Project Report for
CE903

Automatic Title Generation Using Encoder-Decoder Models

Team-9

Group Members: Karan Bhatt (2112102), Dhaval Patel (2111480), Ashish Gajera (2111045), Vatsal Trivedi (2111154), Vipul Barot (2111824), Niaz Muhammad Umair (2111659),

Supervisor: Dr. Yu, Junto

Table of Content

1. Introduction

Title is key highlight point of any article. In a current world search the article or detail summary of any topic we need to serch it by its keyword. Our aim to generate title of article which is summury of article and user can serch anything by its title's keyword. So a good title create good impression over user as well it can key word to search that article.

Here we are going to perfrom title generation from the article and this task perfrom with the help of encoding/decoding method. In this task encoding/decoding method help us to translate or encode the graphical, symbol, unknown language(except English) and it can be analyzed in digital form. At the end of analysis we decode that digital signal for our output which is in its original form.

Title generation is a task of producing a concise and fluent summery while preserving key information content and meaning of article. There is two approach of text summarization is extractive text summarization and second is abstractive text summarization. In the abstractive summarization we summarize the important key sentences and important notes of article while in extractive summarization we summarize key word that are relevant to article and make a meaning full sentence for that given article text input.

In this article we are going to discuss about extractive text summarization method which is used for title generation. Machine learning is the method which is used for many data analysis task and we used ML for our text analysis task. We use ML to check which words have higher weightage for from the given text artical. Text summerizaton is Netural language processing(NLP) task and it perofm with the help of ML.

We design a system which give extractive text summarization for Title generation from the given text, for that we created one web application with the help of Python and Flask. We train our given data of arxiv.org articals and make a dataset for our machine learning process. Web application platform provide input which is transfer to python using Flask and we send output title summery to the Flask to display on user platform which is web application.

1.1. Project Objective:

Objective of doing this project is to generate title from the article text which is entered by user. We created Unspervised learning method based application for the user to generate title for their written article which is key word oriented and we can use that title for publication so world can easly find that article using that article. We create our article using machine learning method, in which our main task in to find key word for that text and generate title using that relevant key word of article so while searching article from the cloud or any storage we can get it easly.

Title of article is unique identity and we are generating this title using machinn learning so we get most frequent word used in article as our title output so it make user comfortable and flexible to choose their title using this web application.

There are many of the text articles available online & therefore a problem of searching for relevant documents in the number of available documents, and extracting relevant information from them. To solve the two problems above, automated text summaries are very much needed. Text summarization is the process of identifying the most important information in a document or set of related documents and compressing them into a short version that retains their full meanings. Before going to the summary of the text, first, we need to know what the summary is. A summary is a text produced in one or more texts, conveying important information to the original text, and is in short form. The goal of automated text abstraction is to turn the source text into a shorter version with semantics

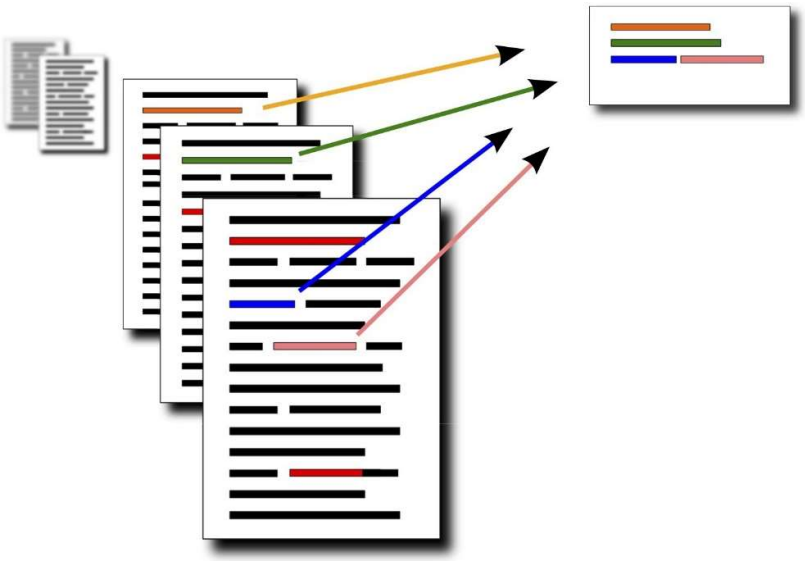

Fig.1 Simple graphical view of or task (Reference: https://www.wordtune.com/)

As shown in above image we use different key word from the article and use it for our text Summerization task to get Title of article which is entered. User can define title by self but machine learning is a new way to get automatic title where we use machine learning to decide the key word and it generate title by learning from large dataset where user can not think about this much large article dataset. We train our model in such way that it get better title for article using our system. it is unique output generation from this system so we don't not have any issue related similar title and copyright issue for user and it is the best way for user to get title for their article. We use encoding/decoding method which consider special charters also so no need to use any translation method and then go for title generation we can fullfil our aim of Automatic title generation using this web application.

1.2. Methodology

Text data classification doing with manual reading is time consuming and tedious task. For that we need automatic process which read our article and make summery of that text which words are used mostly and prdict a new title for that article. An automatic system created for text classification which perfrom over Python

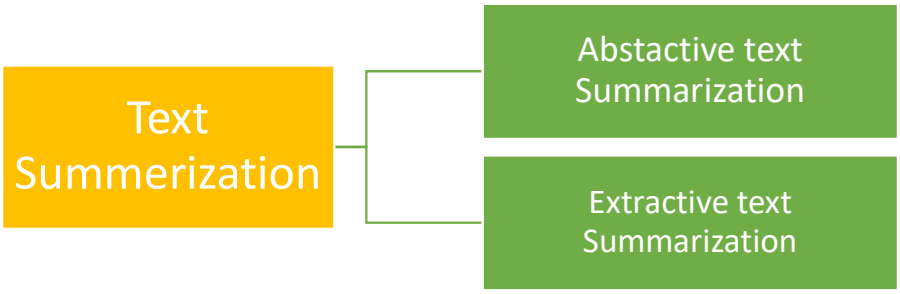

Fig.2 Text Summerization Methods (Reference: self work)

Text classification is the method of searching keyword from the text and categories.Python is preferred platform where this task can be perfrom steadily using its open source libraries Scikit-learn, NLTK, SpaCy, Keras, Tenserflow, Pytorch are popular libraries of python which can use for our Title generation task, from this we are going to use Pytoch and SimpleT5 for classification of text and train the dataset and make good PKL.

Open source are best because its free and flexible to use them and we use all open source platform to perfrom our task and it make user easy and reliable. There is a website which store all scientific/statistic/economical paper in digital form and we use this website's paper to generate title of that article and there is open data source of this arxiv all paper which is JSON file. It is a line by line text file where all the article data is store of the arxiv.org and it can be download from Kaggle.com using this link [arXiv Dataset | Kaggle] which is approx 3 Gigabit information

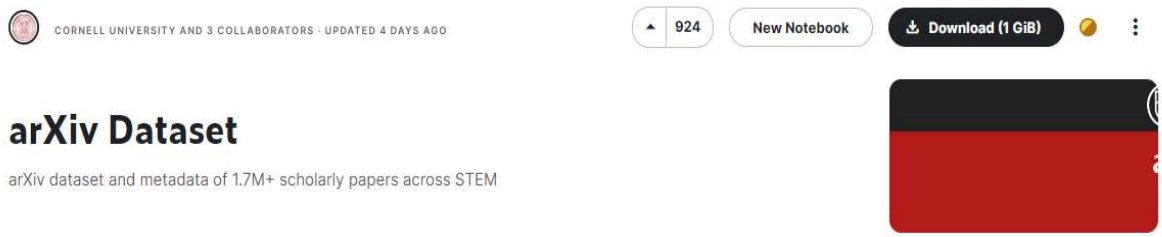

Fig.3 arxiv dataset by cornell university on kaggle(Reference: [arXiv Dataset | Kaggle])

ArXiv On Kaggle
Metadata

- id: ArXiv ID (can be used to access the paper, see below)
- submitter: Who submitted the paper
- authors: Authors of the paper
- title: Title of the paper
- comments: Additional info, such as number of pages and figures
- journal-ref: Information about the journal the paper was published in
- doi: [https://www.doi.org](Digital Object Identifier)
- abstract: The abstract of the paper
- categories: Categories / tags in the ArXiv system
- versions: A version history

We use encoding/decoding method to train this dataset for our title generation process. We train this dataset for our text classification process using SimplT5 and Pytorch libraries. Once you download the dataset we need to clean that data befor we use it further in our task. For that we need to remove all null, not categorized characters and end stop words to get clean data. Then it is train and this train dataset is used for our text classification.

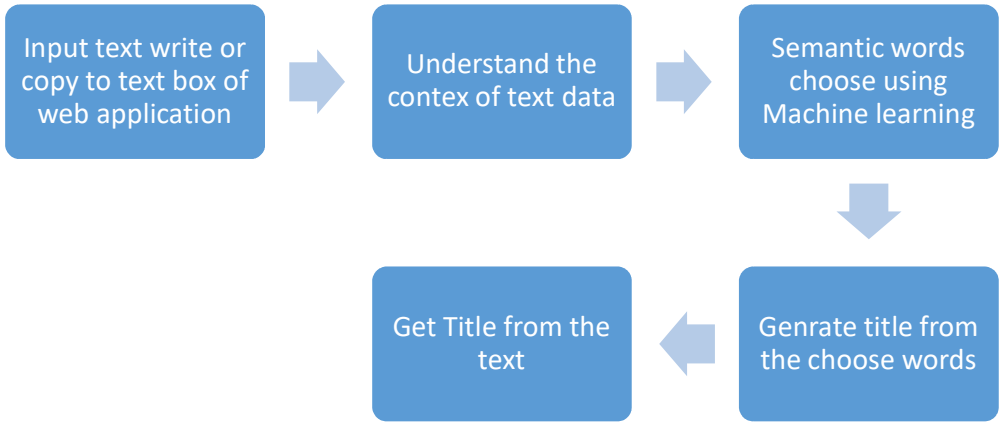

Fig.4 Simple task flow to get output (Reference: Self work)

1.3. Software platform and tools used for design and testing

Our aim to make system Easy and cost effective so it can be user friendly and that's why we use open sources to design the system. our software requirement is as bellow

Google Colab:

Colab is a open platform where you can create your code and you can get online GPU for your process it is cheaper method to do our project and it is most used platform for this type of application where you required GPU. Second benefit is that we need a platform where we can store our train dataset safely and Google colab can ealy mount you google drive to its programming platform. Storage problem can also solve using Colab so we choose Google colab for our main platform to run code.

Flask:

Our aim is to create web application and we required on platform to devlop and web application. Flask is the open source and free to use in python to devlop web application. Flask have large number of libraries that can perfrom a better for our web application. Flask used to fetch the data from the web and it also send the data to web to display. It provide low level support for web application where you can launch you application using local host or using web server. Here we use web server to launch our application it help to reduce the use of GPU process time and run fast.

Ngrock web server:

To make web application, web server is must required. That web server is used to perfrom the task over it. We use temporary web server to launch our web application using ngrock. It's the webserver provider that provide free web server for application try and error check. It give a token which used in our app.py to use its web space.

HTML/CSS:

Web application required web page which design in visual studio and it is simple html file. CSS is used to give graphical representation to our web page. HTML file used in our Flask to launch web page for application.Software requirement for Visual studio is as below.

