CS 410 Project Documentation

Overview

Our project involved extending the functionality of TA Bhavya (https://bhaavya.github.io/) Text Mining System by implementing several new features in order to extend its functionality. The existing system already allowed users to generate topics and search for related keywords based on a corpus of historical newspaper articles. The purpose of this system was to aid in social science research by finding relevant strings of words in order to quickly find historical information about a specific topic.

Our functionality extension can be seen in the “Topic Mining” web page which we created. This web page gives a user the option to upload .txt files to generate a visualization of up to 20 topics from the uploaded file using an LDA algorithm. Clicking the submit button takes the user to an html page containing an interactive visualization which displays the output generated by the LDA in a clear and concise manner. The user is then able to browse through the different topics, adjust model parameters, and explore the visualization to find relevant information.

Implementation Documentation

Understanding the existing code base and file structure was a big challenge of our project because the code base is relatively large. Because the server on which the Text Mining System was initially hosted is temporarily offline, a significant amount of time was spent in configuring the file paths in order to be able to run the application locally as no one in our group was familiar with flask.The web pages are written in HTML with flask and various python libraries handling all of the functions and calls. The majority of code changes were to the following files:

- base.html contains the main page which links to Topic Mining page
- topic_mining.html contains the Topic Mining web page
- topic_modeler.py generates and returns the visualization to display as an html
- word2vec.py is runs the entire application

Usage Documentation

There are several ways to run the code depending on user preferences. Generally, we recommend creating a python virtual environment, installing pip and the relevant packages necessary to run word2vec.py using some sort of terminal or command line. If there is anything missing that is not covered by this list simply install whatever package is missing

- pip install --upgrade pip
- pip install virtualenv
- pip install flask
- pip install gensim
- pip install flask_session
- pip install pyLDAvis

- pip install nltk
- pip install pandas
- pip install metapy pytoml
- pip install matplotlib
- pip install pandas

When you are done, simply navigate to judel/tms and run python word2vec.py After successfully running it, navigate to http://localhost:6600. It may take some time to load the webpage, but when it does navigate to the Topic Mining tab which contains an upload button that allows the user to select a .txt file to upload for which to generate topics for. Clicking the “generate topic” button returns a visualization which has buttons to cycle through the topics, clear the topic and adjust the relevance metric. The user is also able to interact with the distance map and bar graph to obtain further information.

Contribution

Nivedita Chatterjee (nc19) - visualization development, back end functionality
Sudhendu Sharma (sharma78) - visualization development, back end & front end functionality
Yuriy Kotskyy (ykotsk2) - front end functionality, documentation, project proposal
Captain: Yung Chieh Huang (ych10) - front end functionality, presentation, progress report