java package is used called 'Stanford_core_NLP'. The environment variables are set in system and the classpath is configured. The Stamfor_core_NLP package is executed onto the directory that consists of the raw files. This tokenizes the articles and then it is converted into binary files '.bin'. Each binary file is chunked by chunking about thousand articles and their summaries per one binary chunk file. This result into creation of binary chunked files used for training the model and a Vocabulary file.

4.3 Topic classification phase

In the topic modelling phase of the system, it aims at predicting a more generalized topic that describes the article and its meaning. For this task SVM (Support Vector Machine) classification is used. The SVM classifier is trained on a processed dataset of 'News category' from the kaggle repository. The dataset consists of about 41 topics ranging from Crime, Business, and Entertainment etc. In this phase, the generated summary of the document is classified to predict the generic topic that describes the input document. For our system, we selected 41 topics which are categorized from 37,000 data sets. The document is classified to a particular class of category such as CRIME, ENTERTAINMENT, and POLITICS etc. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting).

4.3.1 How SVM classification works:

In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well an SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible which can be observed in the following figure 4.4 and figure 4.5 gives an example of hyper-plane.



CRIME	STYLE
ENTERTAINMENT	GREEN
WORLD NEWS	TASTE
IMPACT	HEALTHY LIVING
POLITICS	THE WORLDPOST
WEIRD NEWS	GOOD NEWS
BLACK VOICES	WORLDPOST
WOMEN	FIFTY
COMEDY	ARTS
QUEER VOICES	WELLNESS
SPORTS	PARENTING
BUSINESS	HOME & LIVING
TRAVEL	STYLE & BEAUTY
MEDIA	DIVORCE
TECH	WEDDINGS
RELIGION	FOOD & DRINK
SCIENCE	MONEY
LATINO VOICES	ENVIRONMENT
EDUCATION	CULTURE & ARTS
COLLEGE	
PARENTS	
ARTS & CULTURE	

Fig 4.4: Topic Classes (41 Topics)

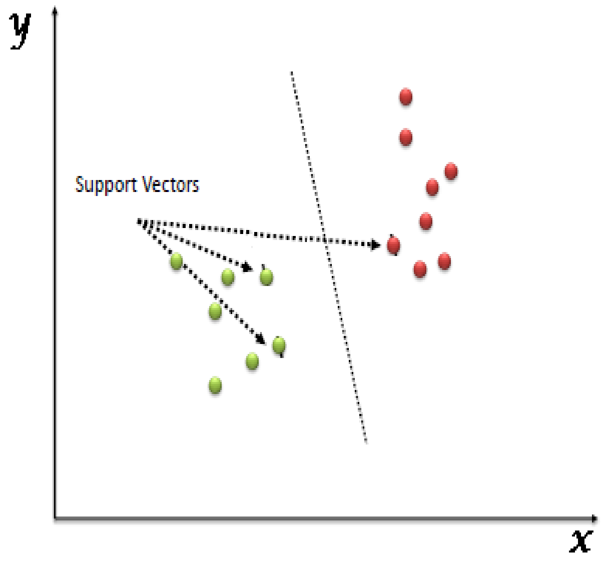


Fig 4.5: Hyper-plane

4.3.2 Dataset for SVM topic modeling and training:

The 'News category dataset' from Kaggle is used for training the classification model. This is a JSON file which consists of several news articles with short descriptions, headlines and categories. This consists of about 40 topic categories. Primarily, the JSON file is downloaded and the file is processed to remove all unwanted attributes and keep only the two important attributes which are – the short description of news and the topic category.

After that, a short description is pre-processed to remove stop words and lemmatize the words. Then the topic categories are assigned an integer id ranging from 0 to 40. These attributes are then saved to .CSV file In the next step, the topic categories and their ids are saved in a dictionary structure and saved locally using the pickle python library. Next The SVM classifier is built using the Scikit learn python library. The dataset is firstly vectorized using the tfidf vectorizer because string data cannot be directly used to train the model. The model is trained on Colab and saved using the python pickle library.

Below figure 4.6 is the example of how raw JSON data file is being converted into processed dataset using various mentioned pre-processor techniques.