- Operating system require-

Windows 10 version 1703 or higher: Home, Professional, Education, and Enterprise (LTSC and S are not supported)

- Hardware requirement-

1.8 GHz or faster processor. Quad-core or better recommended and 2 GB of RAM; 8 GB of RAM recommended (2.5 GB minimum if running on a virtual machine)

- Additional requirement for software-
  - Administrator rights are required to install or update Visual Studio.
  - Refer to the Visual Studio Administrator Guide for additional considerations and guidance for how to install, deploy, update, and configure Visual Studio across an organization.
  - .NET Framework 4.5.2 or above is required to **install** Visual Studio. Visual Studio requires
  - .NET Framework 4.7.2 to run, and this will be installed during setup.
  - .NET Core has specific Windows prerequisites for Windows 8.1 and earlier.

JavaScript:

To run any web application need support of java which is required for brower compatibility requirement with system, we need to install above version of JDK8 of oracle for supporting environment. We also need JDK for visual studio software. It support visual studion tool to execute.

Bootstrap:

To support our HTML we need CSS and for that we need bootstarp. Bootsratp provide CSS fram work to our HTML web page. Bootstrap 5 is required for CSS design and Bootstrap 5 support all the web browser upto latest and it provide graphical supported platform for our web pages.

Selenium:

Python code need to be testedover selenium where we can test simple as well complex python code. In our project we use it for unit testing as well for automatic testing of python file. We use colab's inbuilt finction to check error on every step but need on software platform where we can check our final code is it run without query or not. For that we use selenium and use its assert function to check evey step of code.

2. System Design

2.1. Brief System requirement

This architecture requires Colab of Google which is open platform and web based GPU provider also. Colab provide Python programing support whee you can write code and run using web platform. We created our python coding over it and it is a ".IPYNB" file extension is essential for computing notebooks that can access by jupyter notebook also.

We required large space to upload our train dataset. So we use Google drive which provide 15GB free storage where we can store our dataset and it can easly connect with our Colab. To instruct the model it require pro version of google colab because it provides 2 GB RAM and also provide service of full time GPU scaling.Particularly this command is used to mount google drive ,once drive is mount your data will be visible in dataset.

Fig5. Command to mount our dataset form the drive to python platform of Colab(Reference: self work)

As previous discussion in point number 1.3 we required that much environment befor to start coding, testing and execution to run our web application. Need to istall bellow libraries for python first using command prompt.
Install this libraries first befor start python coding.

```
!pip install simplet5
!pip install flask_ngrok
!pip install pyngrok
!pip install numpy
!pip install pandas
```

Now we can use this libraries for our python code whivh provide arithmetic and logical support to our programming.

Need to know about

FLASK:
It is a one of the module of python ,which allows you to do framework and to develop application easily. it is more accessible for amateur developers.

NumPy:
It is the fundamental tool for scientific computing in python . it is basically used when to work with arrays and has function of working in algebra, Fourier transform, and matrices

Pandas:
It has a specific work of analyzing data .learning by reading . This package provides fast, flexible, and expressive data structure design to working with labeled data.

Simple T5 Machine learning model:
T5 is a Text-to-Text Transfer Transformer, is a Transformer based architecture that uses text-to-text. All the work - which includes translation, answering questions, and classification - is done as feeding the model text as input and training it to produce a specific target text. This allows the use of the same model, job loss, multiple parameters, etc. throughout our various set of tasks.

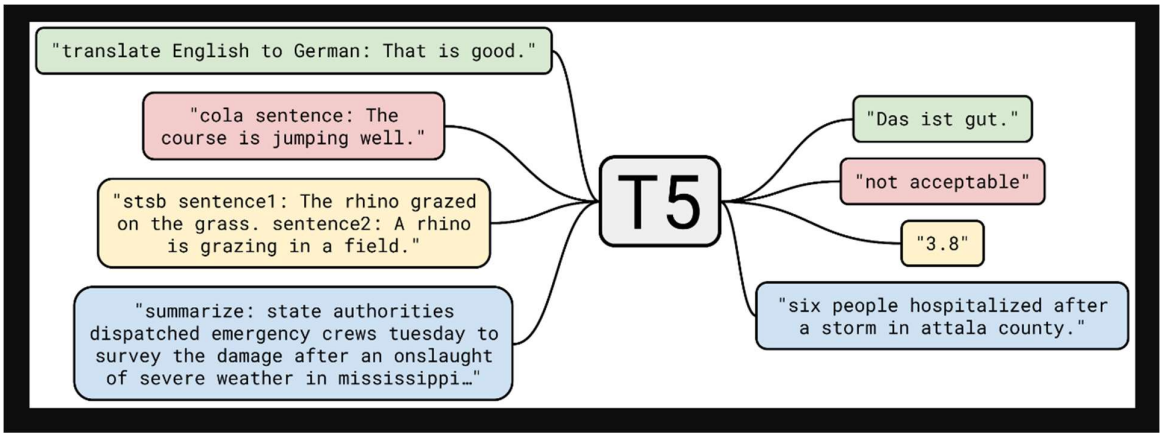

Fig.6 SimpleT5 library explain in graphical form (Reference: Jay Alammar blog on github & simpleT5 paper))

Ngrok:

It is useful to create secure tunnels to locally hosted applications using reverse proxy. It provide web server for web application launch. It can easly install in python as well. It provide token and this token is conform in python “app.py” which designed to launch web application using flask.

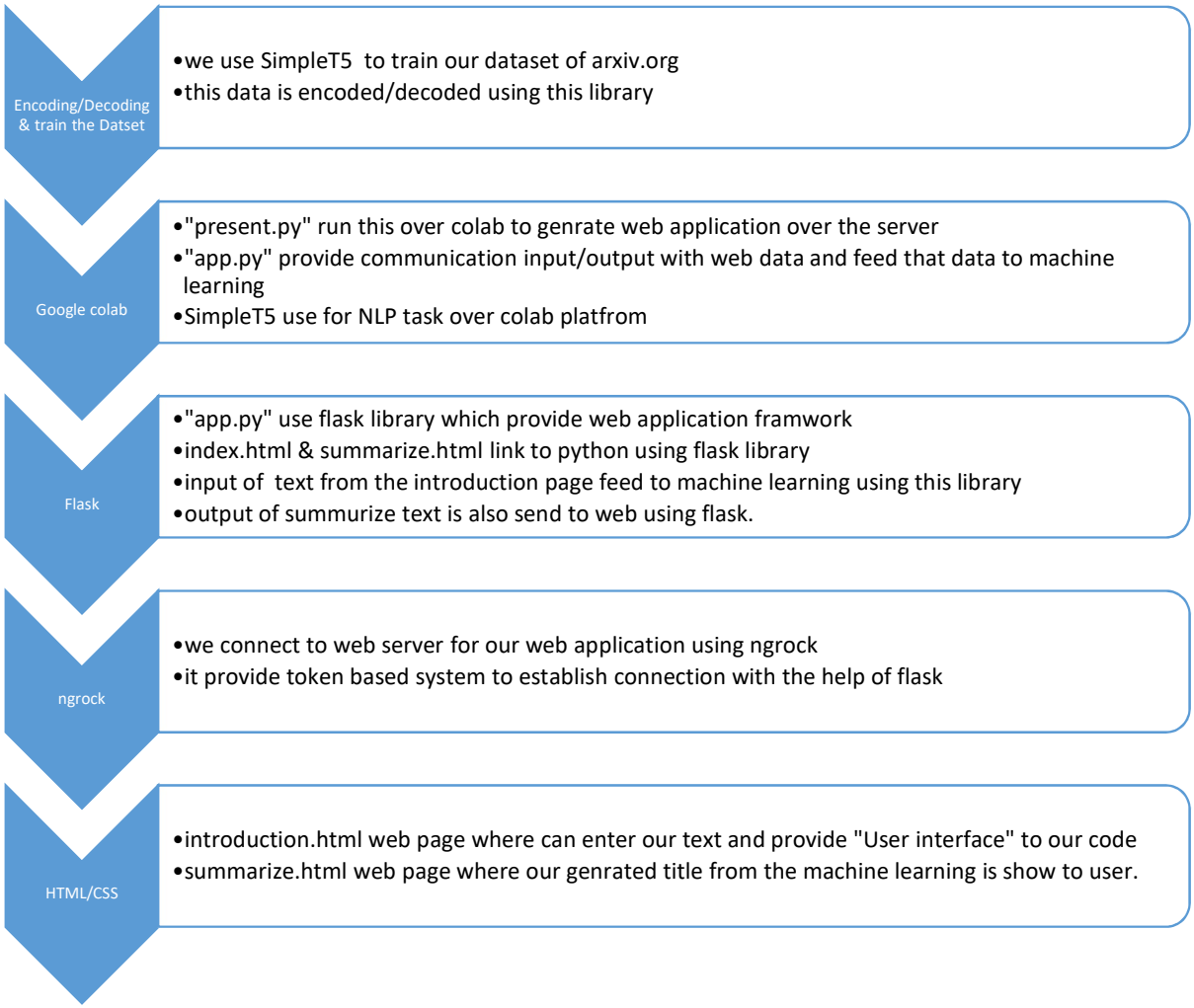

Fig.7 Brif system introduction (Reference: Self work)

2.2. System Architecture

System architecture have differnet part that are inter relevant to each other and creat system by it self. Every system have input data and after process we got output in form of text,graph,image,or any other form. Here is the fundamental system architecture for our title generation project is shown below. This structure serves as the foundation for the entire project.

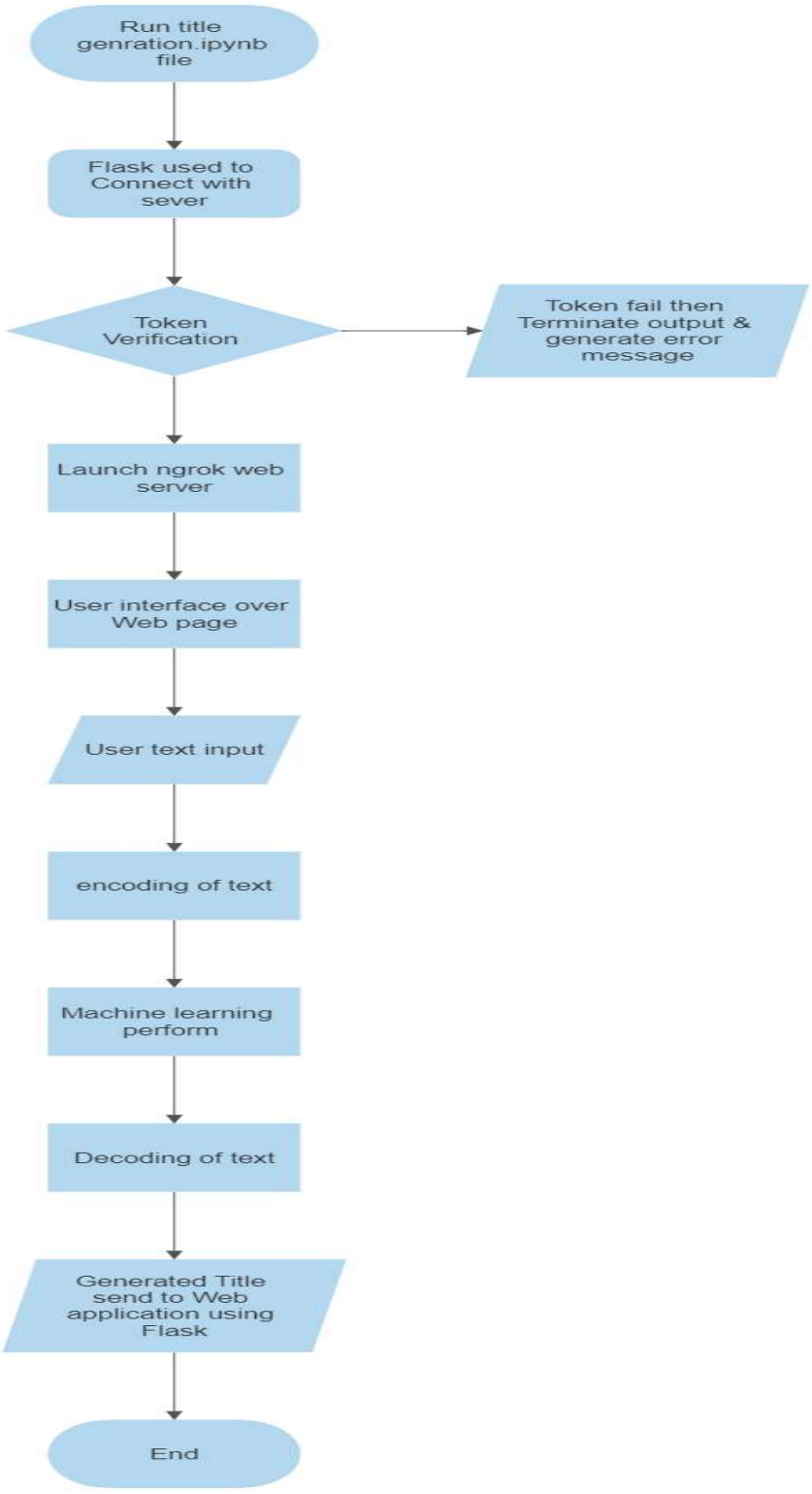

Fig.8 System Architecture for Title generation(Refernce: Self work)

Folder structure can be found in "Title Generation.ipynb." This folder contains all of the essential components, including the backend and frontend user interfaces, which are written entirely in Python. In addition, we used the arXiv dataset in that folder to generate titles, and our user interface is built using HTML/CSS and bootstrap.

