iNeuron iNews Project

Project Requirements:

Project Description: News app that selects latest and best news from multiple national and international sources and summarises them to present in a short and crisp 50-60 with top 10 hashtag.

Technology:- Git, Python, MongodB, Web Scraper, NLP, Django/Flask, AWS, etc

Project Leader: Harshit

Project Team : Akash Gupta, Askhay Ashish, Avinash Yadav, Md Akram Khan, Mounika Inavolu, Nitin Rajput, Rahul More, Raju Datla, Saikat Pandit, Sanjay Challal, Sankalp Talankar, Santhosh Y,Suman Biswas Sailesh Pandit, Sofia Saini, Sasidhar Kambhampati,syedkhadarvalli.

Table of Contents

1. Overview:

A few years back, most of the information was available to us in physical form, either in newspapers or in books / magazines. Today technological advances like the internet, smart devices (Phones, Tablets, etc) enabled the information available everywhere with a single mouse click.

However, the information available on the web is not in precise form and does not give importance to sentiment of the user.

The aim of this project is to collect news articles from various news sites, consolidate, analyse the news for positive and negative sentiment and present to the user with the right contextual titles and news in concise form.

Benefits of iNews:

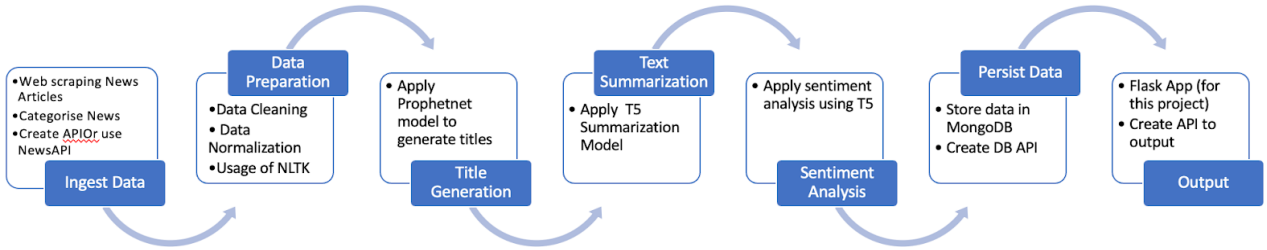
The following are key benefits of news summarization and applying sentiment analysis

1) The application will provide titles with right context with rightful summary thus saving time for the reader.
2) With each news, a sentiment of the article is displayed. Based on the sentiment, the user can decide whether they want to read the article or skip the article.
3) Users will get latest information thereby allowing the application to be used / extended to various business applications like sentiment analysis for stock brokers or fund managers giving them valuable input so that the right decisions can be taken.
4) The application is created and data is exposed as API so that it can be consumed by various media like websites, mobile applications or by other software for further processing.

Technologies:

Language	Database	Summarisation model	Sentiment analysis	Title generation
Python	Mongodb	T5	T5	Prophet Net

2. High Level Design and Application flow:



Ingest Data:

Ingest data using web scraping various news sources. Web scraping is the process of using bots to extract content and data from a website. It extracts HTML code and stores data in the database. Web scraping is the first part of this application where we will scrape the web for recent news using python web scraping script. This scraped data in the raw form will be stored in mongodb database.

** For this project, the API from https://newsapi.org/ is used.

API Structure (assuming prebuilt API from newsapi is used) [Check Section3]

To use the API, you have to generate an API key by creating an id at https://newsapi.org/ .

For all articles about Tesla from the last month, sorted by recent first

http://newsapi.org/v2/everything?**q=tesla**&from=2021-02-02&sortBy=publishedAt&apiKey=58f304ff540642adbe4816847fcefbc4

For top business headlines in the US right now

http://newsapi.org/v2/top-headlines?**country=us&category=business**&apiKey=58f304ff540642adbe4816847fcefbc4

For any query (put your query in place of 'data' below)

https://newsapi.org/v2/everything?q='+data+'&apiKey=58f304ff540642adbe4816847fcefbc4

For API documentation you can visit the link below

https://newsapi.org/docs

Data Cleaning / Data Transformation

Data cleaning is the process of preparing data for analysis by removing / modifying the data which could be incorrect, incomplete, irrelevant or duplicates.
NLTK library will be used for data transformation.