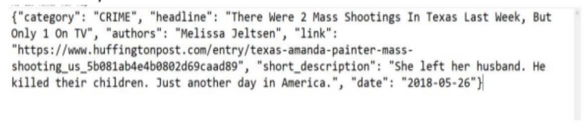
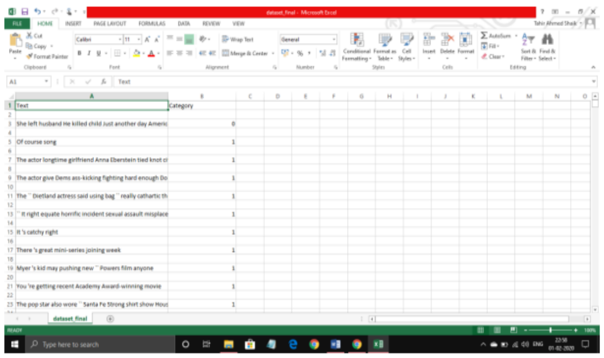

{"category": "CRIME", "headline": "There Were 2 Mass Shootings In Texas Last Week, But Only 1 On TV", "authors": "Melissa Jeltsen", "link": "https://www.huffingtonpost.com/entry/texas-amanda-painter-mass-shooting_us_5b081ab4e4b0802d69caad89", "short_description": "She left her husband. He killed their children. Just another day in America.", "date": "2018-05-26"}

Fig : Raw JSON Data File



	A	B
1	Text	Category
2		
3	She left husband He killed child Just another day Americ	0
4		
5	Of course song	1
6		
7	The actor longtime girlfriend Anna Eberstein tied knot ci	1
8		
9	The actor give Dems ass-kicking fighting hard enough Do	1
10		
11	The `` Dietland actress said using bag `` really cathartic th	1
12		
13	`` It right equate horrific incident sexual assault misplace	1
14		
15	It 's catchy right	1
16		
17	There 's great mini-series joining week	1
18		
19	Myer 's kid may pushing new `` Powers film anyone	1
20		
21	You 're getting recent Academy Award-winning movie	1
22		
23	The pop star also wore `` Santa Fe Strong shirt show Hous	1

Fig : Processed dataset.

Fig 4.6: Processed dataset from Raw JSON data file

4.4 Entity recognition phase

This phase carries out the task of Named Entity Recognition (NER), where the tokenized text is tagged and annotated with a particular entity tag such as PEOPLE, PLACE, and ORGANIZATION etc. In this phase, the primary task is that it first takes the abstract output text, pre-processes the text(removing stopwords, root words) then using the nltk library the processed text is tagged that produces the nouns, adjective etc. words, These tagged tokens are then input to the spacy model that attaches an entity category to each token. Then Tokens that represent people, organization, location, products, money are chosen as the output entities. The tokens tagged with categories of Person, GPE (Location), Organization, Products and Money is extracted.

- SpaCy: spaCy is an open-source software library for advanced natural language processing, written in the programming languages Python and Cython. It features convolutional neural network models for part-of-speech tagging, dependency parsing and named entity recognition, as well as API improvements around training and updating models, and constructing custom processing pipelines. spaCy also supports deep learning workflows that allow connecting statistical models trained by popular machine learning libraries like TensorFlow, Keras, Scikit-learn or PyTorch. spaCy's machine learning library, Thinc, is also available as a separate open-source Python library.

- Final Output: A final report in a PDF format is generated by the user at the final phase which describes the summary, classified topic and all the real time entities present in the text. pyPDF library is used for generating the PDF file. This file can also be used to perform TTS (Text to speech) tasks. For speech synthesis Pytts library is used to read out the final summary of text document

5. INTERFACE

This chapter provides the graphical user interface of the developed application and an overview of how this system is providing the output. Figure 5.1 depicts the main screen of this application following this sample text document (figure 5.2) is been considered as input for which the summary is being generated shown in the figure 5.3. Topic modeling and entity extraction results from the provided input are observed in the figure 5.4.