First step of the system is to get dataset from the drive where we store our trained dataset of arxiv.org. Web application is our system and it consist of following part where we run python code using google colab to connect with web server where our application can perfrom. Ngrok is a server provider which give the token based server access to our system. flask is the web application framework which used here to establish connection with server. In the flask we check the and authorization of token is done. If authorization fail then we can not launch our web application over proxy server and our system fail at that point

If system authorize the token then user can get the server location at the end of result where our web application run. User interface is done at this point and we can enter the text of article at web application which is run over the ngrok server. This data is feed to our machine learning model where we get the article text and it is summarize using NLP method of SimpleT5.genrated output is send to server using Flask and final output show on web application's page

2.3. Main component and therir relationship

System contain dataset which is the main Part of this System where we store the relevant train model of the arxiv.org. we use research paper articles of arxiv.org to generate title of it and for that we need to connect our dataset to our python file. We store our dataset to Google drive and it will interconnect with our python code platform which is google colab using mount command.

Flask is the web application framework provider which is import to python code in “app.py” to use web data interface with our python code. We launch our web application using token verification process using flask in python. If our token is match then we get the web server where our user interface application show the introduction.html page which is a platform where user can enter the text article and click to submite button to generate the title using machine learning

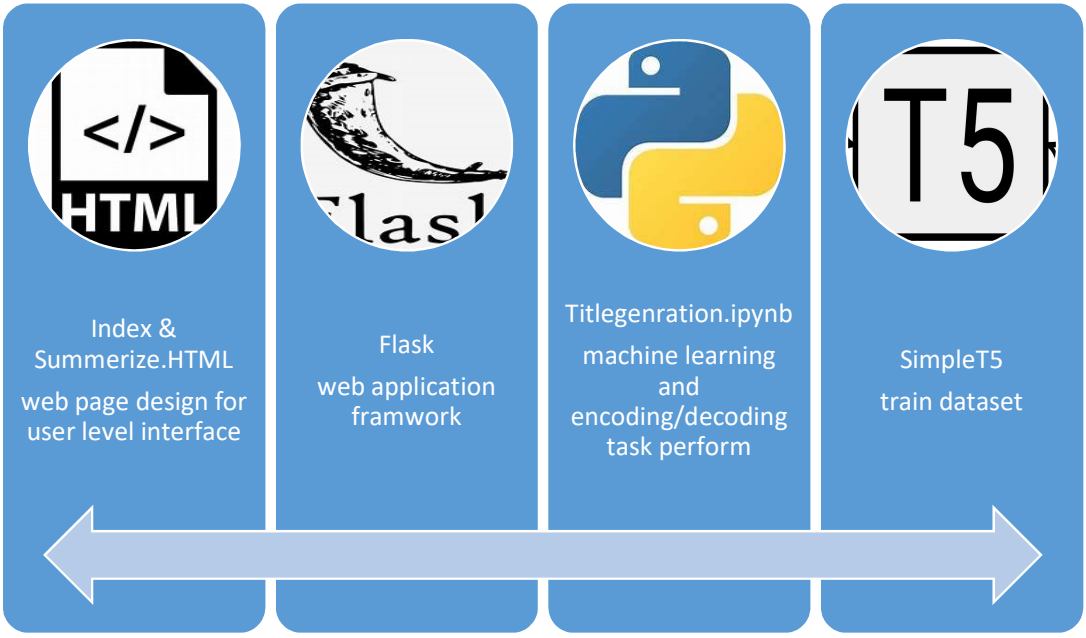

Fig.9 Relationship between key components (Refernce: Self work)

Machine learning is also a main part of the system where we process our dataset and text input which we get through Falsk. Machine learning is task perform over “present.py” and Flask for web application interface perfrom over “app.py”

Relationships between backend and frontend: In our project, users are sent to our main index page using flask routing from the file "app.py.". The page will render after we validate our ngrok auth token, and it is connected to both our model and our backend. The user will now provide input via the "POST" method , and it will be saved to our "input.txt" file. Our actual machine learning model was revoked in that "Input.txt" file, and the entire article was saved there temporarily before even being retrieved by our machine learning model. The next model procedure will produce a title, which it will place in the "output.txt" file along with temporary output. Now that the Title has been fetched, it will be redirected to our output screen in the frontend using our Python code.

2.4. Use case

These are better applications for text summarization software. Words by picking sensible statements from the subject and organising them in a thorough manner. This indicates that the article has been unchanged other than the removal of summary phrases. Intense summary: it functions by including its interpretation of the key phrase in the title.

- Watch media

In-depth discussion was had regarding the issue of information overload and "content shock." The ability to break up continuous material into smaller bits is made possible by automatic summarization.

- Newsfeeds

Weekly newsletters typically consist of an introduction and a few carefully chosen, pertinent topics. As opposed to a list of links, the summary will enable organizations to create and improve newsletters, which might be an excellent format for mobile phones.

- Search engine optimization

It's crucial to fully comprehend what your competitors are writing about in their content when examining SEO search inquiries. Due to Google updating its algorithm and switching to title authorities, this has become much more crucial (compared to keywords). Summaries of many documents can be an effective tool for swiftly assessing numerous search results, identifying recurring themes, and scrutinizing essential points.

- Internal document workflow

Internal information is frequently created by large firms and is frequently archived and utilized sparingly as informal data. These businesses must accept the tools that let them utilize previously collected data. Using a summary, analysts may immediately comprehend all the work a corporation has done on a particular subject and swiftly put together reports that include various

- Financial analysis

Investment banking companies invest a significant amount of money to gather data for decision-making, including computerized stock trading. If you are a financial analyst who reads daily news and market reports, you will eventually run into a wall and not be able to read everything. Analysts can easily spot market signals in material with the aid of summary systems created for financial documents like earnings reports and financial affairs.

- Analysis of legal contracts

A specific summary software can be created to examine legal papers, which is related to point 4 (internal document workflow). In this scenario, the summary might be useful for adding value by condensing the contract to risk categories or for aiding in contract comparison.

Data Flow diagram:

The first step in processing user input will involve testing out various data schemes and produce various patterns. The data will be processed with attribute analysis on one side, and the user's input will be analyzed once more. The majority of the useful data will be filtered out here and sent for preprocessing when data is processed using attribute analysis. Extraneous data will be eliminated during pre-processing, this data is now encoded and feed to the machine, and it will then be clarified using the given dataset's raw data. Following data cleaning, the derived data will be forwarded to the user's output in accordance with user requirements. Important: For pre-processing, the user input will be compared to our raw data.

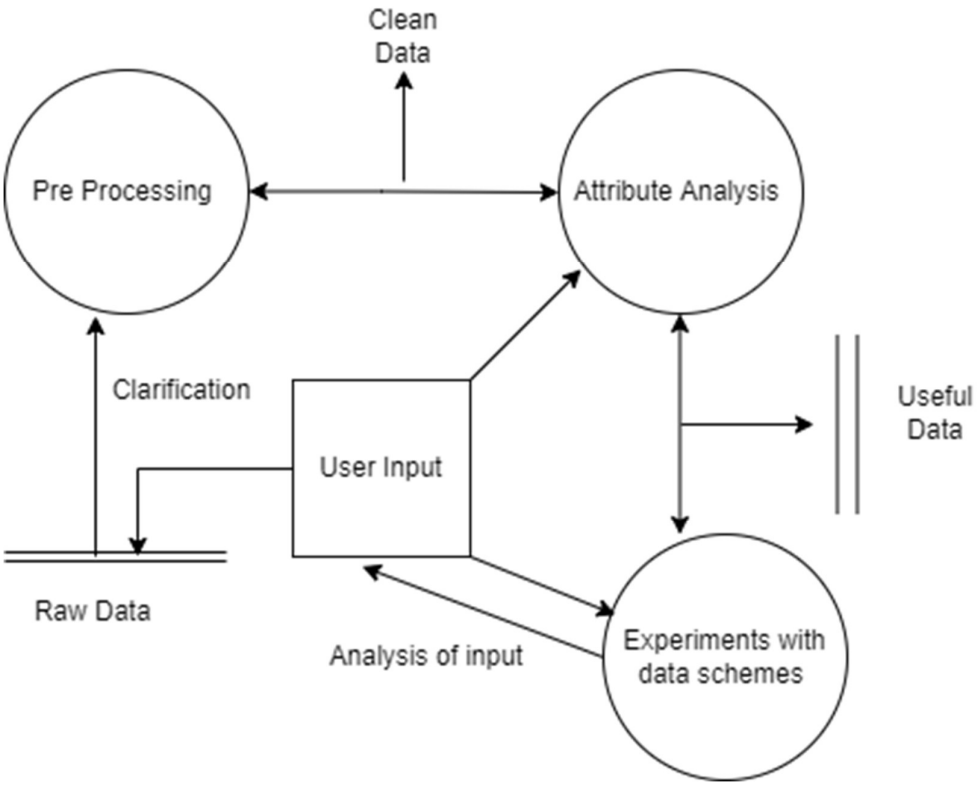

Fig.10 Flow diagram for title generation using designed web application(Reference: Self work)

Now our machine learning part start and it decide which words are key word for the article and that are used most in the article are separted and Usefull data seprate from the whole article where system doens not know it is a keyword. System just know which is high weighted and that words are seprated.

