Final Project Notebook

DS 5001 Exploratory Text Analytics | Spring 2024

Metadata

- Full Name: Ryan Kim
- Userid: rjk9tt
- GitHub Username: rkim422
- GitHub Repo URL: https://github.com/rkim422/DS5001_Final_Project
- UVA Box URL: https://virginia.box.com/s/bffpfu6xdd8yumvezzrhdbv7gsrndkqo

Overview

The goal of the final project is for you to create a **digital analytical edition** of a corpus using the tools, practices, and perspectives you've learning in this course. You will select a corpus that has already been digitized and transcribed, parse that into an F-compliant set of tables, and then generate and visualize the results of a series of fitted models. You will also draw some tentative conclusions regarding the linguistic, cultural, psychological, or historical features represented by your corpus. The point of the exercise is to have you work with a corpus through the entire pipeline from ingestion to interpretation.

Specifically, you will acquire a collection of long-form texts and perform the following operations:

- **Convert** the collection from their source formats (F0) into a set of tables that conform to the Standard Text Analytic Data Model (F2).
- **Annotate** these tables with statistical and linguistic features using NLP libraries such as NLTK (F3).
- **Produce** a vector representation of the corpus to generate TFIDF values to add to the TOKEN (aka CORPUS) and VOCAB tables (F4).
- **Model** the annotated and vectorized model with tables and features derived from the application of unsupervised methods, including PCA, LDA, and word2vec (F5).
- **Explore** your results using statistical and visual methods.
- **Present** conclusions about patterns observed in the corpus by means of these operations.

When you are finished, you will make the results of your work available in GitHub (for code) and UVA Box (for data). You will submit to Gradescope (via Canvas) a PDF version of a Jupyter notebook that contains the information listed below.

Some Details

- Please fill out your answers in each task below by editing the markdown cell.
- Replace text that asks you to insert something with the thing, i.e. replace `(INSERT IMAGE HERE)` with an image element, e.g. `![](image.png)`.
- For URLs, just paste the raw URL directly into the text area. Don't worry about providing link labels using `[label](link)`.
- Please do not alter the structure of the document or cell, i.e. the bulleted lists.
- You may add explanatory paragraphs below the bulleted lists.
- Please name your tables as they are named in each task below.
- Tasks are indicated by headers with point values in parentheses.

Raw Data

Source Description (1)

Provide a brief description of your source material, including its provenance and content. Tell us where you found it and what kind of content it contains.

The source was found on the Box suggested material from Professor Alvarado. The corpus included more words than the maximum so a sample was taken to reduce computation time. The content of the corpus included various news articles between 2013 and 2020 from various sources. Each document included 1 "paragraph" of content.

Source Features (1)

Add values for the following items. (Do this for all following bulleted lists.)

- Source URL: https://virginia.app.box.com/s/bj8f1khrkfd6thm9umq35m6xp2an4zej Given by Professor Alvarado
- UVA Box URL: https://virginia.box.com/s/x2c7xd8fy81y9zg2eo6hilbh2avsjo59, https://virginia.box.com/s/64afgxqbgdfqigk78vpy1kwnfb7x4e1e
- Number of raw documents: 1,026,347, randomly sampled 50,000
- Total size of raw documents (e.g. in MB): 431.6, 21.1
- File format(s), e.g. XML, plaintext, etc.: csv

Source Document Structure (1)

Provide a brief description of the internal structure of each document. That, describe the typical elements found in document and their relation to each

other. For example, a corpus of letters might be described as having a date, an addressee, a salutation, a set of content paragraphs, and closing. If they are various structures, state that.

The original corpus included 6 columns: doc_id, doc_source, doc_title, doc_content, doc_date, and doc_url. Each document was formatted with the same structure, each having one of the above variables. While initially, I thought the titles were unique along with the ids, I found that many titles were repeated, therefore, the ids were necessary to uniquely identify the documents.

Parsed and Annotated Data

Parse the raw data into the three core tables of your addition: the `LIB`, `CORPUS`, and `VOCAB` tables.

These tables will be stored as CSV files with header rows.

You may consider using `|` as a delimitter.

Provide the following information for each.

LIB (2)

The source documents the corpus comprises. These may be books, plays, newspaper articles, abstracts, blog posts, etc.

Note that these are *not* documents in the sense used to describe a bag-of-words representation of a text, e.g. chapter.