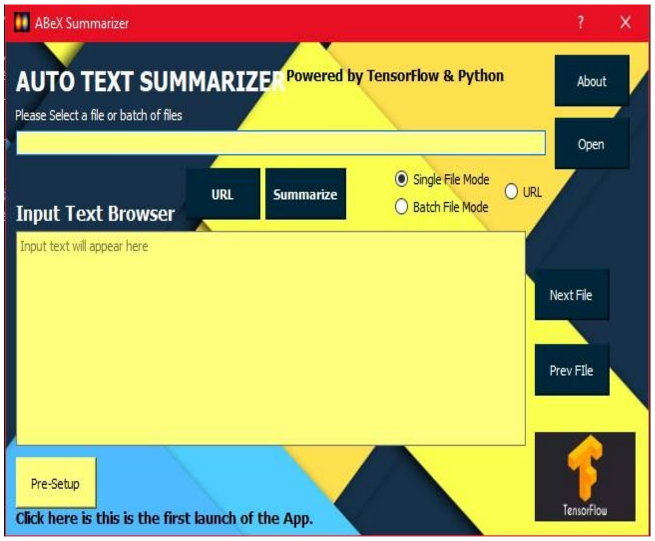
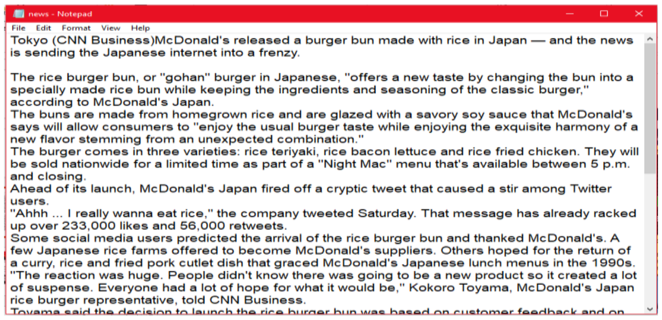

ABeX Summarizer
AUTO TEXT SUMMARIZER
Powered by TensorFlow & Python
About
Please Select a file or batch of files
Open
URL
Summarize
Single File Mode
Batch File Mode
URL
Input Text Browser
Input text will appear here
Next File
Prev FIle
Pre-Setup
Click here is this is the first launch of the App.
TensorFlow

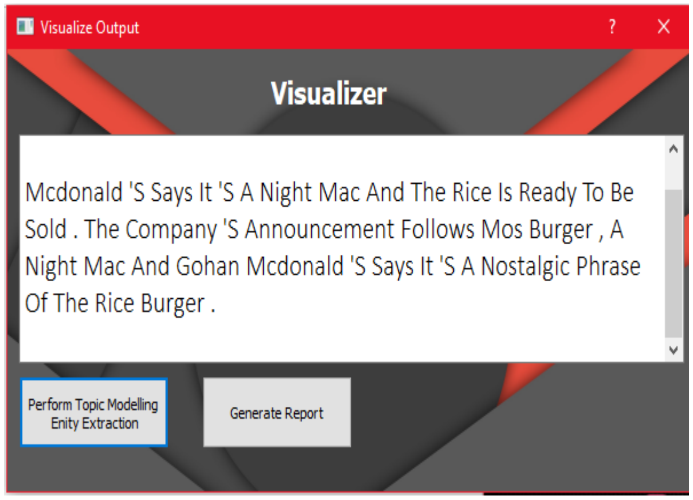
Fig 5.1 Main Screen of the Application


news - Notepad

Tokyo (CNN Business)McDonald's released a burger bun made with rice in Japan — and the news is sending the Japanese internet into a frenzy.

The rice burger bun, or "gohan" burger in Japanese, "offers a new taste by changing the bun into a specially made rice bun while keeping the ingredients and seasoning of the classic burger," according to McDonald's Japan.
The buns are made from homegrown rice and are glazed with a savory soy sauce that McDonald's says will allow consumers to "enjoy the usual burger taste while enjoying the exquisite harmony of a new flavor stemming from an unexpected combination."
The burger comes in three varieties: rice teriyaki, rice bacon lettuce and rice fried chicken. They will be sold nationwide for a limited time as part of a "Night Mac" menu that's available between 5 p.m. and closing.
Ahead of its launch, McDonald's Japan fired off a cryptic tweet that caused a stir among Twitter users.
"Ahhh ... I really wanna eat rice," the company tweeted Saturday. That message has already racked up over 233,000 likes and 56,000 retweets.
Some social media users predicted the arrival of the rice burger bun and thanked McDonald's. A few Japanese rice farms offered to become McDonald's suppliers. Others hoped for the return of a curry, rice and fried pork cutlet dish that graced McDonald's Japanese lunch menus in the 1990s.
"The reaction was huge. People didn't know there was going to be a new product so it created a lot of suspense. Everyone had a lot of hope for what it would be," Kokoro Toyama, McDonald's Japan rice burger representative, told CNN Business.
Toyama said the decision to launch the rice burger bun was based on customer feedback and on

