Event Detection in Historic Text Data

Sabrina Peng, Morgan VandenBerg

Problem Overview

Using natural language processing techniques on historic text data can be valuable in determining what causes significant historical events. Specifically, issues, ideas, and sentiments can suddenly become viral and become triggers for influential events. This project uses time series mining, topic modeling, and unsupervised machine learning techniques on British parliamentary documents to detect what viral events occurred in Britain during the 19th century.

This research can be particularly valuable for historians who are looking to extract relevant information from corpora of data that are too large to be manually analyzed. Specifically, these techniques can help uncover latent information about the influence of ideas on historical events that even domain experts may not be immediately aware of. Events are "commonly considered as the building blocks of historical knowledge with which historians construct their system of ideas about the past."[1] Doing automatic event detection can not only help historians validate their existing research and beliefs, but also potentially introduce new evidence that contributes to our understanding of historical causality.

The corpus is a set of approximately 11 million sentences extracted from House of Commons parliamentary speeches spanning the time period from 1803-1910. Along with the speech text, associated metadata that is relevant include the speech ID and speech date. The speech ID was used to group speech sentences into documents, while the speech date was used to generate pertinent time series on the basis of both year and month.

Experimental Setup

1. A stratified subsampling by speech date was required due to the large size of the dataset.
    a. Each generated sample contained a fixed, user-specified number of speech sentences from every unique year for which speech data was provided in the corpus. The sampling was done by shuffling all sentences in a given year and using the last *x* number of sentences as the representative sample for the year.
    b. Further subsampling was done by randomly sampling a specified number of speeches from every month, which achieved a higher degree of granularity and a more even distribution of data within each year.
2. Stopwords from the nltk corpus English stopwords list and a domain-specific list of government stopwords were removed from the text data.
3. Texts were filtered by parts of speech using *nltk*'s default part of speech tagger (pos_tag) and lemmatized using *nltk*'s WordNetLemmatizer.

[1] https://www.mitpressjournals.org/doi/full/10.1162/coli_a_00347

4. Our evaluation metric was the comparison of events detected by our research to actual historical events in Britain during 1803-1910, knowledge of which can be supplied by domain experts in the SMU history department.

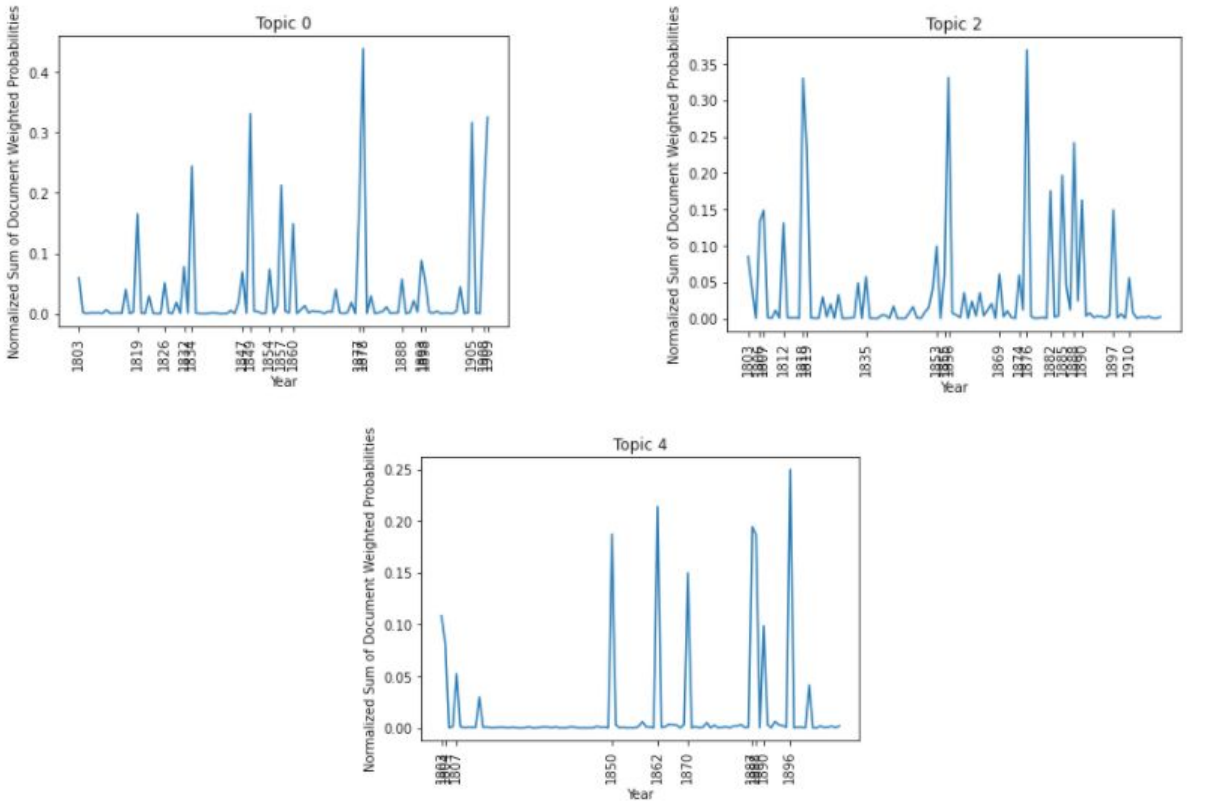
Experiments

Basic LDA Modelling

The first experiment conducted made use of LDA, or Latent Dirichlet Allocation, topic modelling. LDA is a generative probabilistic model often used on text corpora, where documents are represented as random mixtures over latent topics and each topic is characterized by a distribution over words.[2] The implementation of LDA used was from *gensim*. Note that instability between runs is a known weakness of LDA models - as such, the results presented using LDA modelling in this project are for a specific run.

Topic models were created by specifying the number of topics desired. We used n = 50, 100, and 200. For all topics created, better representations of those topics were created by removing common terms from word-probability pairs that showed up in more than 10 topics, therefore achieving more specific distillations for all topics. For each topic generated, the sum of probability weights of speeches was computed for each year, normalized by the number of speeches in the year, and created into a time series.

[2] http://ptrckprry.com/course/ssd/reading/Blei03.pdf



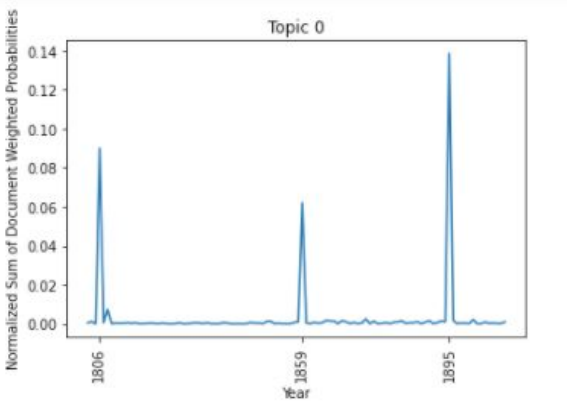

```
Topic 0 Key Terms:
['limitation', 'area', 'cut', 'plantation', 'slavetrade', 'convenient', 'requirement', 'seize', 'infringe', 'adjoin']

Topic 2 Key Terms:
['fraud', 'baron', 'prediction', 'corpus', 'camp', 'outline', 'condemn', 'institute', 'insurrection', 'northcote']

Topic 4 Key Terms:
['corp', 'vacancy', 'ignorant', 'stafford', 'northcote', 'thousand', 'woman', 'readiness', 'servant', 'compose']
```

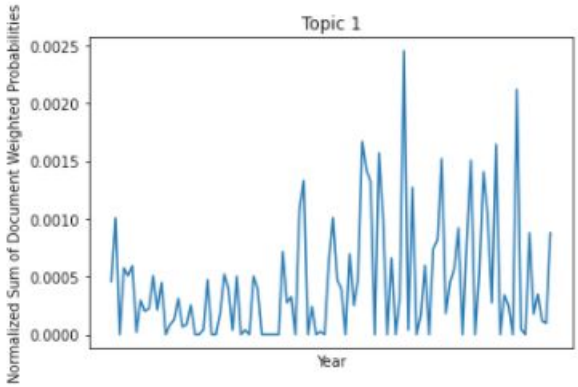
Fig. 1: Sum of weighted probabilities for speeches across time for topics 0, 2, and 4 generated from an LDA model with number of topics = 50, accompanied by distilled representations of each topic.

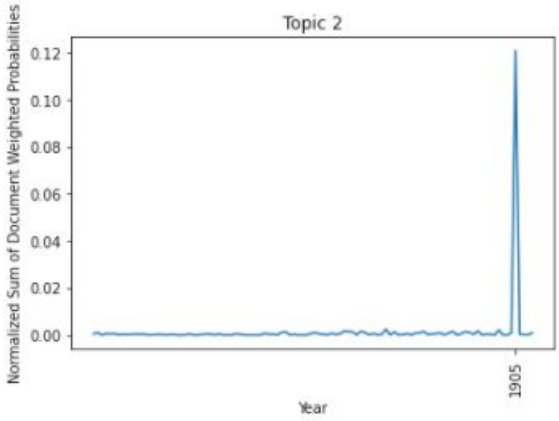
These time series were plotted to give a better visualization of how the sum of weighted probabilities for speeches varied across the entire time period. The plots show that there exists variation across topics in terms of the volatility of the changes over time and how often the volatility occurs.

The plots also provide some preliminary results via human observation that can be compared to actual historical events. For example, the key terms for topic 4, n = 50 include the unique terms “stafford” and “northcote,” which happen to be the first and last name of a British Conservative politician who served Britain in a variety of government positions from 1851 to 1885. Looking at the plot for topic 4, the years 1850, 1862, and 1870 are distinct peaks in the time series data, which may indicate significant discontinuity in those years relating to Stafford Northcote. Indeed, doing some brief research to validate those preliminary conclusions show that Stafford Northcote entered British politics in 1851, published a

book on finance in 1862, and represented British interests in a few business deals with North America around 1870.[3] Though this may be cherry-picking results out of the data, these results can be submitted to domain experts in the SMU history department for further evaluation. More examples of topic plots and corresponding related key terms are shown below for an LDA topic model with n = 100 topics.








```
Topic 0 Key Terms:
['congress', 'dividend', 'williams', 'evasion', 'fairness', 'partiality', 'reckon', 'limitation', 'disapprobation',
'scruple']
Topic 1 Key Terms:
['rumour', 'interpretation', 'destiny', 'newspaper', 'utterance', 'pardon', 'contradict', 'eligibility', 'judicatur
e', 'adorn']
Topic 2 Key Terms:
['locality', 'harm', 'beater', 'stop', 'plantation', 'train', 'couple', 'landowner', 'fence', 'proclaim']
```

Fig. 2: Sum of weighted probabilities for speeches across time for topics 0, 1, and 2 generated from an LDA model with number of topics = 100, accompanied by distilled representations of each topic.

[3] https://en.wikipedia.org/wiki/Stafford_Northcote,_1st_Earl_of_Iddesleigh

Change Point Detection

Change point detection is the detection of abrupt variations in time series data. The original academic paper which we referenced on change point detection was written about detecting events in sensor data; namely, identifying different levels of traffic from sensors in roads which report a great deal of "noise" (oscillations) but contain inherent, harder-to-identify signals.

Algorithms for change point identification can also be useful in detecting where viral events happen, as viral events represent significant changes in history. The specific version of change point detection explored in this project is *offline* change point detection, which considers an entire dataset at once and reviews the data as a whole to find breakpoints. This is in contrast to *online* change point detection, which detects change points as incoming, real-time data arrives and is processed.[4]

To do change point detection, time series must first be generated. As mentioned earlier, we generated time series from the basic LDA modelling technique by summing document weighted probabilities for each topic over the time period. The *ruptures* package for Python was used to perform offline change point detection.[5] The ruptures package includes five different algorithms for detecting change points: dynamic programming exact segmentation, Pelt exact segmentation, binary segmentation, bottom-up segmentation, and window-based change point detection. All five algorithms require different parameters - including the type of segment model, the number of breakpoints, the penalty cost, minimum segment length, and subsample frequency, among others. For each algorithm, an instance of the algorithm is generated using the provided parameters, fit to the time series, and can be used to predict optimal breakpoints.

- Dynamic programming exact segmentation
    - Computes the best partition for which the sum of errors is minimum for a given segment model using dynamic programming
- Pelt exact segmentation
    - Computes the segmentation which minimizes the constrained sum of approximation errors for a given segment model and penalty level
    - Computational complexity is linear on average
- Binary segmentation
    - Computes optimal breakpoints using a greedy sequential approach in which time series data is split around detected change points into two segments and detection is repeated on the resulting sub-signals until a stopping criteria is fulfilled
    - Computational complexity is O(n log n)
    - Useful in instances where true breakpoints are unknown
- Bottom-up segmentation
    - Computes optimal breakpoints using a generous sequential approach that starts with many change points and successively deletes less significant ones
    - Computational complexity is O(n log n)

[4] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5464762/
[5] https://ctruong.perso.math.cnrs.fr/ruptures-docs/build/html/index.html

    - Useful in instances where true breakpoints are unknown
- Window-based change point detection
    - Computes optimal breakpoints using two sliding windows across the time series and calculations of discrepancy measures that determine whether a boundary between windows is a change point

Below is a visualization of the breakpoints of the five change point detection algorithms described above, plotted against the relevant time series data.

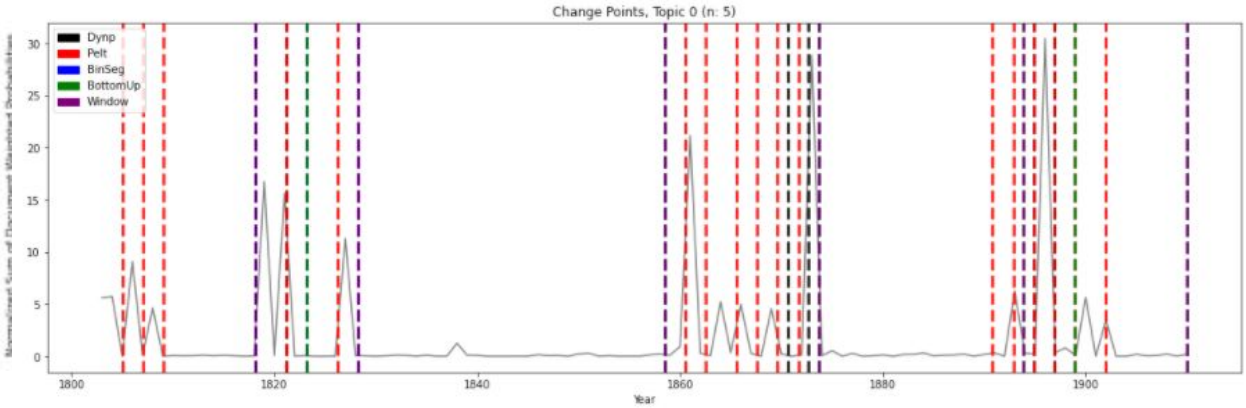


Fig. 3: DP segmentation, Pelt segmentation, binary segmentation, bottom-up segmentation, and window-based change point detection on sum of weighted probabilities for speeches over time.

Analysis of this visualization shows that the number and location of breakpoints detected by the algorithms depends heavily on the parameters used in the algorithm instances. Together, the breakpoints detected by all algorithms capture most of the edges of significant peaks in the time series, indicating that change point detection algorithms using *ruptures* can be effectively applied to time series from topic modelling, if parameters are selected carefully.

Discontinuities Across Topics

The time series plots above mapped discontinuity against time only for a single topic. In order to capture discontinuities across all topics from the LDA model, the number of breakpoints detected by the Pelt algorithm from the *ruptures* package was summed across all generated topics and plotted against the time period, resulting in the following plot.

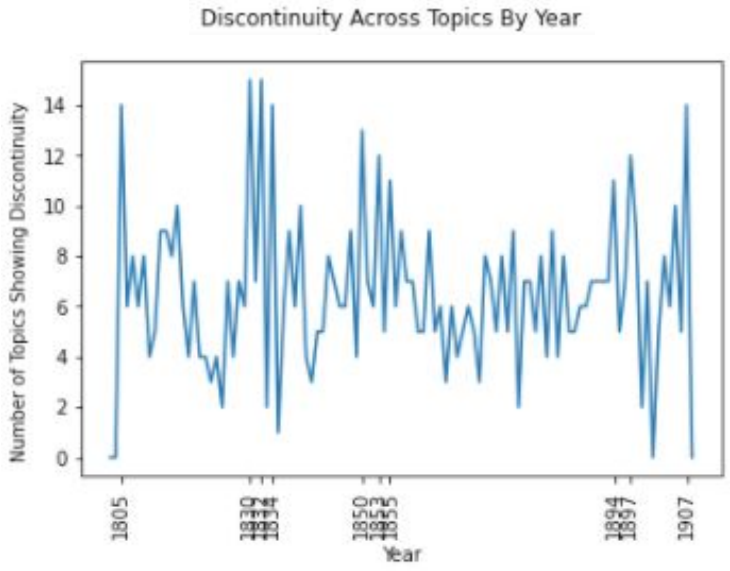


Fig. 4: Number of breakpoints detected by the Pelt change point detection algorithm for each year across all generated topics for LDA with n = 50 topics. Years that had more than 10 out of 50 topics produce a breakpoint are marked on the x-axis.

The plot gives us a better sense of what years showed the most discontinuity across all topics, indicating more "change" occurring during that year over many different realms of British political concerns. In particular, the early 1830s and 1850s stood out as time periods in which significant discontinuity was occurring in over 20% of the generated topics.

In addition, the new time series data was examined in order to determine if there existed discontinuity for the same topics in different years, which could indicate that similar viral sentiments and events occurred during those years. For example, 1825 and 1827 had almost the same set of topics show discontinuity in those years. Though the proximity in these two years may play a role in the similarity of topics, this could be an indication that a strong sentiment or event that made an appearance in 1825 could have resurfaced again in 1827, providing historians with more insight into how and why similar ideas re-emerge over time given certain historical circumstances.

Topic Dissimilarity Over Time

To extend the exploration of the historical dataset, another experiment approach was conducted, in which speech data was aggregated by both month and year and an LDA topic model was built for each subset of data pertaining to the given time slice (for a total of 711 unique topic set models). Creating an LDA model for each time slice allows for comparison of topics across time slices by calculation of month-after-month topic set similarities. Three different topic set similarity metrics were used to compare the topic sets between time slices.

1. Jaccard similarity coefficient: a statistic used for gauging the similarity and diversity of finite sample sets and is defined as the size of the intersection divided by the size of the union of the sample sets.[6]
2. Word embedding-based similarity: a word embedding-based metric that more fully captures the semantic similarity between topics. This metric represents each topic by its top *n* words (ranked by the words' posterior topic probabilities) and is computed by summing together all pairwise word semantic cosine similarity values between corresponding terms and topics in the two topic sets.[7]

$$WES(\theta_i, \theta_j) = \sum_{p \in W_i} \min_{\forall q \in W_j} cosine(Vec_p, Vec_q)$$

3. Aggregated embedding-based similarity: a word embedding-based metric that aggregates each topic in a topic set into a single vector (a summation of the words), normalizes the topic vectors, then computes vector cosine similarity of each topic vector to corresponding topic vectors in the other topic set.

Each of these metrics were used to compute topic set similarity from month to month and generate time series for further exploration and change point detection. Plotting the time series over the given period of time can help us identify low topic set similarity between years, indicating possible change points. The time series generated by aggregated embedding-based similarity is shown below.

[6] https://en.wikipedia.org/wiki/Jaccard_index

[7] https://www.researchgate.net/publication/331175095_Evaluating_Similarity_Metrics_for_Latent_Twitter_Topics

This exact approach used a year-to-year similarity metric - each data point is the similarity between topic sets T_a and T_b, where T_a corresponds to month/year t_k and T_b corresponds to month/year t_{k+1}. We propose, but did not implement, an alternative metric of windowed moving averages for similarity-over-time measures. This could be particularly useful for detecting longer-term changes.

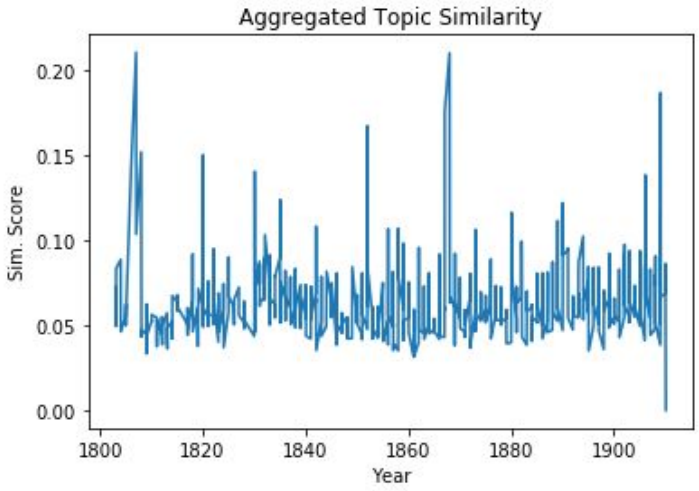


Fig. 5. Example of aggregated word embedding-based similarity score plotted over 1803-1910; Jaccard metric.

Change in Word Weights

Another related approach was looking at the change of weights for a specific word in topics from month to month. The weights for the specified word were aggregated for the topic set models for each month and plotted against the entire time period. The frequency of the term in the documents for each month was then plotted against this time series as a comparison, since we expected the aggregation of the term weights to mirror the term frequency values. However, discrepancies in the word weight and word frequency time series could suggest unexpected significance. It is theoretically possible for a topic model to detect important terms and ideas that are not shown in word-frequency count analyses. This would indicate the subtle significance of a term.

Visualizing the word weights and term frequency time series can help us identify when certain user-specified terms were most prominent in the corresponding time slice's topic sets. For example, the plot below showing the use of the term "India" shows that the term was heavily featured in parliamentary speeches during the 1850s and early 1910s, which could indicate increased relations or conflict between Britain and India during those time periods.

It can be verified that the below graph mirrors historical trends. In 1858, Britain conquered India--and there is a huge spike in the shown signal at approximately that time.

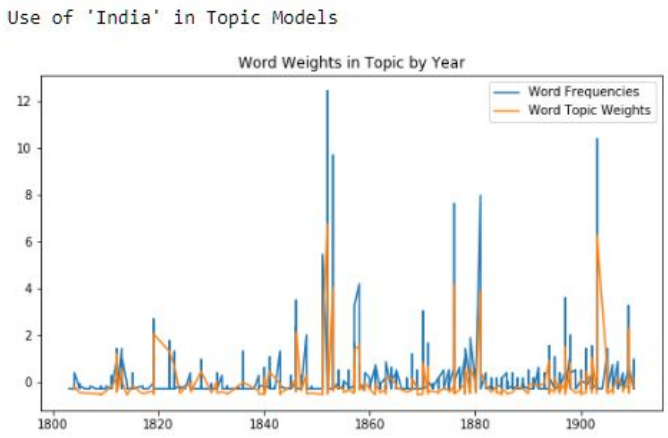


Fig. 6: Word topic weights in topic by year and word frequencies plotted for topic sets over the time period for the term "India."

PCA Element Change Point Detection

Principal component analysis, or PCA, is an unsupervised statistical technique that allows for dimensionality reduction in machine learning. PCA allows for projection of the original large dataset, with many features, into a representation with fewer dimensions by retaining only a smaller subset of principal components as features. This technique is useful for preventing overfitting, allowing machine learning models to more correctly generalize, and making models more computationally efficient to run.[8]

For our experiment, we generated aggregated topic vectors by year by summing the vectors of the top *k* words from the top *n* topics in each month/year set (note that *k* and *n* are hyperparameters which could be tuned in future research). Then, PCA reduction was conducted on the aggregated vectors, which reduced the dimension of the vectors and embeddings from 300 components (the shape of the Google News vectors) to 48 components. This maintained 80% of the explained variance, and was intended to output a smaller number of graphs which can be inspected by humans for apparent change point and term importance detection. We plot the numerical value of each reduced embedding for each month/year combination over time (noting that this single-dimensional scalar has expanded meaning due to the PCA reduction).

From there, we can choose items for which there seems to be some significant signal. Those items can be reconstructed to a 300-dimensional word vector; then, the original word embeddings can be used with cosine similarity to approximate the "meaning" of that individual reconstruction. We use the *ruptures* algorithms to perform change point detection on the reduced vectors, and provide their approximate terms from the reconstructions above.

An example graph of the embedding dimension (index 21) is shown below.

[8] https://medium.com/apprentice-journal/pca-application-in-machine-learning-4827c07a61db

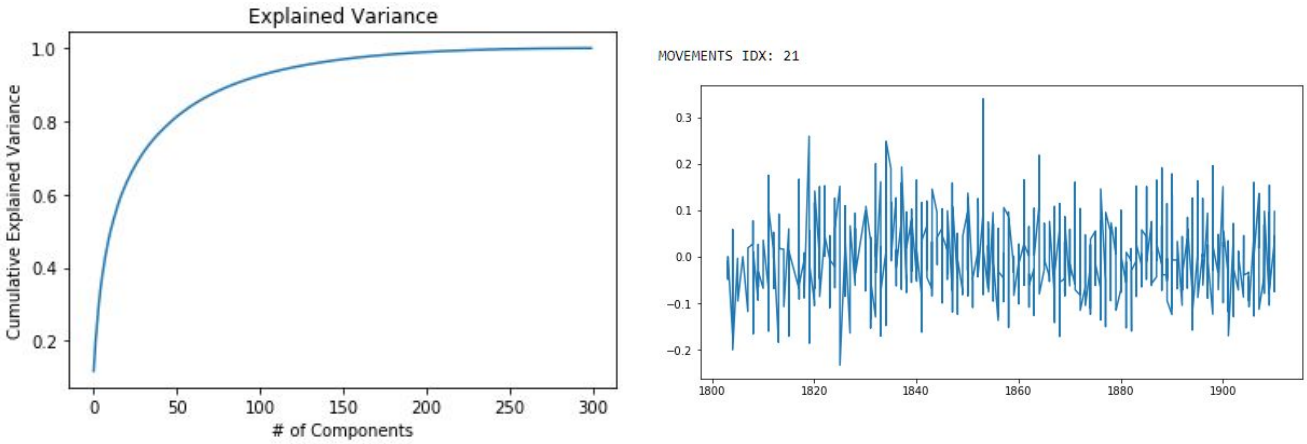
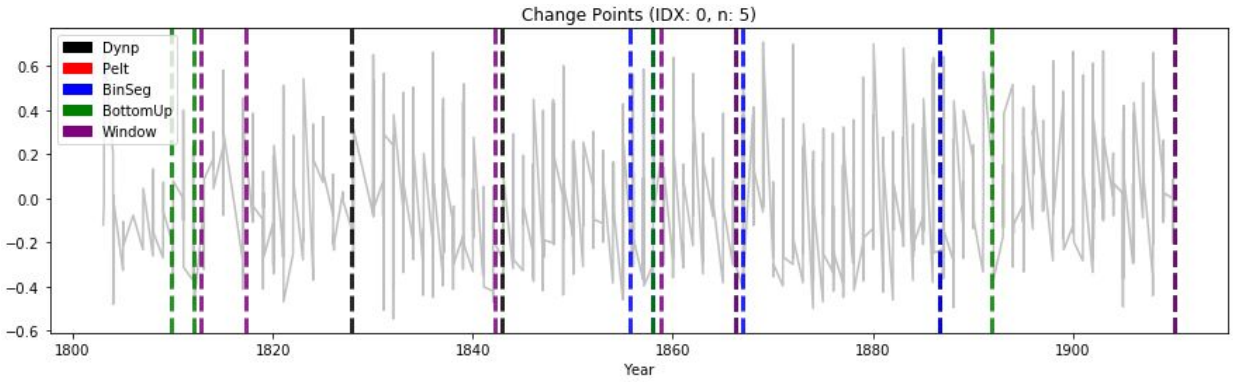


Fig. 7: The plot on the left shows the cumulative explained variance captured by PCA as the number of components increases. The plot to the right shows the vectorized topics by year after PCA reduction was conducted for the 20th PCA component.

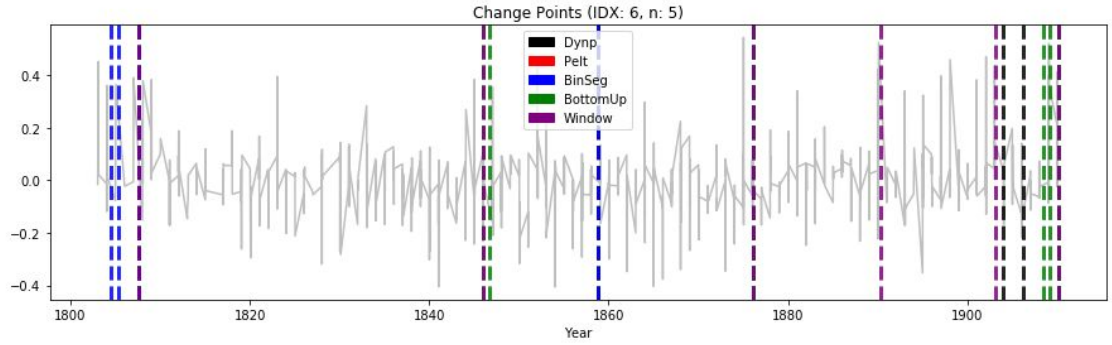
The five change point algorithms were run on the PCA-reduced topic vectors and plotted over the time period. The following plot is an example of the break points that were detected for a component with significant terms “lord,” “house,” and “ship.”




```
('lord', 0.6021789312362671), ('house', 0.5635101795196533), ('ship', 0.5448867082595825)
```

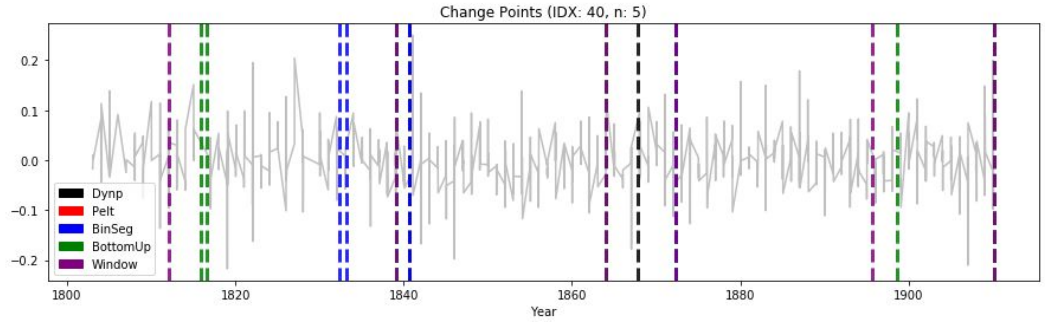
The below change point detection plot and corresponding reconstructed terms set shows that the change points detected may correspond to significant events during the time period related to time-based scheduling, based on the presence of the terms “year,” “month,” and “week.”

```
INDEX: 6 associated topics:  [('year', 0.6880605220794678), ('month', 0.5398091673851013), ('week', 0.4769560694694519), ('Giun
ta_hitch', 0.47373974323272705), ('Icelanders_overwhelmingly', 0.46956366300582886), ('Rautaruukki_Oyj_Rautaruukki', 0.4674089550971985), ('Rupee_surges', 0.46519577503204346), ('By_CITIZEN_STAFF', 0.46149754524230957), ('faking_comprehension', 0.46144935488700867), ('symbol_DDF', 0.4607742428779602)]
```



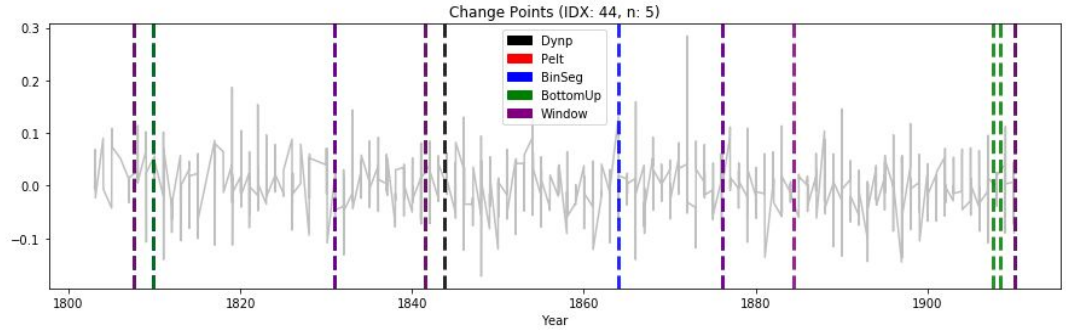
The below change point detection plot and corresponding reconstructed terms set shows that the change points detected may correspond to significant events during the time period related to housing or building.

```
INDEX: 40 associated topics:  [('house', 0.945417046546936), ('houses', 0.7071628570556641), ('bungalow', 0.6849117279052734),
('apartment', 0.6303699612617493), ('residence', 0.6255773305892944), ('bedroom', 0.6063220500946045), ('mansion', 0.6055839657
783508), ('townhouse', 0.5925908088684082), ('farmhouse', 0.5907357931137085), ('home', 0.5665984153747559)]
```



