Project Proposal

CS 410 Text Information Systems - Fall 2020

Team: The Electric Moccasins

James Coffey, NetID: jamesfc2 – Captain and Praveen Bhushan, NetID: bhushan6

We have chosen to reproduce a listed paper:

Hyun Duk Kim, Malu Castellanos, Meichun Hsu, ChengXiang Zhai, Thomas Rietz, and Daniel Diermeier. 2013. Mining causal topics in text data: Iterative topic modeling with time series feedback. In Proceedings of the 22nd ACM international conference on information & knowledge management (CIKM 2013). ACM, New York, NY, USA, 885-890. DOI=10.1145/2505515.2505612

The function of the tool that we will implement from this paper is to "combine probabilistic topic modeling with time series causal analysis to uncover topics that are both coherent semantically and correlated with time series data." (Kim et al.) This tool can be used to find positive and negative correlation of topics in textual data with time series data. This could be useful for example in predicting stock prices by a real user such as an investment bank. Our tool will be different in that we plan to use Python for implementation instead of the R implementation done in the paper. This will be impactful as Python is a general-purpose programming language and would make it easier to deploy in more programs. Lastly, the techniques/algorithms we will use are those in the paper: PLSA topic model, Pearson correlation coefficients, and Granger testing.

We can obtain the datasets used in the paper. We can get the New York Times Annotated Corpus from https://catalog.ldc.upenn.edu/LDC2008T19, IEM 2000 U.S. Presidential Election: Winner-Takes-All Market from https://iemweb.biz.uiowa.edu/closed/pres00_WTA.html, and historical stock prices for AAPL and AAMRQ from https://finance.yahoo.com/ and https://thestockmarketwatch.com/. Using this data, we can demonstrate the usefulness of our tool be replicating the results published in the original paper on the same datasets.

Our rough timeline for the proposed project is as follows:

- Nov. 1st – Implement paper in Python.
- Nov. 8th – Verify performance of against paper and compare results.
- Nov. 15th – Write interface for using tool on new data to generate results
- Nov. 22nd – Troubleshoot and bug fix.
- Nov. 29th – Submit progress report.
- Dec. 6th – Finish documentation and record presentation.
- Dec. 9th – Submit code with documentation and tutorial presentation.