Decoding of that data is done using SimpleT5 where we got the sentence from the keyword which is the title for our article and this title feed to the flask and it will write that data to the web page and it will show to user

Data Dictionary:

When we have to work with large text dataset it will be complicated. We need to perform extractive text Summerization over random data of this large dataset. We need to train our model that much we can get relevant title of entered text

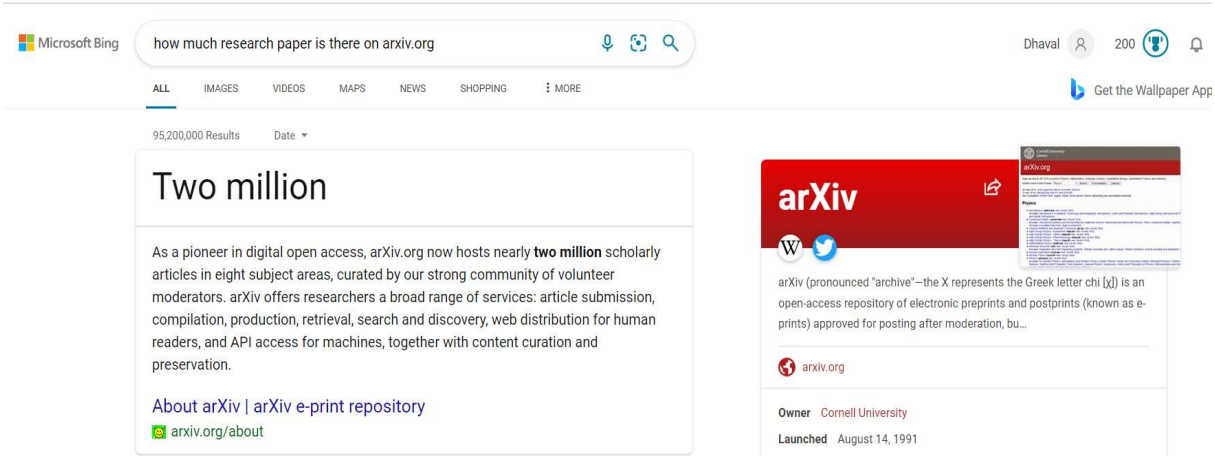

Fig.11 Screenshot of dataset detail of arxiv.org(Reference: Self screenshot of google answer)

Arxiv.org, where research papers are stored in electronic from on cloud and this data is openly available for researcher to gain knowledge. We have some portion of this large dataset in from of JSON file which is a line to line fromated text file. Which have id, admiter, author, title, summary etc. as its metadata.

Machine Model:

SimpleT5 is library which is widely used of NLP task where text need to process and our system use that open source library to train our dataset and machine learning process.

A T5 is an encoder-decoder model which converts all NLP tasks such as summarization, translation, question-answering, text generation etc. to a sequence-to-sequence task — converting a sequence of text (source text) to another sequence of text[Shivanand roy, 2021]

Encoding decoding is our key point. Our title generation should be done using Encoder/decoder method. And SimpleT5 is a library which perfrom this task. It have inbuilt function for text analysis , text Summerization , NLP, text processing etc.

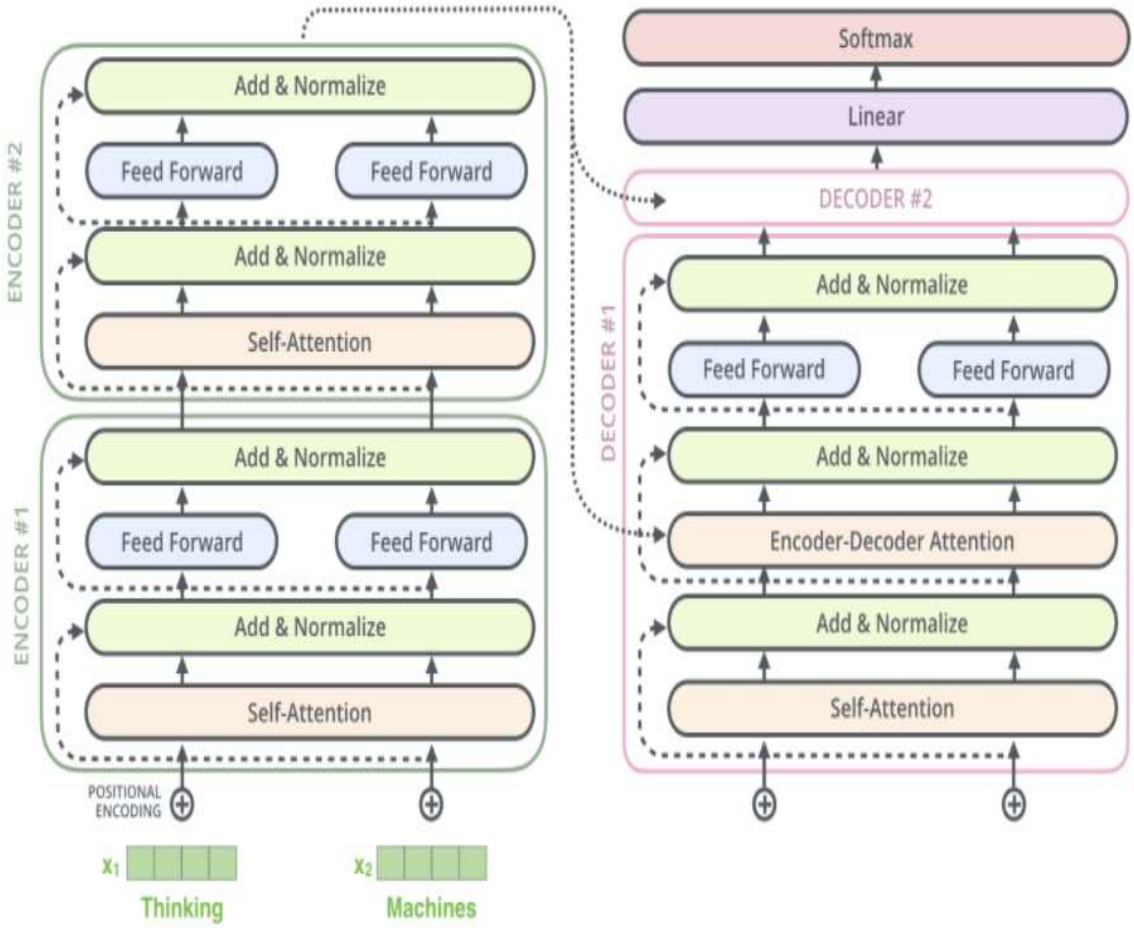

Fig.12 T5 model structure(Reference: Jay Alammar blog on github & simpleT5 paper)

We can see the example how encoding/decoding method work using T5 where input language is unknown(Not English) and we encode that in system and at the end decoding is done for that sentence and we got the result in unique language which is accepted by our dataset which is English. And T5 train that input information and execute it in English.

.

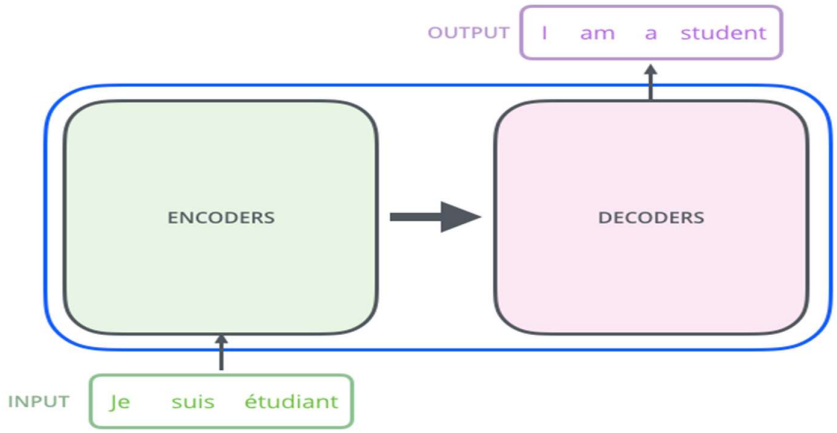

Fig.13 T5 model for encoding decoding work example (Reference: Jay Alammar blog on github)

Graphical user interface (GUI):

To make GUI for web application we need CSS for that and we can design CSS using visual studio which is platform where we can design website and web application. Web application is application which work as a website on local or proxy server and here Ngrock is used as proxy server provider and tokenization verify using Flask so we get the User interface for our application and its web pages are as bellow

- Index.html
- Sumerize.htm

Input text article is write in the text box created using this GUI in html we created “Summerize” button to transfer entered text to our system to generate title. It show as below.

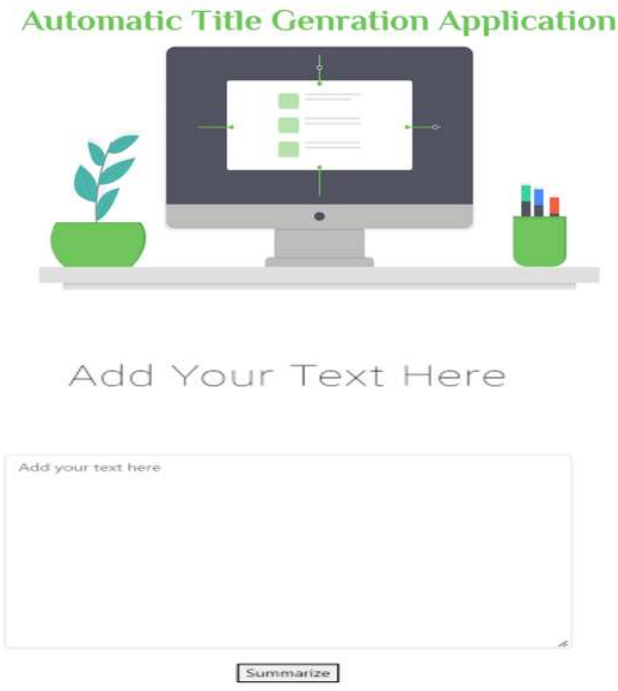

Fig.14 Screenshot of Index.html web page(Refernce: Self work)

This web pages are design in visual studio to provide graphical interface to the user. We design page layout using CSS which is backend platform for the html based web page and we have below index.html page where user can enter its text and when it click to the summarize button then input text will feed to to machine learning portion of system which use SimpleT5 model which summarize the input text and generate title from the relevant key word of the article which is feed to Summarize.html web page of application which show the result as below.

Fig.15 Screenshot of summerize.html web page(Refernce: Self work)

3. Implementation

3.1. Programming language issues

Insufficient Training Data:

- Lack of both quality and quantity of data is the main problem when employing machine learning algorithms.
- Despite the fact that data is essential for the processing of machine learning algorithms, many data scientists contend that insufficient data, noisy data, and filthy data severely tax these algorithms.
- A basic task, for instance, calls for thousands of sample data, whereas a sophisticated one, like speech or picture recognition, calls for millions of sample data examples. Additionally, for the algorithms to function optimally, data quality is also crucial, but machine learning applications frequently suffer from a lack of it.

Noisy Data:

- It causes an incorrect prediction that influences the decision and the precision of classification jobs.

Incorrect data:

- It is also responsible for faulty programming and results obtained in machine learning models. As a result, inaccurate data may potentially affect how accurate the results are.
- Generalizing output data: It is occasionally discovered that generalising output data grows difficult, leading to comparably subpar future actions.

Data of poor quality

- Data is crucial to machine learning, as we've already covered, and it needs to be of high quality.
- Less accuracy in categorization and poorer outcomes are caused by noisy data, incomplete data, inaccurate data, and unclean data. As a result, poor data quality can also be seen as a serious issue when using machine learning algorithms.

Non-representative training data –

- Sample training data must be indicative of the new cases we need to generalize in order for us to determine if our training model generalizes successfully or not.
- All instances, both past and present, must be included in the training data.
- Additionally, the model produces fewer accurate predictions when non-representative training data is used.
- A machine learning model is said to be optimal if it predicts well for generalized scenarios and offers correct judgments.
- The non-representative training set, which occurs when there are insufficient training data, causes sampling noise in the model.
- It won't make reliable forecasts. It will be prejudiced towards one class or group in order to combat this.

Overfitting and Underfitting:

- One of the most frequent problems that data scientists and engineers working with machine learning encounter is overfitting.
- A machine learning model begins collecting noise and erroneous data into the training data set once it is trained with a large amount of data.
- The use of non-linear techniques in machine learning algorithms, which produce non-realistic data, is the primary cause of overfitting.