Fig 5.2 A Sample plain text input document (.txt)



Mcdonald 'S Says It 'S A Night Mac And The Rice Is Ready To Be Sold . The Company 'S Announcement Follows Mos Burger , A Night Mac And Gohan Mcdonald 'S Says It 'S A Nostalgic Phrase Of The Rice Burger .

Fig 5.3 Summary Output For the input document shown in fig 5.2

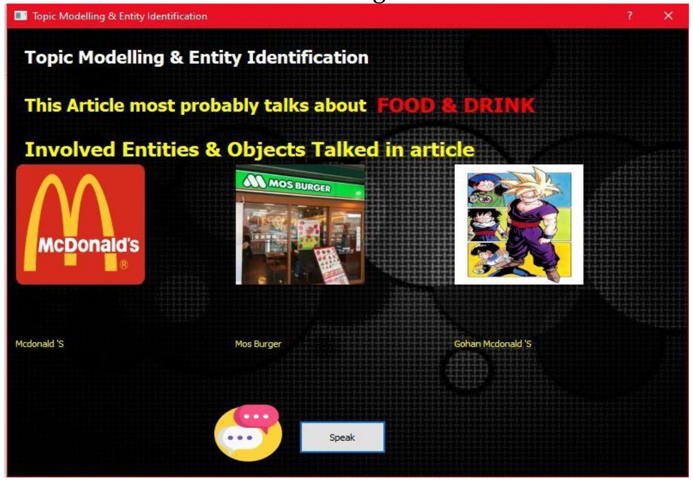

Topic Modelling & Entity Identification
Topic Modelling & Entity Identification
This Article most probably talks about FOOD & DRINK
Involved Entities & Objects Talked in article
McDonald's
MOS BURGER
Mcdonald 'S
Mos Burger
Gohan Mcdonald 'S
Speak

Fig 5.4 Topic and Entity results for input document shown in fig 5.3

6. Results

6.1 Testing

The Testing of the project is done and reported by using the TestLink platform. Testlink is an open source test management framework that provides creation of test cases, test plans and also generation of test reports. The project is tested for functionality and specifications. The tests are carried out for the following cases: (10 Cases)

- Unreachable URL (URL that cannot be opened 404 errors).
- Incomplete document input (providing a file that is having missing data and is incomplete.)
- Wrong file mode (Opening a file in the wrong file mode)
- Wrong file extension(Opening a file other the .txt)
- Inputting encrypted document.
- NULL or empty input.
- Opening files present on different folders and directories.
- Testing on Highly styled websites i.e websites that are more advanced in terms of looks and styles.
- Cancelling while opening a file.
- Empty text document.

In all these scenarios, the results obtained from the testing were accurate according to the type of input document provided and there was no discrepancy shown on the results.

6.2 Observations

The performance of the system, its various aspects of the system have been recorded and analyzed and shown below through plots.

6.2.1 Summarizer Accuracy

The below figure 6.1 describes the accuracy of the summarizer with respect to the no of input lines of text. As seen the accuracy increases as the input size increases. It is observed that the accuracy increases as the input size increases. The greater number of lines or words the higher the accuracy of the summary obtained. As the system analyses each and every word and predicts what the upcoming word might be, if the document has fewer number of lines or words then the resultant summary obtained might not be accurate or show few discrepancies in the output summary.

