MINDS Sentiment Analysis Task – MINDScraper

- Rhythm Girdhar (6742001330)

Description:

As given in the assignment document, I have performed the 6 tasks in the following way:

1. I have created a GitHub repo for the Python 3.8 project – https://github.com/RhythmGirdhar/MINDScraper

   It contains the requirements.txt file which can be used to install external libraries and other dependencies using command:
   pip3 install -r requirements.txt

2. I have taken 10 recent articles from https://www.aljazeera.com/where/mozambique/ by using Beautiful Soup library to scrape the website. Here are the links to the articles:

   a. https://www.aljazeera.com/news/2022/5/23/floods-hit-south-africas-kwazulu-natal-province-again'
   b. 'https://www.aljazeera.com/news/2022/3/18/mozambique-cyclone-gombe-death-toll-rises-to-53
   c. 'https://www.aljazeera.com/news/2022/3/4/mozambique-announces-new-prime-minister-and-finance-minister'
   d. 'https://www.aljazeera.com/economy/2022/3/1/analysis-can-african-gas-replace-russian-supplies-to-europe'
   e. 'https://www.aljazeera.com/news/2022/1/27/at-least-70-dead-from-tropical-storm-ana-in-southern-africa'
   f. 'https://www.aljazeera.com/news/2022/1/12/southern-africa-bloc-sadc-extends-mozambique-mission'
   g. 'https://www.aljazeera.com/program/start-here/2021/10/3/climate-change-and-famine-start-here'
   h. 'https://www.aljazeera.com/news/2021/9/24/in-mozambique-kagame-says-rwandan-troops-presence-to-continue',
   i. 'https://www.aljazeera.com/news/2021/8/8/rwanda-mozambique-forces-recapture-port-city-from-rebels'
   j. 'https://www.aljazeera.com/news/2021/7/10/rwanda-deploys-1000-soldiers-to-mozambique-cabo-delgado'

k. 'https://www.aljazeera.com/news/2021/6/23/southern-african-nations-agree-to-deploy-forces-to-mozambique', 'https://www.aljazeera.com/news/2021/6/20/infographic-world-refugee-day-journey'
l. 'https://www.aljazeera.com/news/2021/6/9/dozens-of-children-mostly-girls-abducted-by-mozambique-fighters'
m. 'https://www.aljazeera.com/news/2021/5/14/whites-prioritised-ahead-of-black-people-in-palma-rescue-amnesty

The data is stored in JSON format in *data.json* file, in this format:

[{'title' : '##title###', 'subtitle' : '###subtitle###', 'content': '###content###'}, …]

3. There are different parts in the article, and I have kept the title, subtitle, and main content for sentiment analysis. For this part, I have considered content only. (There is another approach I have tried to use, that is explained later).

   For preprocessing of data, I am removing all the stop words in English language (as given in the NLTK library), and the punctuation marks from the content. I have additionally converted all the text to lowercase.

4. For Sentiment Analysis of content, I used SentimentIntensityAnalyzer from nltk.sentiment. For each article, I tokenized it into sentences and ran the analyzer on it to get polarity scores (pos, neg, or neu). The content_score is the mean of the scores from each sentence.

5. Then based on the scores received for each article, they are put into different classes.
   a. If the score is between -0.5 and +0.5, it goes to neutral.
   b. If the score is greater than 0, it goes to positive.
   c. If the score is less than 0, it goes to negative.

   Maintaining the order of the above statements. I put it all in a dataframe and use plotly to plot a pie chart of frequency of polarity scores stored in the dataframe.

Tech Stack Used:

Programming Language: Python (ipython – Jupyter notebook)
Libraries:

1. Scraping – BeautifulSoup, newspaper3k, requests
2. Text Manipulation – Pandas, Json, copy
3. Sentiment Analysis – nltk, statistics
4. Visualization - Plotly

What's different?:

Along with the approach of using only the content of the article, I tried two different things:

1. Using the title, subtitle, and content – This was a little tough, but I came up with an equation to give different weightages to all the three in the sentiment analysis. So, content gets 50%, title gets 25%, and subtitle gets 25% weightages.

2. Using TextBlob of content – Instead of tokenizing, I am directly using the entire pre-processed content as a text blob and calculating the sentiment polarity for it.

Visualization: