Team Buddie

Topic: Reproducing a Paper (Casual Topic Mining)

Katie Shin, Sandeep Venkata, Aman Gupta

General Idea

- Can we combine probabilistic topic models and causal analysis with external time series to find topics in a corpus of documents that are both coherent semantically and correlated with the time series?
- Reproduced this paper by utilizing two data sets from the paper:
  - Corpus: NYT Articles
  - Time Series: Iowa Presidential Stock Markets
  - Looking for topics that specifically caused support for Bush or Gore to change.

Based on: 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 '13). Association for Computing Machinery, New York, NY, USA, 885–890. DOI:https://doi.org/10.1145/2505515.2505612

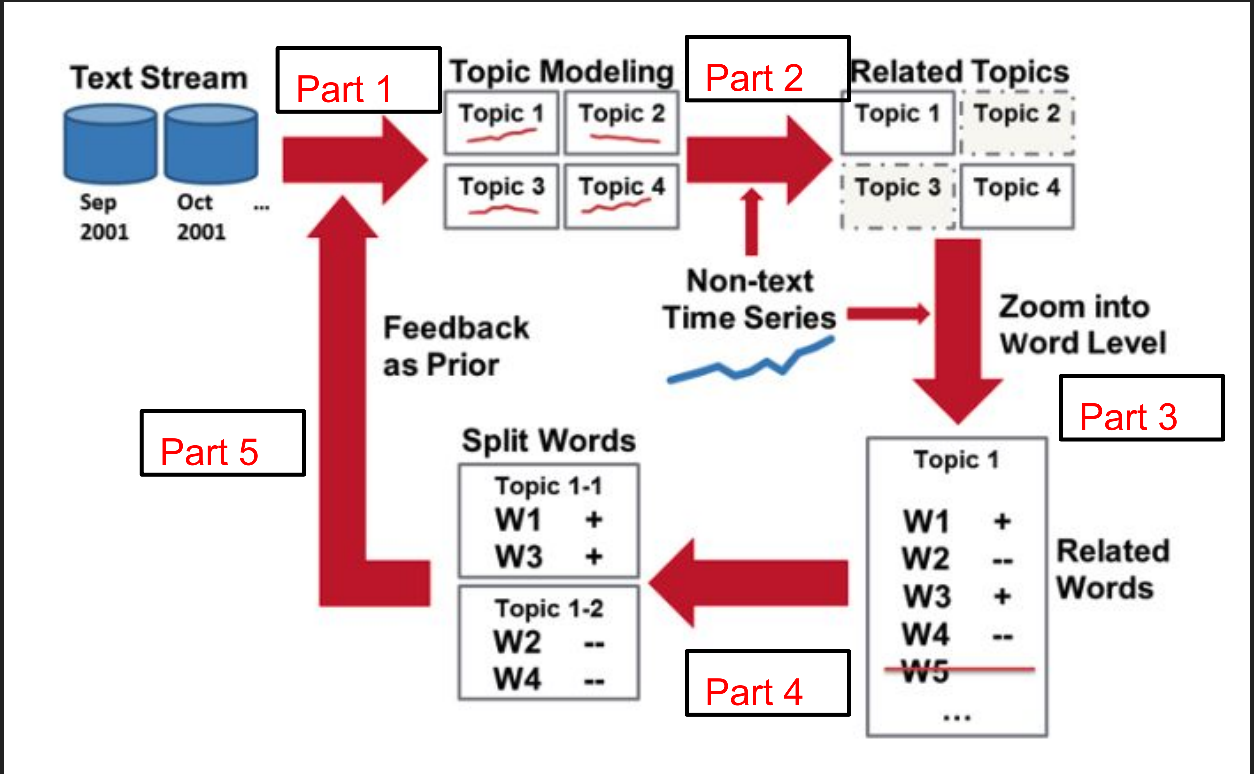
How to Use Software

- Pull from github repo https://github.com/97agupta/CourseProject
- Install the necessary libraries, with the most notable one being
  - pip install plsa
- Place NYT corpus in the correct folder format
  - Can be retrieved from https://catalog.ldc.upenn.edu/LDC2008T19
  - Folder: CourseProject/data/<month>/<day>/<filename>.xml

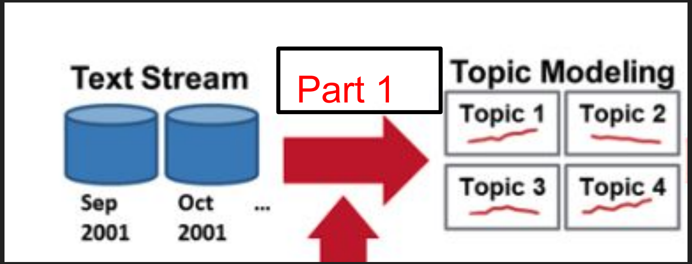
- To run the iterative code, run ‘python3 main.py’
- To run sections of the iterative code, run ‘python3 <filename>.py’