- UVA Box URL: https://virginia.box.com/s/htu2lnulet7n92ryddket3rhhjlnxinc
- GitHub URL for notebook used to create: https://github.com/rkim422/DS5001_Final_Project/blob/main/parsing_newzy.ipynb
- Delimitter: ,
- Number of observations: 50,000
- List of features, including at least three that may be used for model summarization (e.g. date, author, etc.): doc_source, doc_title, doc_date, year, doc_url, num_chars
- Average length of each document in characters: 227.35344

CORPUS (2)

The sequence of word tokens in the corpus, indexed by their location in the corpus and document structures.

- UVA Box URL: https://virginia.box.com/s/mvf424zcabtlvfqcvml1e8bvin6ephjd

- GitHub URL for notebook used to create: https://github.com/rkim422/DS5001_Final_Project/blob/main/parsing_newzy.ipynb
- Delimitter: ,
- Number of observations Between (should be >= 500,000 and <= 2,000,000 observations.): 1,786,062
- OHCO Structure (as delimitted column names): doc_source, doc_id, sent_num, token_num
- Columns (as delimitted column names, including `token_str`, `term_str`, `pos`, and `pos_group`): pos, token_str, term_str, pos_group

VOCAB (2)

The unique word types (terms) in the corpus.

- UVA Box URL: https://virginia.box.com/s/6mt1ox35aansk0hy69a5ebazbppfq6vm
- GitHub URL for notebook used to create: https://github.com/rkim422/DS5001_Final_Project/blob/main/parsing_newzy.ipynb
- Delimitter: ,
- Number of observations: 78,198
- Columns (as delimitted names, including `n`, `p'`, `i`, `dfidf`, `porter_stem`, `max_pos` and `max_pos_group`, `stop`): n, n_chars, p, i, max_pos, max_pos_group, porter_stem, stop, dfidf
- Note: Your VOCAB may contain ngrams. If so, add a feature for `ngram_length`.
- List the top 20 significant words in the corpus by DFIDF.

watson, reservations, telephone, centered, cent, 60s, gig, rip, cement, cells, cellphones, temptation, rid, tenants, celebrates, richmond, globally, tennis, wrongful, tent

Derived Tables

BOW (3)

A bag-of-words representation of the CORPUS.

- UVA Box URL: https://virginia.box.com/s/447076uvuryxk4pgkmn68zgaeb4q6036
- GitHub URL for notebook used to create: https://github.com/rkim422/DS5001_Final_Project/blob/main/derived_tables.ipynb
- Delimitter: ,
- Bag (expressed in terms of OHCO levels): [doc_source]
- Number of observations: 196,720
- Columns (as delimitted names, including `n`, `tfidf`): n, tfidf

DTM (3)

A represenation of the BOW as a sparse count matrix.

- UVA Box URL: https://virginia.box.com/s/konbdjs6c1p84ik507pqblbiodsavkla
- UVA Box URL of BOW used to generate (if applicable): https://virginia.box.com/s/447076uvuryxk4pgkmn68zgaeb4q6036
- GitHub URL for notebook used to create: https://github.com/rkim422/DS5001_Final_Project/blob/main/derived_tables.ipynb
- Delimitter: ,
- Bag (expressed in terms of OHCO levels): [doc_source]

TFIDF (3)

A Document-Term matrix with TFIDF values.

- UVA Box URL: https://virginia.box.com/s/cgzcquc5wobdwdiad673zny2ya8sloz2
- UVA Box URL of DTM or BOW used to create: https://virginia.box.com/s/447076uvuryxk4pgkmn68zgaeb4q6036
- GitHub URL for notebook used to create: https://github.com/rkim422/DS5001_Final_Project/blob/main/derived_tables.ipynb
- Delimitter: ,
- Description of TFIDIF formula: I used the 'max' TFIDF method. This means that the maximum TFIDF value for the given word in the documents was used when evaluating the importance.

Reduced and Normalized TFIDF_L2 (3)

A Document-Term matrix with L2 normalized TFIDF values.

- UVA Box URL: https://virginia.box.com/s/6hibiipjanzulfueryh665fr8iyd5isv
- UVA Box URL of source TFIDF table: https://virginia.box.com/s/cgzcquc5wobdwdiad673zny2ya8sloz2
- GitHub URL for notebook used to create: https://github.com/rkim422/DS5001_Final_Project/blob/main/derived_tables.ipynb
- Delimitter: ,
- Number of features (i.e. significant words): 1,000
- Principle of significant word selection: I filtered the TFIDF based on the 1,000 most relevant words according to the DFIDF value.