The below change point detection plot and corresponding reconstructed terms set shows that the change points detected may correspond to significant events during the time period related to religion, specifically the Catholic Church.

```
INDEX: 44 associated topics:  [('catholic', 0.8027719259262085), ('Catholic', 0.6132189035415649), ('Roman_Catholic', 0.5905075
073242188), ('catholicism', 0.5685599446296692), ('athiest', 0.5624523758888245), ('catholic_church', 0.561346173286438), ('pro
testant', 0.5586223006248474), ('chruch', 0.5531352162361145), ('christian', 0.5517715811729431), ('catholics', 0.5473130941390
991)]
```



Conclusion

In this project, we approached the problem of event detection in historic text data using a variety of techniques. We started by generating one LDA topic model across all years and using the sums of probability weights of speeches in each year for each topic generated as time series for analysis and evaluation. Visualizing these time series allowed for manual observation of time periods in which given topics were more prevalent in the associated speech documents. In addition, we removed common terms across topics from each topic's term-probability representations in order to distill topics down to differentiated concepts. Five change point detection algorithms from the *ruptures* package were used on the generated time series to produce optimal breakpoints, which can be evaluated for accuracy by consulting with domain experts. In order to capture discontinuity across all topics, the number of breakpoints was summed across all topics for every year, giving us a better sense of what years showed the most discontinuity across all topics.

Next, we explored topic dissimilarity over time by generating topic models for each time slice and using three different topic set similarity metrics (Jaccard similarity coefficient, word embedding-based similarity, and aggregated embedding-based similarity) to compute topic set similarity over time. Low topic similarity between given years could indicate significant historical changes and events. Change in word weights for specific terms plotted against the frequency of those terms allowed us to identify when certain terms were more prominent during the overall time period and express detected historical events on the basis of those terms. Finally, PCA was conducted on aggregated topic vectors to reduce dimensionality by around 80%, the reduced embeddings for each principal component were plotted across time, and change point detection algorithms were used to detect years in which each component's corresponding reconstructed terms showed significant change.

More specific and detailed results will be discussed with domain experts in a separate meeting in order to validate and evaluate the accuracy of our techniques.

Future Work

One approach that extends our understanding of LDA models is the use of the LDA Sequence model, which David M. Blei and John D. Lafferty developed in order to analyze the time evolution of topics in large document collections.[9] The LDASeqModel, available via Gensim, allows latent topics to evolve over time based on context from the pertinent topic in the previous time slice implemented using approximate posterior inference. This could be useful in analyzing how certain topics defined at the beginning of the time period in 1803 evolve over the years, and change point detection could be run on time series generated from the sequential topic model, which have been proven by Blei and Lafferty to perform better than static topic models. This approach was actually tested, but the posterior inference calculations are expensive and were not able to be run on the computing machine. In the future, we may test this on a more powerful server or on ManeFrame.

In addition, embedding-aware topic models such as the Dirichlet Multinomial Mixture model and the Latent Feature Topic Model could be explored. These topic models extend LDA by incorporating latent feature vector representations of words trained on large corpora (word embeddings) in order to achieve better word-topic mappings for smaller corpora.[10] Experiments done by Dac Quoc Nguyen and his fellow authors show that the information from the external corpora improves performance for topic coherence, document clustering, and document classification texts for corpora with short documents.

Our existing machine learning methods may also benefit from the use of larger sample sizes and hyperparameter tuning. Due to performance constraints, stratified subsampling was used to get a representative sample of data across all time periods, but using more data from the corpus would allow us to capture more context in our conclusions. Hyperparameter tuning is also necessary in order to find optimal hyperparameters for our models that minimize loss function on our data.

Transfer learning with word embeddings, in which models originally trained for a specific task are used on other tasks, is also another possible approach to the problem. This could be useful if the models were trained on datasets with similar historical vernacular and vocabulary, which could yield better results on the current dataset.

Finally, further consultation with domain experts would be ideal in order to explore the detected discontinuities and verify the effectiveness of the implemented natural language processing techniques.

[9] https://mimno.infosci.cornell.edu/info6150/readings/dynamic_topic_models.pdf
[10] https://transacl.org/ojs/index.php/tacl/article/view/582/158