(Note: This takes hours to run, and therefore we have provided a sample run w/ end-results here)

Implementation



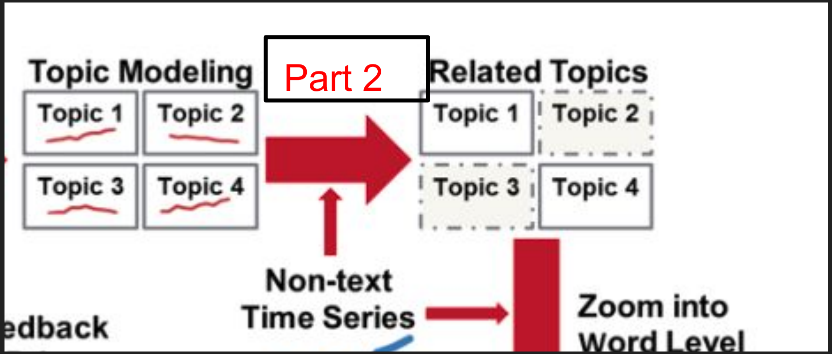
Implementation



- For each group of xml files that belong to a single day (05/01/2000 - 10/31/2000), we ran and picked the best model out of 5 PLSA models.
    - From the top model, we extract the top 5 highest probability words for each date that will be considered the topic.

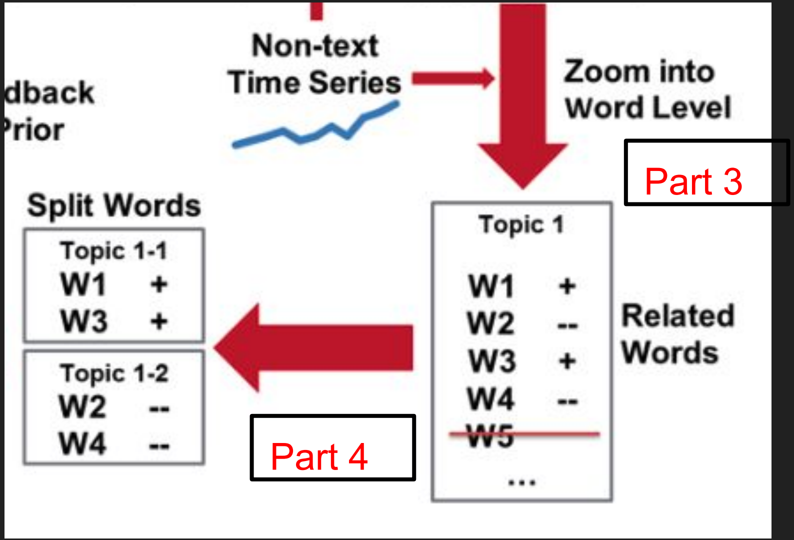
- Library used: PLSA

Implementation (Part 2)



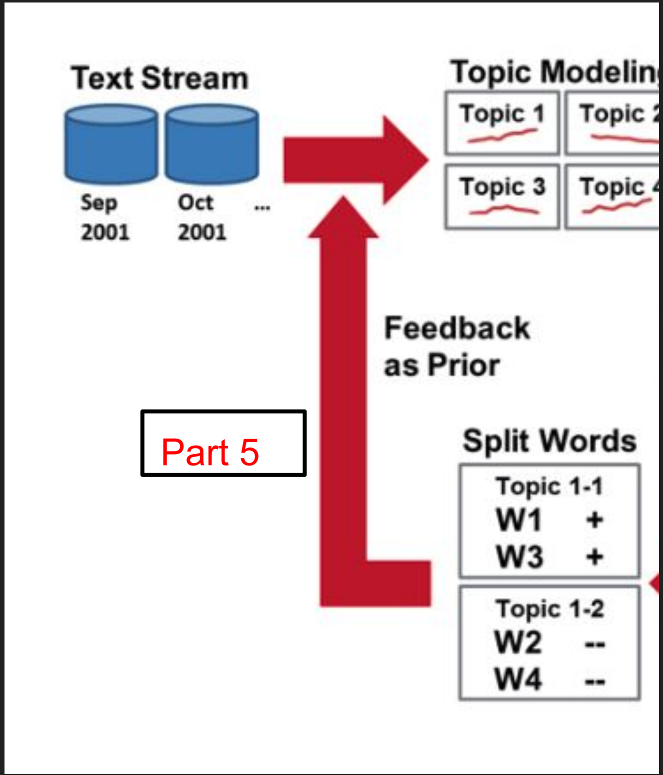
- Based on topics identified during PLSA, we identify related topics by running a Granger Test between the change in probability of topics and the normalized price of one candidate.
  - We use a lag of 5 days & create stationary time series for both of our Granger Test Variables.
  - Not every topic has enough data points for testing, so we skip over topics where the Granger Test fails
  - Based on the highest F-value for the each Granger Test, we determine if a topic is or isn't correlated with the external time series.