Data Persistence

Data persistence or storage is the process of storing information for lateral usage.
The cleaned data is stored in the MongoDB database.

Text Summarization:

Text summarization is done using the T5 model. T5 is an encoder-decoder model that converts all NLP statements into a text-to-text format. The text summarization is stored/updated in Mongodb database.

Title Generation:

Titles are generated using ProphetNet. Prophetnet is an encoder-decoder model which can predict n-future tokens for ngram language modelling instead of just the next token. The titles are updated in the Mongodb database.

Sentiment Analysis:

T5 model is used to do sentiment analysis of the news and results are updated in the mongodb database.

Output

Output the results as API and display results using flask as website.

3. Developer Guidelines:

1) At each stage, always create a Python API so that the next service can use it. List of API (update as go-along).
2) You can use JSON or CSV for outputs (create a reusable json/CSV inputter/outputter class)
3) Each key stage either producer or consumer or both in terms of data.

3. 1 API

API Source	API	API Description
database_mongo.py	store_news(title, descr, other)	
	fetch_news_mongo(*args)	Fetching info from database
generators.py	generate_title(*kwargs)	Run the model and generate title
	title_output	Stored (pandas dataframe)
	Header generation(f5)	F5
main.py	Business logic	
prophenet	get_title(params1, param2)	Return dataset, pandas, json, string

3.2 Developer Standards

1. Please read PEP8 standards at https://www.python.org/dev/peps/pep-0008/. Few naming conventions we adopt in the project are:
    a. All class names will follow CapWords convention. Ex: DatabaseUtils
    b. All function names will follow lowercase, with words separated by underscores Ex: get_news [Please note most of the developers are habituated with camelCase/mixedcase so the example becomes getNews().]
    c. Use lowercase for method names [similar to function names]
    d. Use one leading underscore only for non-public methods (private methods) and instance variables.
    e. Use CAPITAL letters for constant declaration

f. Always specify exceptions whenever possible rather using a bare except: class.
g. Always make a habit of inline comments for the codes you write.

3.3 Source Code Management:

All source codes (Except models as they are huge files) will be maintained in each of the developer git repositories.
Please update the following table with your name and git repository url for the project.

Developer Name	Git URL
Raju Datla	https://github.com/drraju/iNews
Akash Gupta	https://github.com/iseakash/newsscrapper
Md Akram Khan	https://github.com/mdakram09/ProphetNet-News-Titles-Generation https://github.com/mdakram09/Scraping-News
Mounika Inavolu	https://github.com/mounika131/ineuron-inews
Sofia Saini	https://github.com/sofiasaini/iNEWS
Sankalp Talankar	https://github.com/sankalpt27/News-Scraping
Shailesh Pandit	https://github.com/shailesh897/iNews-Web-Scraping
Suman Biswas	https://github.com/suman889/Django_Project
Syed khadarvalli	

