INFM 603
Angela Tseng
Yogesh Boricha
Abdul Shaik
Sadaf Davre

Project Test Report

Project changes

Since the project prototype, there have been various alterations to the system. The system still runs by fetching top hashtags from an area based on WOEID, gathering articles with the hashtag, selecting five and cleaning before passing the articles through an algorithm that gives a 0 or 1 based on validity. However, there are a few changes that can be noted.

First, there have been minor refactoring adjustments. All the pip installations can now be found at the top of the code, to be run before the rest of the code. This change will fix the errors that occur when the Colab Notebook is refreshed or when the code is rebooted.

Second, article cleaning and exception handling has been improved. The original code pulls five articles per hashtag and then removes the exceptions. Exceptions happen when the program is unable to derive the requested information from the website or when the article itself is behind a paywall. Removal of articles after selection often resulted in less than five articles being saved into the data.csv file since exceptions were removed last. Now, the exceptions are removed from all articles before fetching five articles to be saved into the csv file for further parsing. This increases the number of news articles that are available to the user.

Third, a prompt for the user to enter a WOEID has been added before the code chunk that fetches the trends. When that block is run, the user can enter a location ID and see the trends for that area after running the following block.

Fourth, we attempted running the system with a different classifier. Our original classifier was the TF-IDF Tokenizer with Passive Aggressive Classifier. The biggest advantages of TF-IDF come from how simple and easy to use it is, which is why it was used with the initial system. To further our testing, we attempted to use the BERT tokenizer. BERT uses a WordPiece tokenizer where it splits words either into the full forms or into word pieces. Full forms mean that one word can be broken into its own token, and word pieces are where one word can be broken into multiple tokens.

The project can be accessed here.

System Instructions

The prototype can be accessed from this folder under the Project Prototype Colab Notebook. All other necessary files to run the program are contained in the folder. The project focuses on programming and machine learning components instead of database or website design. Centered largely around executing and displaying results of the machine learning code directly to the user, the Colab notebook is the only file the user needs to open. The notebook contains code that requests top hashtags from Twitter, parses for news regarding the hashtags, and assigns a 0 or 1 label to a news source depending on its reliability according to trained data. A label of 0 implies that the source may not be reliable, while a label of 1 suggests that the source is worth looking at.

Running the prototype takes some time since certain packages must be installed and data is trained to assign a label to the fetched news sources. However, due to the iterative nature of Colab notebooks, simply running the cells one by one and waiting for cells to complete running is enough to receive results. To change the location of trends, simply run the code block with the prompt for the WOEID before running the block to fetch the trends. This parses for links for that particular topic. Beyond that, each block after should be run chronologically. The last block of code assigns a 0 or 1 to the left of a linked article, allowing the user to scroll through the list in the console to select an article of their choosing. More detailed instructions and explanation of the implementation can be found in the Colab notebook itself.

Project Test Plan

- Your project test plan.
    - Based on the initial draft test plan that you submitted with your project detailed design with any changes that resulted from design changes or differences between your plan and your actual implementation, and with any improvements based on what you have learned from our comments, from the experience of building the system, and from actually trying to run the test yourself.

The initial draft test plan submitted with the project detailed design had too large of a scope. The original plan included a website that would present the user with a list of the top hashtags currently trending on Twitter. These hashtags would be clickable links that take the user to a list of news articles related to that hashtag. Alternatively, the user could type a hashtag of their choosing in a search bar on the website and search to look for relevant news sources that mention that hashtag. We have since then eliminated the website component altogether, replacing the user interface with simply displaying the results in the Colab Notebook. Users are no longer able to select a hashtag of their choice, but can now adjust their WOEID (Where On Earth ID) to their desired location. The top hashtags are still displayed to the user.

In the original plan, only articles that have been determined trustworthy would be displayed. Trustworthiness is based on various parameters, for example; incorrect grammar usage, incorrect spelling, language crudeness, and random symbols usage. The fewer the errors in the article, the greater the possibility that the article is trustworthy, and the higher the chance that the article will be displayed in the results. However, this has since been changed. Now, in the interest of transparency and recognizing the possibility of bias in the program's decision-making, all results are displayed with a 0 or 1 accompanying the link to access the article. Users are able to select and view both rated articles if they are interested.

Since every process, from fetching articles to rating the articles, is all completed in the Colab Notebook, there is no longer a need for a MySQL database to store the data. As a result, both the database component and website components have been eliminated for us to focus on writing the program.

Since the program already successfully assigns numbers of 0 or 1 to the news articles, there are multiple different ways that the project could have been tested. For example, the code could be assessed for precision and recall. Precision is the fraction of relevant instances among the retrieved values, while recall is the fraction of relevant instances that were retrieved. Alternatively, the classification used in the project could be assessed with balanced accuracy. Finally, the system's decisions can be validated against a human's decisions.

The BERT classifier is a transformer based technique where we coded sentences into vectors and retrained using the BERT model. The model which we have explored in depth up till now i.e. the TFID and Passive Aggressive Classifier Algorithm vectorised a text sequence from left to right, or in a combination of sequential traversal forms. However, BERT applies bidirectional training of the attention model, to compute word embeddings. It has a much deeper sense of language context, than any other existing models.

The team chose to try the last method to manually check samples and compare how the program assigned score compared to the human-ranked results. There are two problems associated with this. First, there is a possibility that the checker themselves are biased and may assign the human-ranked results incorrectly. To counteract this, the program's decisions are not disclosed to the checkers. The scores are tallied and compared to the computer generated number by a separate person who does not participate in the scoring. The second issue lies in human error. There is a possibility that even if the checker is not biased, they may just simply be incorrect. This issue can be alleviated by simply having multiple checkers to compare the results.

Test Results

For the keyword “McCarthy”, two out five articles were assigned a “1”. These two were articles from CNN and Fox News respectively, and were voted trustworthy by the checkers. However, there was disagreement about the last two articles in the list. The program rated an NPR and Washington Post article untrustworthy. Both news companies are widely considered a prime source of news, and so the checkers disagreed with the program’s decision about rating these articles with a “0”.

For the keyword “Grammy”, only one article was assigned a “1”. This article was an article from the New York Times listing the full list of nominees with a short description. The first article is from CBS and consists largely of graphics rather than text. This may have contributed to its rating of 0, since the article itself was determined to be fairly trustworthy by the checkers. The next following articles were from the Vulture, ET, and billboard. While the articles themselves may have presented accurate information about the Grammy’s, the articles were written in a more casual language than a typical news article. As a result, the ratings varied between each checker.

For the keyword “World War III”, there were 2 articles assigned a “1”. Of the two, the checkers disagreed with both ratings. The program had rated an article partially behind a paywall as trustworthy, and when the available content was reviewed, the checkers marked it as untrustworthy. The article includes phrases such as “Dr. Realist” and “Dr. Doom”, lowering how reliable the checkers believed the article to be. The other article rated “1” is a bloomberg article written as an opinion piece, and as a result, was rated unreliable by the checkers. The checkers agreed with the other articles that the program rated “0”.