3.2. implementation of key components of the system

Dataset train:

Dataset is key component of our systema and it is large as well in size. Above 2 million research paper store in JSON file which is used to train the dataset and we have to use this dataset as our trained model for extarcrtive text Summerization. For this task we use SimpleT5 library which provide netural language processing task as a function. It generate the title from the text which is entered by user.

Web application:

Our task is perfrom over web application and this web application is created using Visual studio which is in form of HTML file. Flask is used for web application framework and it used to intercommunicate our GUI to python source code.

When user enter the text into the designed text box to generate title. We press sumurize button and our input text is linked to our SImpleT5 library which is used to generate NLP task to get summarize output

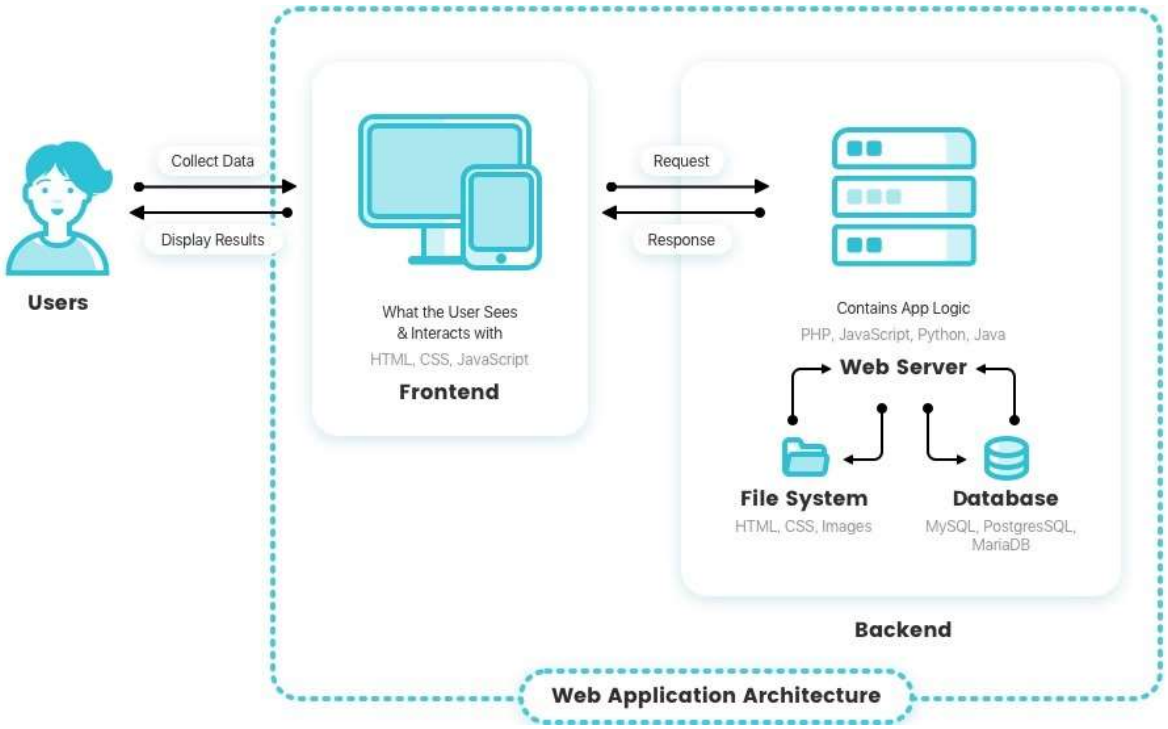

Fig.16 Web application architecture (Reference: MUSA-509 on Github)

Machine learning:

Automatic title generation is a unsupervised learning method in which we create title of give input article with user monitoring. Our system automatic decide which words are important and create relevant title from the choosen wored. For text analysis there are many open source libraries which can be used as our machine learning. SimpleT5 is a library which is used for project and its main task is text processing, text analysis and text summarization using NLP method. Our T5 model used to encoding/decoding for text input/output. As we know T5 required 2 elements. 1) train dataset 2) input article text

3.3. Overview of code listing

Automatic title generation using encoding/decoding method, this task done in 3 part, in which 1)Python for machine learning and dataset model train 2)Flask to check token and provide framework to web application 3)GUI in whichwe design web application over visual studio.

Present.py:

- It is a python based IPNYB file wich is colab/Jupiter nootbook file and it perform two taks one is machine learning and second is Encoding/decoding of text.
- Here we use SimpleT5 of Pytorch for our both task. It summarize the title from the given input text and it encode/decode the as well selected word to generate title.

App.py:

- Use of this notebook is to create web framework between pthon and web application
- It feed entered input text of user to the python machine learning model
- It send generated title of machine learning model to web application for GUI.

Index.html:

- It create under visual studio. It is a web page used for text Summerization process
- Use of this file to make GUI for user to enter text of article
- Flask used to get entered text and send it to our system for machine learning

Summerize.html

- Require to design under visual studio. It is a web page used for text Summerization process
- Use of this file to make GUI for user to display generated title to the web application
- It interconnect with python using Flask

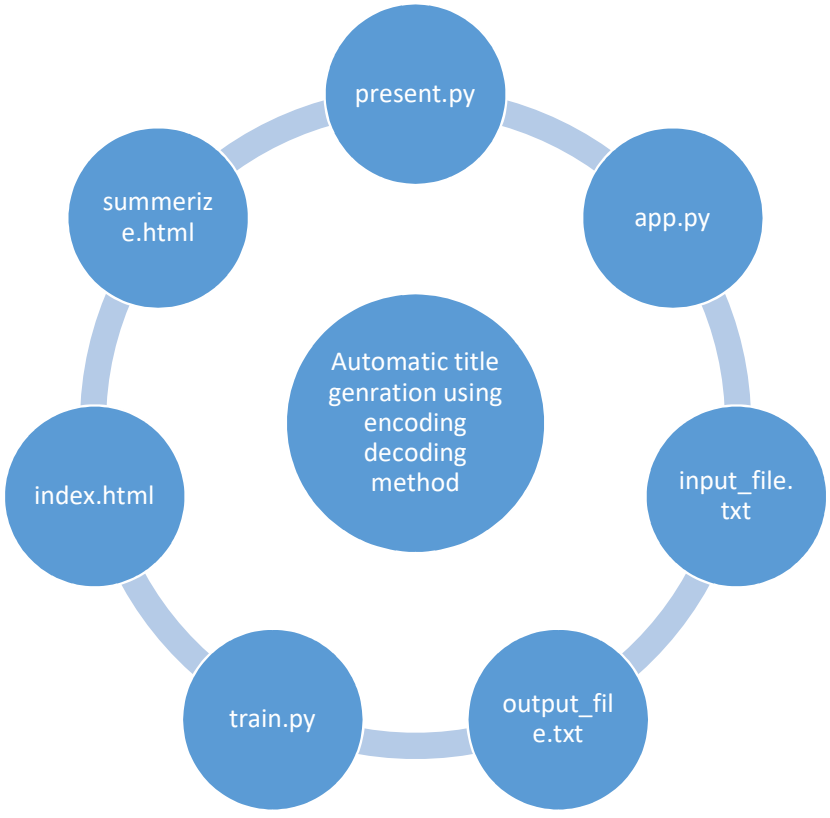

Fig.17 code list used to design oursystem(Reference: Self work)

Train.py

- Large dataset is a JSON file which is line by line data of arxiv.org and it need to train the dataset using algorithm which is flexible and open source. Our this step is call as a NLP task and we used SimpleT5 to train our data set and it generate dataset which is used for our machine learning model.

Input_file.txt:

- Entered text by user is transfer to text file using Flask based “app.py”
- It is used as a input to our machine learning model of T5.

Output_file .txt:

- System generate the Title of the given text input and this title is save in to .txt file and feed to the Flask to send over web application
- This is the output of our systema dn it is send to web application using Fask based “app.py”.

3.4. Tools used to generate code

Google Colab:

- A product from Google Research is Collaboratory, or "Colab" for short.
- Colab is particularly well suited to machine learning, data analysis, and education.
- It enables anyone to create and execute arbitrary Python code through the browser. Technically speaking, Colab is a hosted Jupyter notebook service that offers free access to computer resources, including GPUs, and requires no setup to use.

GPU:

- Originally created to speed up the rendering of images, a graphics processing unit is a specialized processor.
- GPUs are advantageous for machine learning, video editing, and gaming applications because they can handle multiple pieces of data at once.

Python notebook:

- You can create and share documents with live code, equations, visualizations, and text using the free and open-source Jupyter Notebook online application.
- The staff at Project Jupyter is responsible for maintaining Jupyter Notebook.

Visual Studio CODE:

- Debugging, task execution, and version control are all supported by the simplified code editor Visual Studio Code.
- It tries to give just the tools a developer needs for a speedy code-build-debug cycle and leaves more sophisticated processes to fuller featured IDEs, such as Visual Studio IDE.
- It is employed in the creation of computer programmers, websites, web applications, online services, and mobile applications. Microsoft's software development platforms, including Windows Store, Windows Presentation Foundation, Windows API, and Windows Forms, are used by Visual Studio.

4. Testing

4.1. Strategy

Logic and data combine and create dataset of model which creat a system to generate desired behavior. In our text summarization process our trained model isour data of arxiv.org and we create a logic using SimpleT5 to find the key word from the dataset. Behavior of the system can check using making strategy for process output and actual output means original title of article are check and verify manualy and automatically

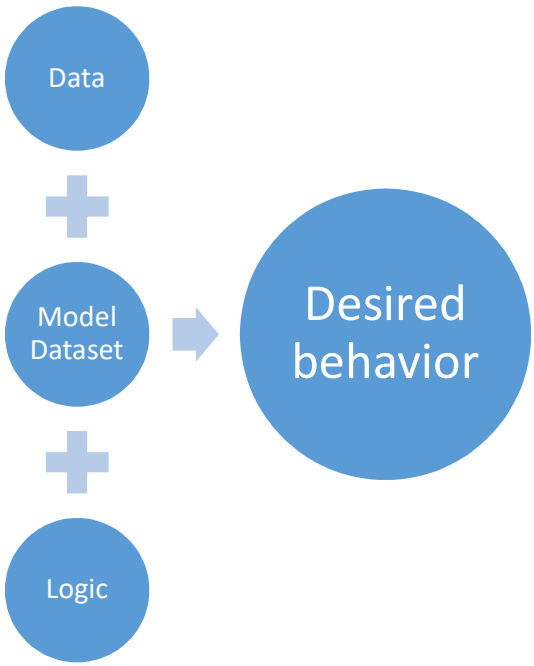

Fig.18 Model dataset Input/output(reference: self work)

In the machine learning, system dataset is train by user to get desired behavior and we create supported process that create our system logic. To check whether our deired system output generate required output continues., this generated output is tested. We generate evaluation report that include bellow factors.

- Need to check Perfomance of system over the unknown dataset and its desired behavior check over the normal system output
- We can use python function like precision and recall curve for graphical representation
- Use of assert function like to check the behavior with our pre-define value.
- GPU speed also check for the memory stackover error overcome.

We can check our system performance over different models and create a check list to verify the model which one is better. We can check behavior of every model also using this step. We can improve our system by watching the fault of differentsimilar models, by doing this we ca verify where models are fail frequently. We can make more investigation over that model and improve our model. Behavioral test for our model can perform as bellow.

When our line code converted to parameters in our machine learning method. It is more difficult to cover line code while running the change to convert it in the process parameter.Our aim is to get desired behaviorof our model, which is nearest to impossible to achive. We have to regular update our model and train it to get better result and this process.