Implementation (Parts 3 & 4)



- From the best PLSA model (refer to Part1), we retrieved the top 20 relevant words (i.e. highest probability) per topic that were deemed related from the Granger test.
- For each word series, we run a pearson coefficient test comparing the external time series and the frequency of top words of each topic.
  - We then segment these words into negative and positive correlations, returning words that when combined, meet our probability threshold (0.75)
  - These words and then used as a Prior for the PLSA algorithm.

Implementation (Part 5)



- We used the PyPI package for PLSA but it didn’t offer support for including priors to guide the PLSA as per user inputs.
- So we enlisted the source code and extended the PLSA algorithm to overwrite the M step.
- The library essentially uses numpy ‘einsum’ to efficiently perform multi-dimensional matrix operations.
- We modified the M-step of the algorithm to include the parameters (μ) and the pseudo counts.
- The key part was to look for only the words that occured in the day.

Results (1st Iteration) PLSA without prior

- After running our first PLSA on the corpus of NYT documents, we found 343 topics. Of these topics, only 4 topics showed causality with our external time series: bush, gore, campaign, and clinton.
- For each of these topics, we extracted the top words and segmented each topic into two topics made up of positively and negatively influenced words. These new topics were used a prior for our second iteration. Here is an example of the Gore topic, being split into two:

gore _positive

	Word	Probability
0	debate	0.4671585084138510
1	oil	0.24000592562023600
2	bush	0.02037089961487880
3	teacher	0.018116499244389200

gore _negative

	Word	Probability
0	convention	-0.2445556313924940
1	party	-0.19336969916928100
2	delegate	-0.18026760013699100
3	missile	-0.07297529159633
4	stock	-0.006330717162448750

Results (2nd Iteration)

- With a new iteration using PLSA with prior, we find the following relevant topics using our granger test:
  - Teacher, oil, drug, debate

```
for key in relevent_topic:
    print(key)
```

```
teacher
drug
oil
debate
```

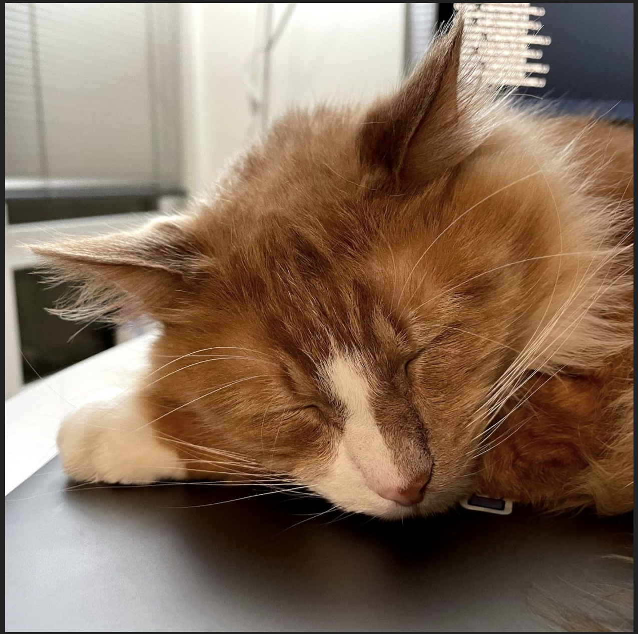
- This is a clear improvement over our original PLSA, as it delves deeper into the issues that moved the campaign likelihoods for each candidate
  - Bush, Gore, Campaign, Clinton

-- This clearly shows that our model is finding topics that are more related to the campaign issues & with further iterations we expect this to further improve.

Next Steps

- We would want to run further iterations on these topics.
- Additionally, instead of using the full NYT corpus we could identify paragraphs that mention Bush, Gore, or Presidential to create a smaller corpus from which to topic mine.
- We would want to find a more robust method of comparing and understanding the different F-Scores from each lag of our Granger Test.

Thanks!



(This is Buddie)