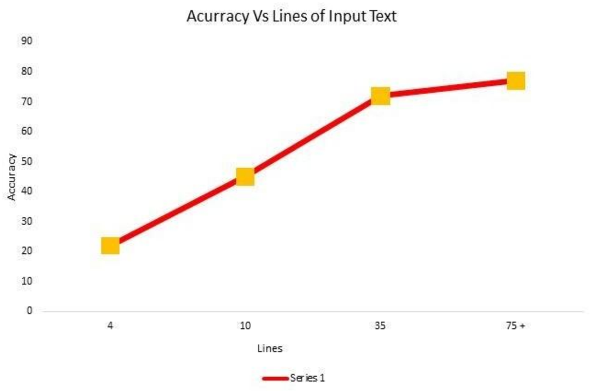


Fig 6.1: Accuracy w.r.t No of lines of input

The below figure 6.2 implies compression ratio of the summarizer which describes the rate of lines to which the summarizer reduces the input text. Generating the accurate summary is not the only vital task, but the summary must have fewer lines so that it would be stress-free for the end-user. So, we trained the system to provide fewer lines as well as generating accurate summary to the end-user.

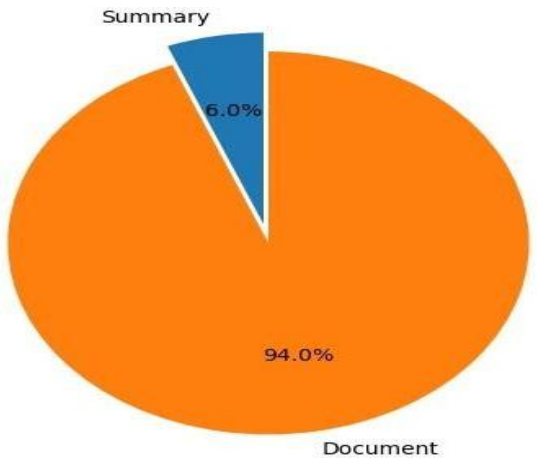


Fig 6.2: Summary Compression Ratio

We have collected different articles from a variety of news channels and compared the accuracy of summary generation for each that is as shown in the figure 6.3. From the given results, CNN news channel provides more accurate summary as most of articles were collected from CNN news channel

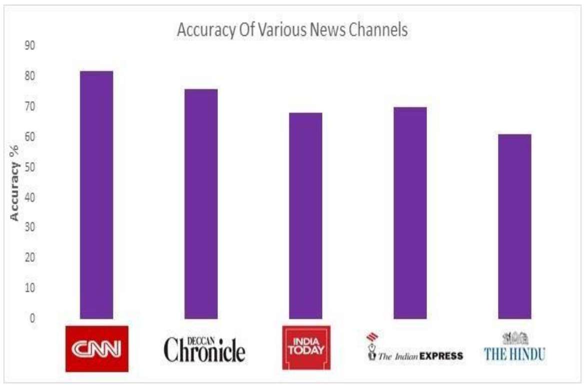


Fig 6.3: Summary Accuracy of Various News Channels

6.2.2 Topic Classifier Accuracy and Metrics

The below figure 6.4 is for accuracy of topic modeling in which the accuracy of topic classifier is measured with increasing the dataset size. The accuracy increases with the increase in dataset. For our system we've mostly opted for 37,000 data sets.

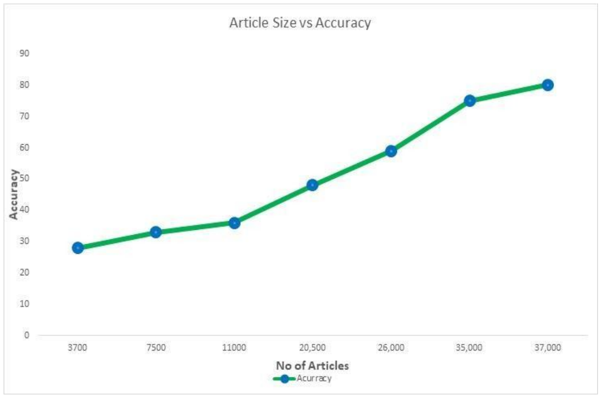


Fig 6.4 Topic Classifier Accuracy w.r.t Dataset Size

The below figure 6.5 shows the overall topic distribution in the news category dataset. It contains 41 topics which is descried by each bar in the graph that shows the no of articles for that topic in the dataset.