4.2. Type of testing

Unit Testing :

Selenium is popular python testing framework and it is open source available on GitHub. Using this we can create simple python test adnwe can create complex testing of the system. we can do scalable test in python

Unit testing done using this is expressive ,compact and readable. Selenium testing of python created to test the dataset testing, cross browser testing and API testing selenium is compatible with python 3 and 3.5+..we can easly port form differnet testing model to unit testing is much easy. Selenium unit test can be done of for TDD(Test driven devlopment) system using open source In unit test we got parameters for testing and we can test different system by changing the simple Common modules/class/session is test commonly using available

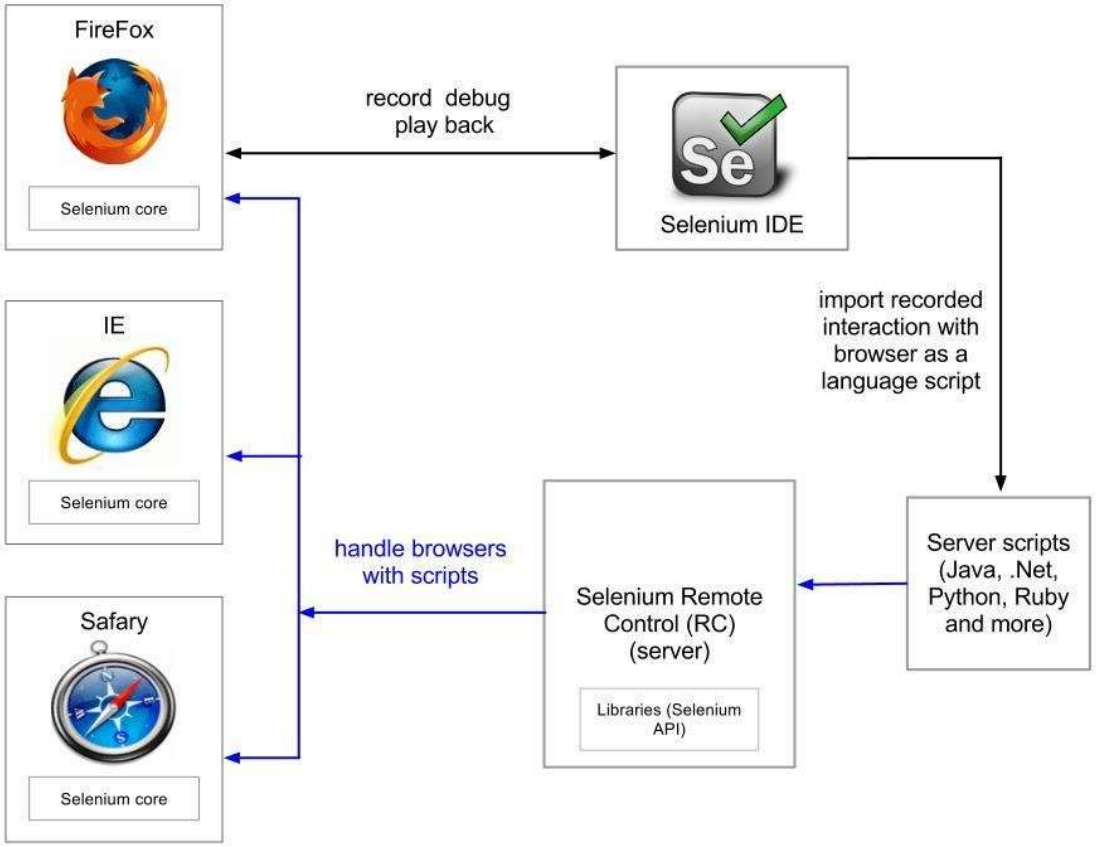

Fig.19 testing of web application using Selenium (Reference: arvindpadmanabhan on slide.com)

As pytest is not a part of the standard Python library, it required to be install separately. To install pytest, run below command in command prompt and write code as below.

- pip install –U pytest

In unit test when our model or code is fail then it broken up with small field test and it can test by passing the dummy value to the system As ATDD use first, it helps to system to resolve the failed error lessen the effort required to resolve bugs and defects as the project moves forward. ATDD doesn't address "How," simply "What." Thus, it becomes much simpler to satisfy customer needs as a result. Together, developers, testers, and customers are brought together through ATDD, which aids in understanding what is expected of the system.

White Box Testing :

Machine learning will as it were ended up more commonplace at undertaking level, but knowing the distinction between black-box and white-box models is vital to making the correct choice for your organization. Machine learning (ML) is quick getting to be a major zone of intrigued for organizations of all sorts and it is making a difference to control everything from cybersecurity defense to enrollment and chatbots.

While dialog approximately the benefits has provoked the intrigued of numerous businesses, choosing how to actualize ML can be overwhelming. White-box models are the kind of models whose behavior, forecast prepare, and influencing variables can all be clarified in detail.

The highlights of a show must be caught on, and the ML handle must be straightforward, in arrange for it to qualify as a white-box. Straight and decision/regression tree models are among these models. Black-box models, on the other hand, are ordinarily exceptionally non-linear by plan and are more troublesome to get it in common.

- Examples of these models include boosting, random forest, and deep learning (deep neural network).
- Users can only see the input-output relationship using black-box models. For instance, enter the customer profile and then produce the propensity score for customer churn.
- However, the underlying causes or methods used to obtain the output are not known.

Black Box testing	White Box testing
It is a way of software testing where the internals of the software is never exposed.	It is a way of software testing where the internals of the software is exposed to the test environment.
This is mostly like testing if the software is working fine and if the software is giving the expected output.	This is mostly like testing if the software is working by consuming the data in the way it is supposed to do.
This can be carried out even by a non-technical person.	This test is carried out by a software test engineer and also by software developers.
The Black Box tester need not have knowledge of the implementation. He just needs to know what must be the output for the corresponding input.	The White Box tester needs to have the knowledge of implementation because he needs to check the internals of the code.
This can also be referred to as 'External Software testing' and 'Closed testing.'	This can also be referred to as 'Internal Software testing' and 'open testing'.
The functionality of the software is tested in this.	The structure and operations are tested in this form.
The design for Black Box testing can be carved out of the requirements specifications.	The design for White Box testing can be carved out of the detailed software design document.
This consumes less time.	This takes more time.
This is not suitable for algorithm testing.	This is the preferred one for algorithm testing.

Fig.20 difference between White box testing method and black box testing method for python(Reference: educba.com)

Blacl box testing:

Machine learning (ML) black-box testing alludes to testing without having get to to the model's inner data, such as the calculation utilized to construct it and the highlights it contains. Black-box testing's essential objective is to reliably keep up the models' quality. Finding the test prophet, a method for knowing in case a test has passed or fizzled, is intense in black-box testing.

In the setting of conventional program improvement, models are rendered testable by utilizing a test prophet, such as analyzers, test engineers, or testing components working in couple with the test program. The comes about of a test can be checked against the anticipated values utilizing an prophet. Be that as it may, since to the challenges of doing black-box testing on ML models, they are as often as possible respected as being untestable. Since ML models give expectations, there are no anticipated values to compare test comes about to. Pseudo-oracles are utilized within the nonappearance of a test prophet.

Supervised learning :

To deliver numerical expectations, relapse models are utilized. What would the stock's cost be, for occurrence, on a specific day? To decide the course of a set of information, classification models are utilized. Whether a individual has an ailment, for occurrence, or not. •

Unsupervised learning :

Finding profitable bits of knowledge from the information is made simpler with the help of unsupervised learning method. Unsupervised learning is impressively more like how people learn to think through their possess encounters, which brings it closer to real manufactured intelligence. Unsupervised learning method is more noteworthy since it works on unlabeled and uncategorized data. Unsupervised learning method is fundamental to handle circumstances when the input and yield are not continuously the same within the genuine world.

Reinforcement in learning methods : Machine learning incorporates the teach of fortification learning. It includes acting fittingly to maximize remunerate in a certa

4.3. Integration level

Common paradox when it comes to information items is that they cannot be subjected to mechanized testing. Due to their exploratory and stochastic nature, a few components of the pipeline cannot be tried utilizing customary strategies, in spite of the fact that the larger part of the pipeline can.

Additionally, more sporadic calculations can be subjected to specialized approval procedures. The distinctive sorts of tests you might make for an application are spoken to by this pyramid. We start with a expansive number of unit tests. Which look at each person piece of usefulness independently from others. Then we make Integration Tests to see on the off chance that combining our disconnected components carries on as expected.

Finally, we make UI or acknowledgment tests that confirm the application capacities as expected from the user's point of view. The pyramid isn't all that distinctive when it comes to information items. We are at generally the same levels.

You ought to too take into consideration different outcomes for each useful component (in case an on the off chance that explanation exists, at that point all conditionals ought to be inspected). After at that point, each commit would trigger the execution of these as portion of your nonstop integration (CI) pipeline.

Unit tests help us in investigating issues in expansion to guaranteeing that the code capacities as planning. We can affirm that a recently found bug is settled when we believe it to be settled which it won't happen once more by including a test that duplicates the bug. Finally, these tests offer assistance us portray the desires we had whereas planning the include, in expansion to guaranteeing that the code works as intended.

These tests are implied to check whether collected components that were made autonomously perform as expected. These can decide whether a information pipeline: which The cleansing of the information yields a dataset reasonable for the model. The demonstrate preparing can oversee the given information and produces outcomes.

4.4. Acceptance :

- Lowered costs overall and development effort
- Increased consistency and quality
- More rapid release cycles
- Simple test dissemination over numerous devices or locations
- Improved reporting capability, among other things.

5. Conclusion

Simple T5 is the machine learning model which is flexible and effective to make text summarization process. It have inbuilt function to work with NLP and text processing task which is more usefull for us.

Arxiv.org research papers are in JSON file which is train with the help of SimpleT5 which used to encode and decode the text input. While we eneter the text to our web application it is send to our SimpleT5 model where it seprate the selective word that are key words and encode them before machine learning task perfrom and it gives better encoding result for our system

Our gerated title is decoded at the end of machine learning process to generate the sentence from the selected words. Machine learning is essentially a probabilistic mathematical model that involves numerous computations. These activities are extremely simple for people to complete, yet computational machines can complete them very similarly very quickly.

Since a model may need to calculate and update millions of parameters in run-time for a single iterative model like deep neural networks, consumer hardware may not be able to perform large computations very quickly. There's hence room for equipment that capacities well with significant calculation. But to begin with, let's get a handle on machine learning stream some time recently we get profoundly into equipment for ML.

The method of making a machine learning show includes four steps: preparing the input data The profound learning model's training Archiving the profound learning show after training Application of the Model

The machine learning model's preparing is the one that requires the foremost compute out of all of these. Now let's talk almost preparing the show, which ordinarily includes a part of computational power. If done without the correct equipment, the method might be disappointing. Distinctive network duplications make up this neural network's intensive portion. How in this manner can we speed up the preparing model? To accomplish this, all the exercises require as it were be carried out at the same time instead of consecutively. The GPU, with its numerous centers built to compute with essentially 100% productivity, enters the scene in this circumstance. It turns out that these processors are appropriate for running neural organize computations as well.

The struggle between CPUs and GPUs is in favor of the last mentioned due to the gigantic number of GPU centers offsetting the 2-3x speedier CPU clock speeds (3500 (GPU) vs. 16). (CPU). The centers of the GPU are a disentangled form of the more complex CPU centers, but since there are so numerous of them, the parallelism and execution of the GPU can be increased. Since CPUs are built to perform about any calculation, they are alluded to as general-purpose computers. A program instrument the Number-crunching Rationale Units (ALUs) which registers to examined, perform an operation (such as an expansion, increase, or coherent AND), and which enroll to utilize for yield capacity, which in turn contains parts of sequencing of these read/operate/write operations. CPUs store values in registers to attain this sweeping statement. CPUs are more costly in terms of control and chip zone since they require more bland bolster (registers, ALUs, and programmable control).