Models

PCA Components (4)

- UVA Box URL: https://virginia.box.com/s/wnahtjedjx07rh8cxfvmleo11c48xa1v
- UVA Box URL of the source TFIDF_L2 table: https://virginia.box.com/s/6hibiipjanzulfueryh665fr8iyd5isv
- GitHub URL for notebook used to create: https://github.com/rkim422/DS5001_Final_Project/blob/main/pca_models.ipynb
- Delimitter: ,
- Number of components: 10
- Library used to generate: sklearn
- Top 5 positive terms for first component: stabbed, rookie, playoff, stabbing, measles
- Top 5 negative terms for second component: guess, specifically, instance, enrollment, quote

PCA DCM (4)

The document-component matrix generated.

- UVA Box URL: https://virginia.box.com/s/ow0wmg1vp9nkgbjk5rizzpgaxc7xvjtu
- GitHub URL for notebook used to create: https://github.com/rkim422/DS5001_Final_Project/blob/main/pca_models.ipynb
- Delimitter: ,

PCA Loadings (4)

The component-term matrix generated.

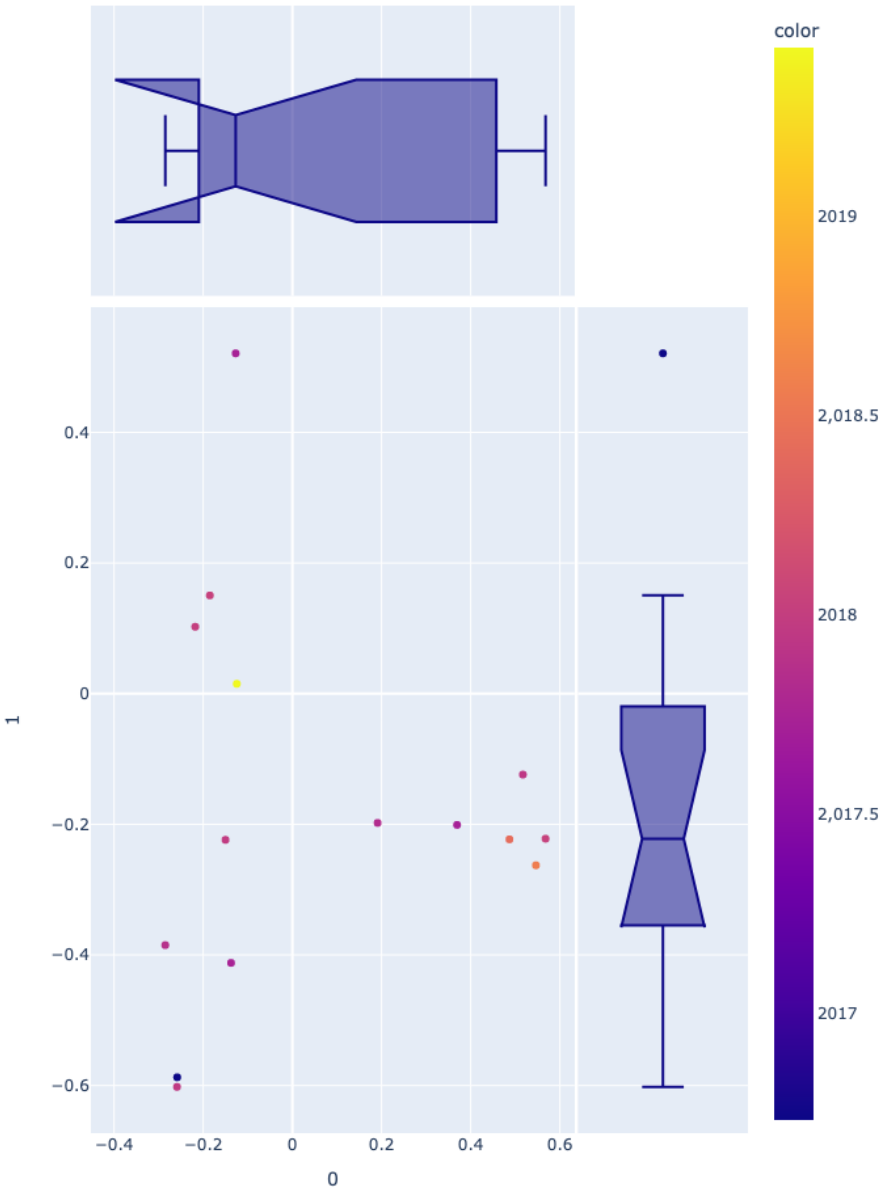
- UVA Box URL: https://virginia.box.com/s/rf3ykbh5j21ds22jmpgaah2biahk2rpz
- GitHub URL for notebook used to create: https://github.com/rkim422/DS5001_Final_Project/blob/main/pca_models.ipynb
- Delimitter: ,