3.4 Models

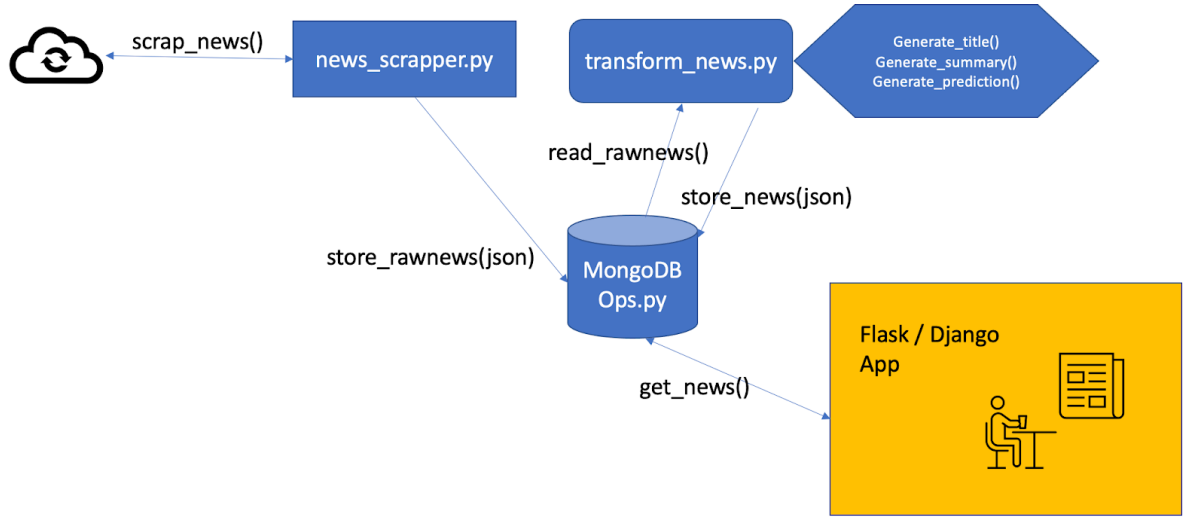
For this project it is decided to use models - T5, ProphetNet, etc. Due to high disk space required, these are hosted on google shared drive. Once a developer downloads and makes a part of the project, from git point of view, add these into ignore so that these will not be uploaded to the developer github. This saves bandwidth and also any file size limitations set up by free tier of github.

Google shared drive for iNews models:
https://drive.google.com/drive/u/0/folders/1MiShXFiZrRuv-i90XnrE7uoOeLZGFeGX

Model Name	Model Description	Remarks
ProphentNet		
T5		
T5 Sentiment Analysis		

3.5 Detailed Flow (rough thoughts)

app.py



1. Web Scrape a news site for 10 top news. To do this create a python file and write function which reads news.[ex: scrap_news.py]. Function : def scrap_news(url, no_of_articles). Number of articles defines how many news articles you want to retrieve. It returns json object
2. Create database api, say mongodbapi.py and create functions like store_rawdata(json), read_rawdata(), store_news(json), read_news(). All functions return data in json format.
3. Create transformations.py (example name) and create functions like get_title_prophetnet(), generate_summary_t5(), generate_prediction_t5().
4. From app.py / main.py, call above functions and store final results into mongodb using store_news(json) from mongodbapi.py

3.6 Important URLs

a) Kamaurya and Rohit Demo of web scraping video
https://www.youtube.com/watch?v=mazz0P93aBY
b) Kamourya source codes
https://drive.google.com/file/d/1r11Mu0MBNUX59ckCrMlE5fq7jhWG3pWT/view
c) Sudhanshu's Mongodb explanation -
https://www.youtube.com/watch?v=U09KoC3Cbkk

4. Project Implementation Notes

4.1 Project Phases (Project deadline - 31st March 2021)

Phase	Description	Deadline
Data Ingestion	Create API or use existing API to scrap news	Completed
Data Persistence	To create an API to interact with the mongodb database. To store and retrieve. As we are using API, we may not need to use NLTK so check on this	8th March 2021
Title generation	Learn and use ProphetNet and test to work with ingested news	15th March 2021
Usage of T5	Learn to use T5 model and integrate with project for text summarization	20th March 2021
Sentiment Analysis	Learn to use T5 for sentiment analysis and integrate with project	26th March 2021
Output	Showcase on Flask/Django is a continuous process. Final output to showcase overall functionality of the application	30th March 2021
Demo Day	Demo completed application	31st March 2021

4.2 Project Assigned Tasks

Task Description	Assigned To	Remarks / Status
Database API	Nitin, Akash, Sofia	14th March / 15th Demo
Title Generation	Nitin to update model api	