GPU :

Polygon-based computer design are delivered by GPUs. GPUs have developed in preparing capability as of late as a result of the request for authenticity in present day video recreations and visual motors. A GPU could be a parallel programming setup that employments CPUs and GPUs to handle and examinations information in a way comparable to how an picture or other realistic frame would be prepared. Initially aiming for more compelling and generalized visual preparing, GPUs were in the long run appeared to be a great fit for logical computing. The to begin with GPU-based matrix multiplication computation took put in 2001. The primary strategy to be put into utilize on a GPU was LU factorization in 2005

Prediction : Our aim is to predict the better title from the key word andhere machine learning and model that is used are tunned in such a good way that we got approx. nearest result for our entered text.
Classification : Distinguishing a case's thing course or category through text classification. The course or category of a case is anticipated employing a classification approach. For occasion, whether a cell is generous or cancerous, or whether a buyer will adhere around.
Clustering : Information summarization, information clustering, and information structure revelation. Amassing clusters of related cases. For occurrence, it may be utilized to recognize comparable patients or for client division within the managing an account industry. Anomaly location : Finding atypical and exceptional cases is known as peculiarity location. Finding bizarre and exceptional cases is finished through peculiarity detection. For occasion, it is utilized to recognize credit card fraud.

Sequence mining :
Sequence mining, event prediction, and click-stream (Morkov Model, HMM).The next event is predicted via sequence mining. Like the website click-stream.

Vs code :
Debugging, task execution, and version control are supported by the simplified code editor Visual Studio Code. It seeks to give developers only the resources they require for a short code-build-debug cycle and leaves more sophisticated workflows to IDEs with more features, like Visual Studio IDE.

Google colab :
Colaboratory, sometimes known as "Colab," is a Google Research product. Colab is particularly well suited to machine learning, data analysis, and education. It enables anyone to create and execute arbitrary Python code through the browser.

GitHub :
A platform for collaboration and version control is called GitHub. It enables remote collaboration on projects between you and other people. You will learn about GitHub fundamentals like repositories, branches, commits, and pull requests in this tutorial.

Jira :
Jira Software is a member of a family of tools for managing projects in teams of all sizes. Jira was initially intended to be a bug and issue tracker. Today, however, Jira has developed into a potent work management solution for a variety of use cases, including agile software development and the management of requirements and test cases.

How we manage the project :

- Team work is key to success of our project.
- We face many challenges like someone cant complete the task on time then we all need to work together to complete that task.
- We conduct Daily meetings for sharing information to seek new ideas for better outcome in project.
- We Regularly research on our project to make our model more efficient and accurate.
- we checked in checked out codes on GitHub.
- We also conduct review session daily to know each others perspective and opinions on model.
- pipelines for machine learning, data preparation, and exploration.
- Experimenting and fine-tuning models.
- Versioning of the data, pipeline, and models.
- Infrastructure administration and run orchestration.
- Dishing up models and industrialization.

Fig.21 Teamwork win always(Refernce: teamwork.com)

Our jira work of team is as shown bellow.

Automatic Title Generation Using Encoder-Decoder Models

Supervisor: Juntao Yu
Modual Supervisor: Dr Vito De Feo

Project Start: 16-May-22
Today: 29-Jun-22
Display Week: 1

TASK	ASSIGNED TO	PROGRESS	START	END
Phase 1: Research about Project				
Research Paper search	All	100%	16-May-22	19-May-22
Flow Chart Design	All	100%	16-May-22	19-May-22
Making List and Install required Software and envionment on each System of team memb	All	100%	16-May-22	19-May-22
Phase 2: Work Distribution & Design				
SimpleT5 Learn and use to train Dataset of arxiv.org	Dhaval Patel, Karan Bhatt	100%	19-May-22	24-May-22
Flask learn	Ashish Gajera, Vatsal Trivedi	100%	19-May-22	23-May-22
CSS design for Webapp	Vipul	100%	19-May-22	26-May-22
Webpage Design	Niaz	100%	19-May-22	24-May-22
Reporting of work to team leader	All	100%	24-May-22	25-May-22
Phase 3: Design				
Check Train model with present.py	Dhaval Patel, Karan Bhatt	100%	25-May-22	14-Jun-22
Flask inter-communication with python and make app.py	Ashish Gajera, Vatsal Trivedi	100%	24-May-22	13-Jun-22
Find Tokenizer Server for Web application platform to launch	Vipul, Niaz,Dhaval Patel	100%	25-May-22	16-Jun-22
Introduction & Summerization Web page connection establish with Flask	Niaz,Ashish Gajera	100%	25-May-22	14-Jun-22
Reporting of work to team leader	All	100%	14-Jun-22	16-Jun-22
Phase 4: Model Training and Integration				
Choose model which one is good for our text search and use more epoch for better result	Dhaval Patel, Karan Bhatt	100%	15-Jun-22	24-Jun-22
Use Ngrock server in Flask based Web application and execute web application function	Ashish Gajera, Vatsal Trivedi	100%	24-Jun-22	26-Jun-22
Check status of Textbox fucntion of web page and INPUT/OUTPUT from flask to present.p	Vipul, Niaz	100%	15-Jun-22	26-Jun-22
Check all the function of application	Dhaval Patel, Karan Bhatt,Ashish Gajera	100%	18-Jun-22	23-Jun-22
Upload all data to Drive and create flask for title genration	Dhaval Patel	100%	23-Jun-22	24-Jun-22
Phase 5: Testing and Report Writing				
Unit test of every individual files of Python and HTML	All	100%	24-Jun-22	27-Jun-22
Automatic test perfrom by random check for title genrration form arxiv.org	Dhaval Patel, Karan Bhatt,Ashish Gajera	100%	11-Jun-22	27-Jun-22
Code analyzing	All	100%	26-Jun-22	28-Jun-22
Report writing	All	100%	28-May-22	28-Jun-22

Fig.22 Ganttchart of our task with task date for project planning(Reference: Self work)

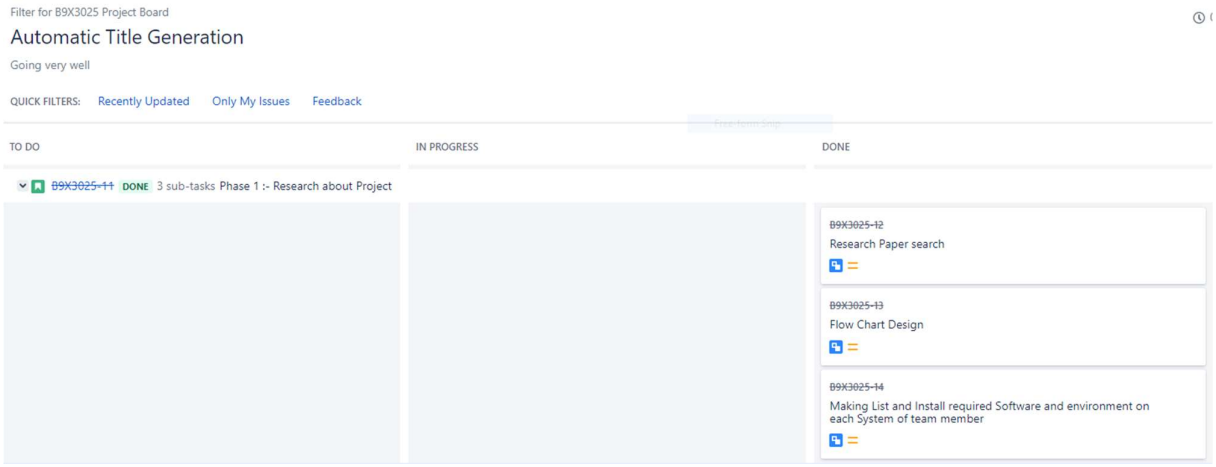

Fig.23 Jira Phase1 work assign(Reference: Self work)

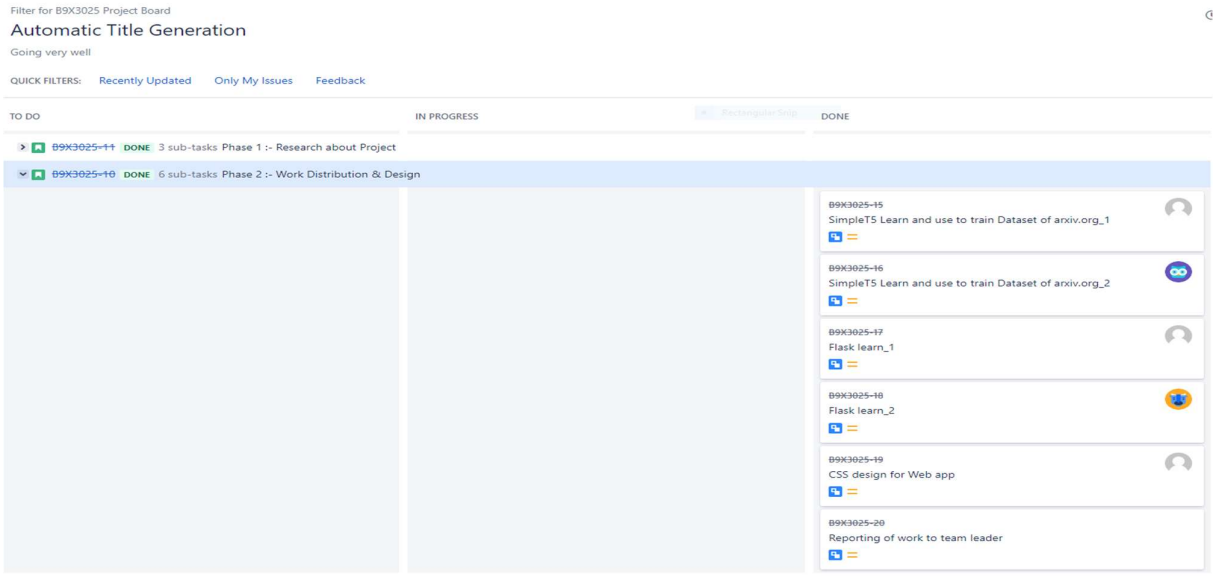

Fig.24 Jira Phase2 work assign(Reference: Self work)

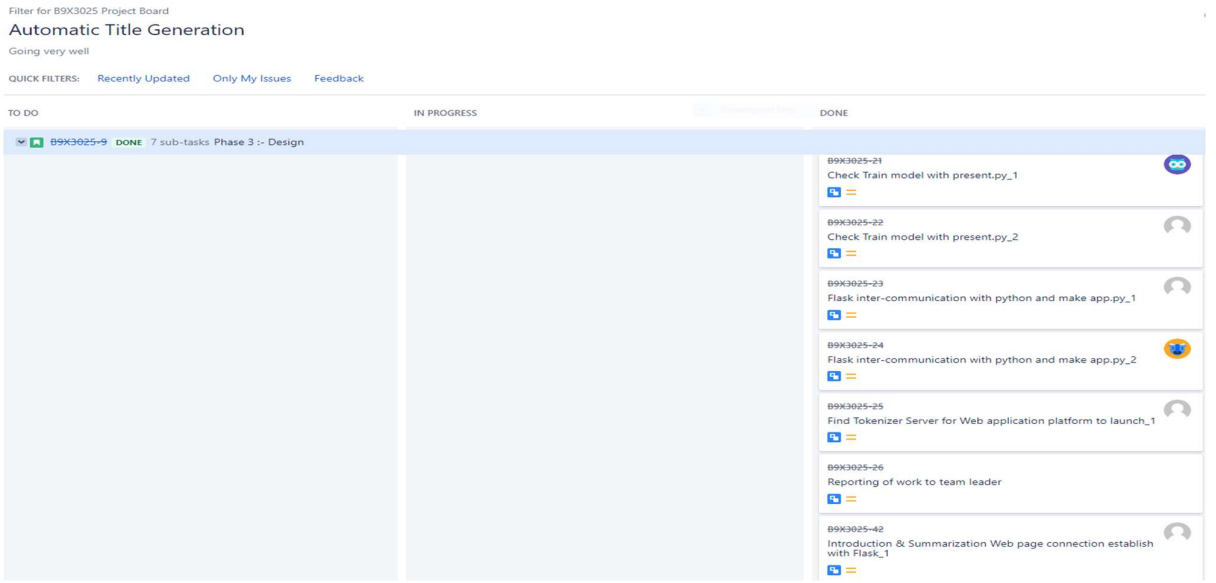

Fig.25 Jira Phase3 work assign(Reference: Self work)