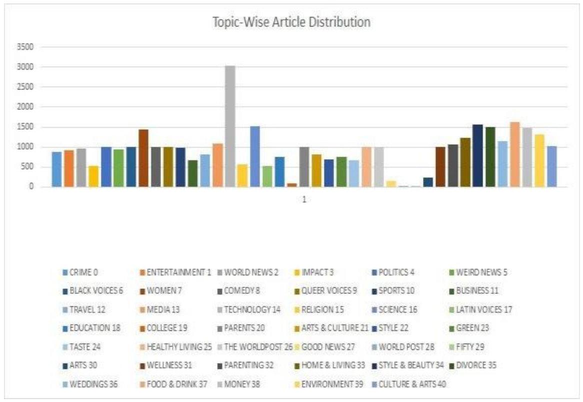


Fig 6.5 Topic – Wise Article distribution in Topic dataset

The last below figure 6.6 shows the accuracy per topic for different types of articles where a set of 5 topics are chosen and three types of articles sources (News, Reports and Wiki) are chosen and accuracy for each is plotted in the graph.

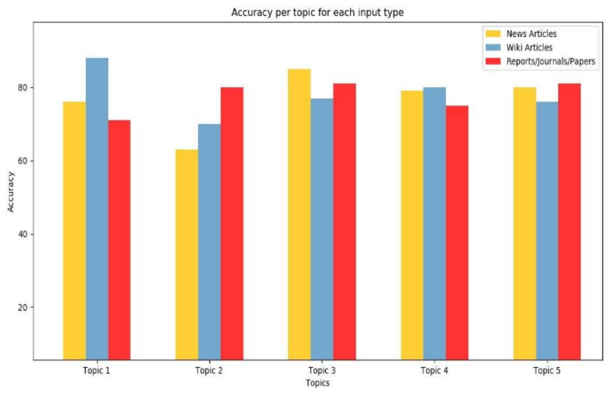


Fig 6.6 Accuracy of classifier per topics and article type

7. CONCLUSION

The rate of information growth due to the World Wide Web has called for a need to develop efficient and accurate summarization and information extraction systems. Although research on summarization started about 50 years ago, there is still a long trail to walk in this field. Over time, attention has drifted from summarizing scientific articles to news articles, electronic mail messages, advertisements, and blogs. Both abstractive and extractive approaches have been attempted, depending on the application at hand. Usually, abstractive summarization requires heavy machinery for language generation and is difficult to replicate or extend to broader domains. In contrast, simple extraction of sentences has produced satisfactory results in large-scale applications. The system achieves a part in combining the efforts of both the approaches and at the same time appends a higher level of information extraction through topic classification and entity identification. The system manages to achieve its defined objectives. Although a lot of scope still remains to be achieved in this immortal field.

8. FUTURE ENHANCEMENTS

In this section, we mention some of the possible future extensions of this research. The rate at which the information is growing is tremendous. Hence it is very important to build a multilingual summarization system and this research could be a stepping stone towards achieving that goal provided there is availability of online lexical databases in other languages. The work presented by the thesis can also be applicable to multi document summarization by using minimal extensions. In future work, new metrics can be investigated which can be used in automatic evaluation environment to measure the overall quality such as grammar, readability, prominence and relativeness. Research in summarization continues to enhance the diversity and information richness, and strive to produce coherent and focused answers to users information need.

9. References

[1] J. N. Madhuri and R. Ganesh Kumar, "Extractive Text Summarization Using Sentence Ranking," 2019 International Conference on Data Science and Communication (IconDSC), Bangalore, India, 2019, pp. 1-3.

[2] Abigail See, Peter J. Liu, Christopher D. Manning (2017), Get to the point: Summarization with pointer generator networks, doi.org/10.18653/v1/P17-1099,55th meeting of the Association for Computational Linguistics, 1073-1083.

[3] Freek Boutkan, Jorn Ranzijn, David Rau, Eelco van der Wel (2019), Point-less: More Abstractive Summarization with Pointer-Generator Networks, arXiv:1905.01975v1[cs.CL].

[4] Soumick Chatterjee, Pramod George Jose , Debabrata Datta (2019), Text Classification Using SVM Enhanced by Multithreading and CUDA, I.J. Modern Education and Computer Science, 2019, 1, 11-23.

[5] A.Anantharaman, A. Jadiya, C. T. S. Siri, B. N. Adikar and B. Mohan, "Performance Evaluation of Topic Modeling Algorithms for Text Classification," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2019, pp. 704-708.