Twitter and Misinformation UROP Project – Mini Assignment

Starter Files

You are given code written by former UROPs who worked on the project, that generated the results shown during the interview. Specifically:

- **`code/storywrangler_api.py`**: Uses the StoryWrangler API to query the popularity of given n-grams on Twitter over periods of time.
- **`code/matching.py`**: Reads the sample CorpusData (1% of the entire dataset) that contains news articles, and output the most popular 2-grams and 3-grams.

The starter files also contain the datasets we used. **`sample-data/text.txt`** contains the sample CorpusData, where each entry is a news article. **`sample-data/database/sources.txt`** contains the news sources that corresponds to each entry.

Instructions

First, read through the given code and try to understand what they're doing. Familiarize yourself with the datasets, APIs and packages used.

Now implement any ideas you may have! Here are some examples you may consider, but feel free to implement anything out of the box. If you do decide to choose from these ideas below, one of them would be enough, and you do not necessarily need to address these questions fully – just make as much progress as you can.

- Pick one of the Python scripts and replicate the progress yourself by rewriting from scratch. Feel free to define your own input/output format or use different packages. It doesn't have to output the exact same results nor handle all corner cases perfectly, but they should achieve the same overall goal. You may even make the code more efficient too, if you can!
- Based on the given code, reproduce the StoryWrangler visualization that shows the popularity of one or more given n-grams on Twitter over time. You will want to use the StoryWrangler API, and use **`matplotlib`** or similar packages to generate the plot in Python.
- Alternatively, examine the news corpus over periods of time. What are the most popular n-grams in each year or each month, and can you identify any trends in their popularity? You can find the dates of each news article in **`sources.txt`**, and match their IDs with **`text.txt`** (see given code).
- How about linking the two pieces together? For example, given the popular n-grams from the corpus data, can you automatically feed them to the StoryWrangler API, and see if their popularities on social media and in mainstream media are correlated in some way? You can address other questions of your choice, too.

Make sure to document your code properly regardless of what you do.

Do not spend more than a couple hours on this mini-assignment. We also don't expect your work to necessarily move us forward in our research questions at this stage, so don't feel pressurized.

By **Sunday, May 29, 11:59pm EDT**, give us any code you have with a brief description of what you did, and any presentable results such as plots and tables if applicable. Also include any special instructions on running the code (e.g. paths to data files) if needed. If you need an extension, kindly let us know.

Good luck and have fun!

APIs

StoryWrangler: Visualization, API

TextBlob tutorial for counting n-grams (This is just one of the many approaches. The given code uses another approach.)

Google Fack Checking API (We mentioned this during the interview as an idea for classifying n-grams. We do not expect you to use this, but just in case you're interested.)

Additional reading (only if you're curious): How Tucker Carlson Stoked White Fear to Conquer Cable