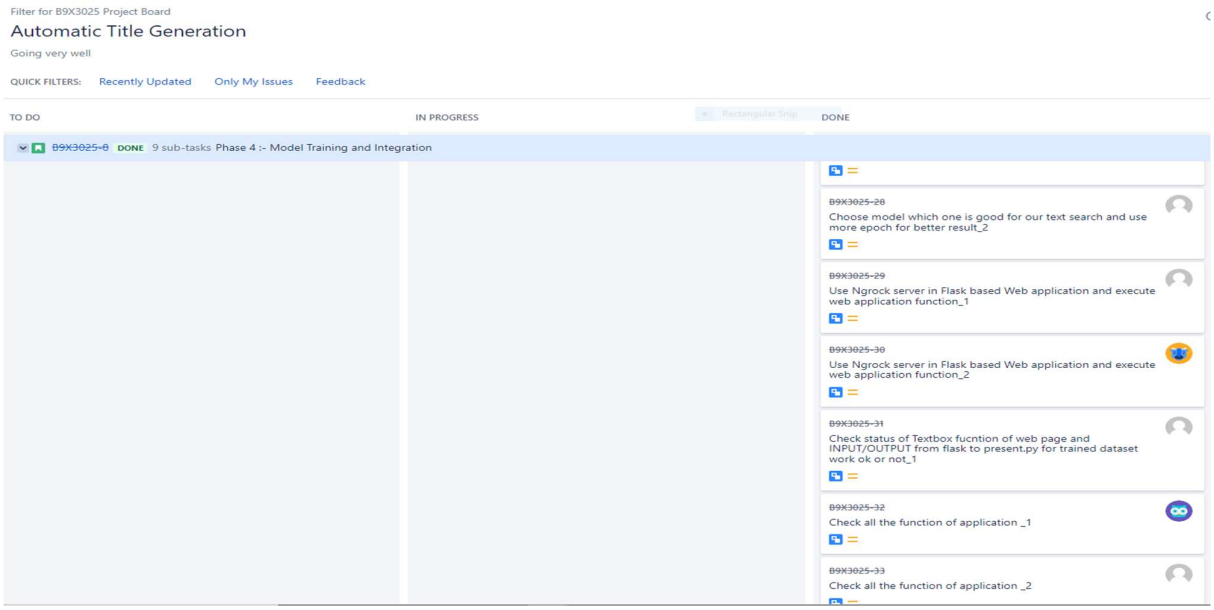

Fig.26 Jira Phase4 work assign(Reference: Self work)

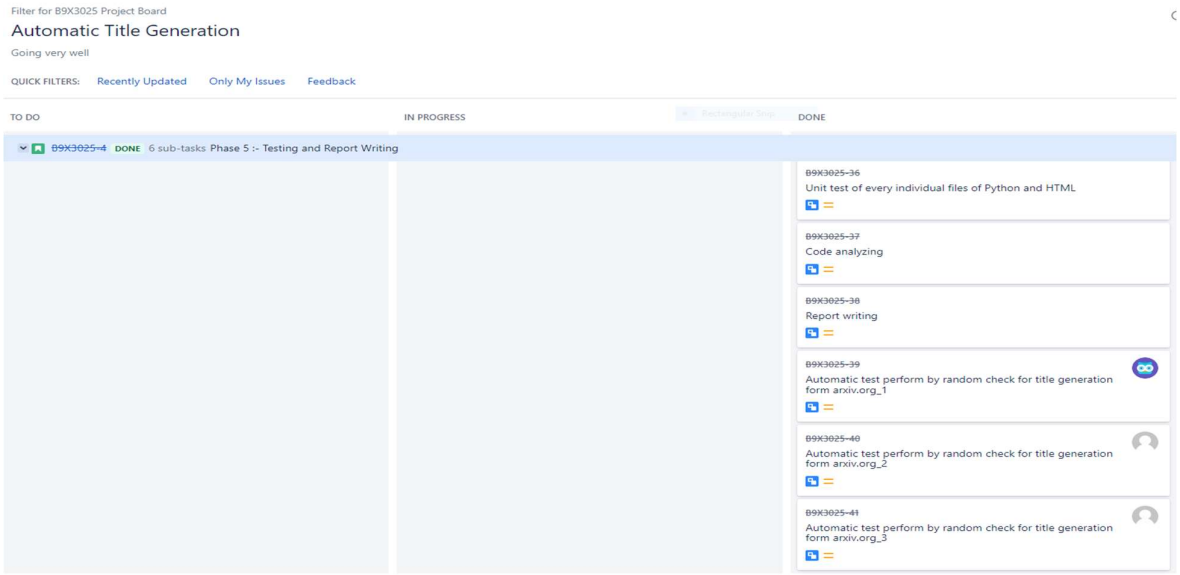

Fig.27 Jira Phase5 work assign(Reference: Self work)

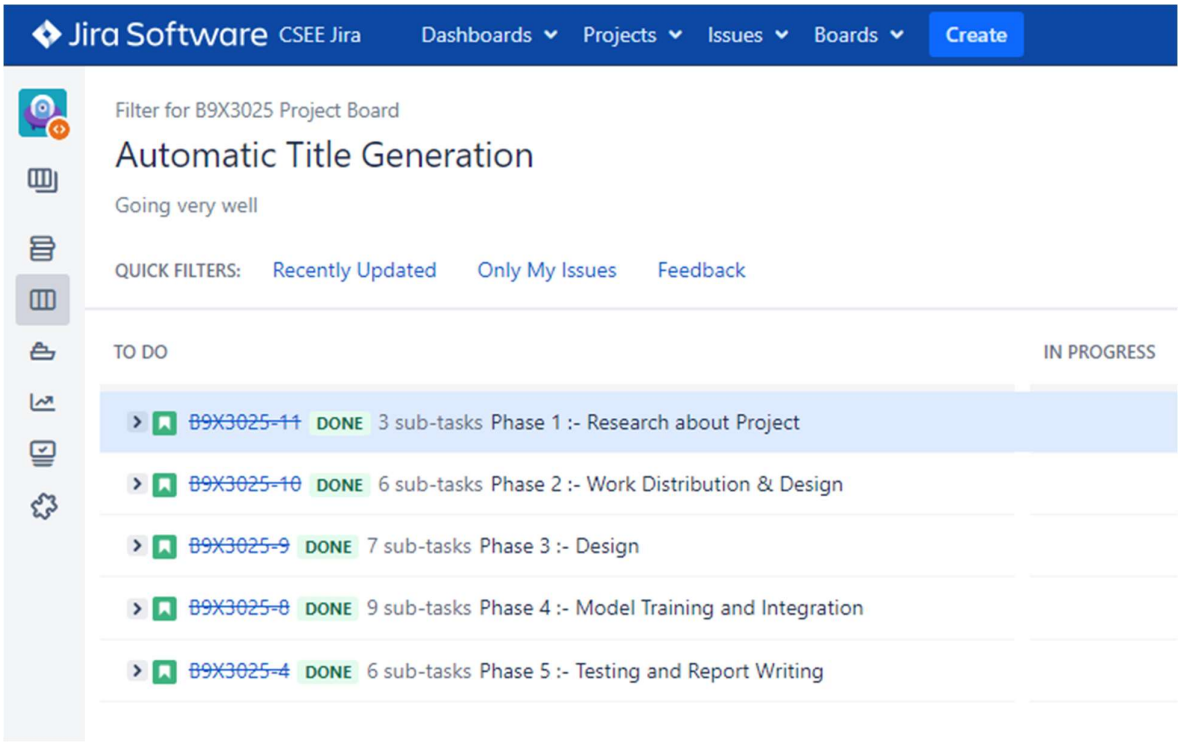

Fig.28 Screenshot of jira, Final all task completion of project(Reference: Self work)

Major benefits of GitHub :

- You can build, test, deploy, and perform CI/CD in the same location where you manage code. To automatically publish updated package versions to GitHub Packages, use Actions. Install GitHub Packages or your favorite registry of records packages and images in your CI/CD processes.
- With the aid of vulnerability warnings from the program, you may secure your job and reduce risks while learning how CVEs influence you.
- Code review is made simple and convenient by the built-in review tools. A team can suggest modifications, contrast different versions, and provide input.

6. Reference

[1] Quick tour (huggingface.co)

[2] Natural Language Processing with Python by Steven Bird, Ewan Klein and Edward Loper

[3] Applied Text Analysis with Python: Enabling Language-Aware Data Products with Machine Learning Benjamin Bengfort, Rebecca Bilbro, Tony Ojeda

[4] https://scikit-learn.org/

[5] Introduction to TensorFlow

[6] Long Short-Term Memory Networks With Python by Jason brownlee (Develop Deep Learning Models for your Sequence Prediction Problems)

[7] Python Natural Language Processing: Explore tools and techniques to analyze and process text with a view to building real-world NLP applications by Aman Kedia

[8] Introduction to Machine Learning with Python: A Guide for Data Scientists by Andreas C. Mueller

[9] Python for Data Analysis, 2e: Data Wrangling with Pandas, Numpy, and Ipython by Wes Mckinney

[10] Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems by Aurelien Geron

[11] Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. From neural sentence summarization to headline generation: A coarse-to-fine approach. In IJCAI,pages 4109–4115, 2017.

[12] Yuko Hayashi and Hidekazu Yanagimoto. Headline generation with recurrent neural network. In New Trends in E-service and Smart Computing, pages 81–96. Springer, 2018.

[13] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1724–1734, 2014.

[14] Abhijith Athreya Mysore Gopinath, Shomir Wilson, and Norman Sadeh. Supervised and unsupervised methods for robust separation of section titles and prose text in web documents. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 850–855, 2018.

[15] https://tutorials.one/encoder-decoder-models-for-text-summarization-in-keras

[16] Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In Proceedings of the ACL Conference onEmpirical Methods in Natural Language Processing (EMNLP), pages 355–362. Association for Computational Linguistics

[17] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. 2013. Audio chord recognition with recurrent neural networks. In ISMIR.

[18] Alex Graves. 2012. Sequence transduction with recurrent neural networks. In Proceedings of the 29th International Conference on Machine Learning (ICML 2012).

[19] [Yu et al., 2016] Lang-Chi Yu, Hung-yi Lee, and Lin-Shan Lee. Abstractive headline generation for spoken content by attentive recurrent neural networks with ASR error modeling. In SLT, pages 151–157, 2016

[20] [Graves, 2013] Alex Graves. Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850, 2013

[21] [Luhn, 1958] Hans Peter Luhn. The automatic creation of literature abstracts. IBM Journal of Research and Development, 2(2):159–165, 1958