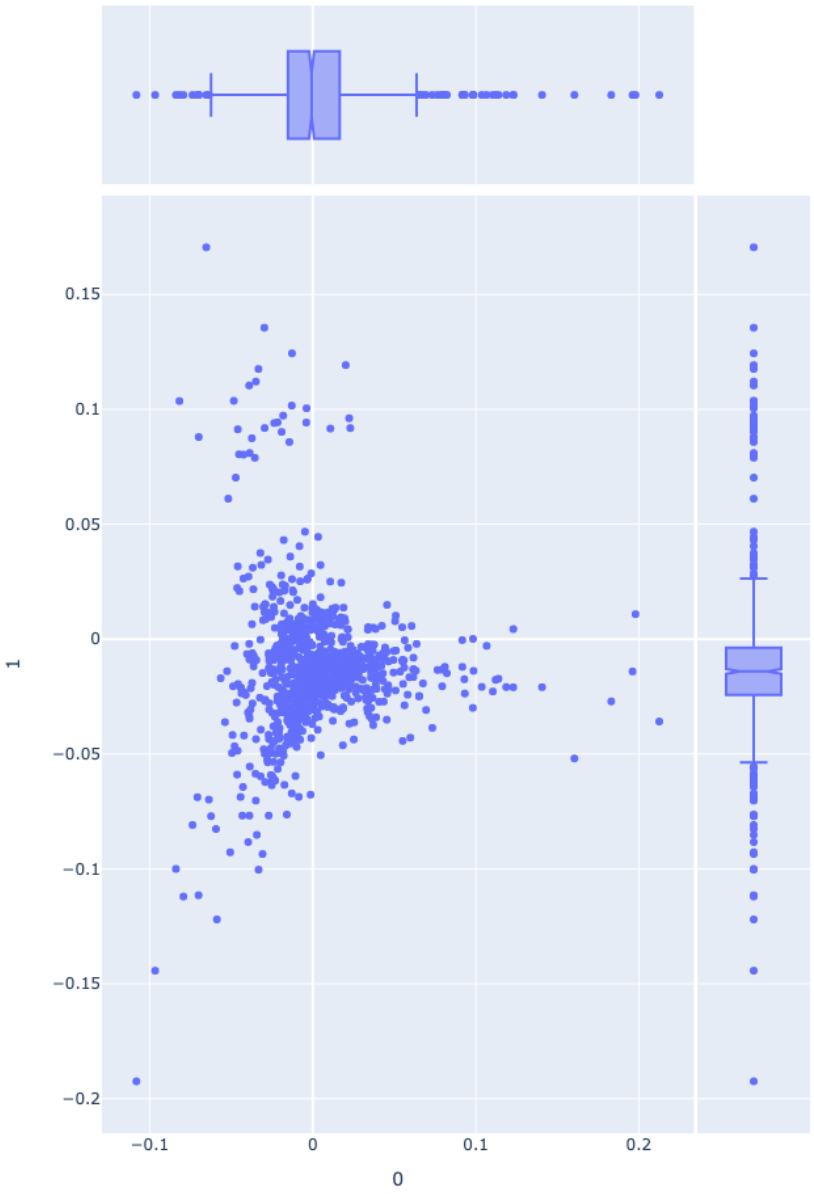
PCA Visualization 1 (4)

Include a scatterplot of documents in the space created by the first two components.

Color the points based on a metadata feature associated with the documents.

Also include a scatterplot of the loadings for the same two components. (This does not need a feature mapped onto color.)


color
2019
2,018.5
2018
2,017.5
2017
0.4
0.2
0
−0.2
−0.4
−0.6
1
−0.4
−0.2
0
0.2
0.4
0.6
0



Briefly describe the nature of the polarity you see in the first component:

The poles for the first component seemed to include words involving danger on one end including: stabbed, measles, volcano, and aligator, while the other end

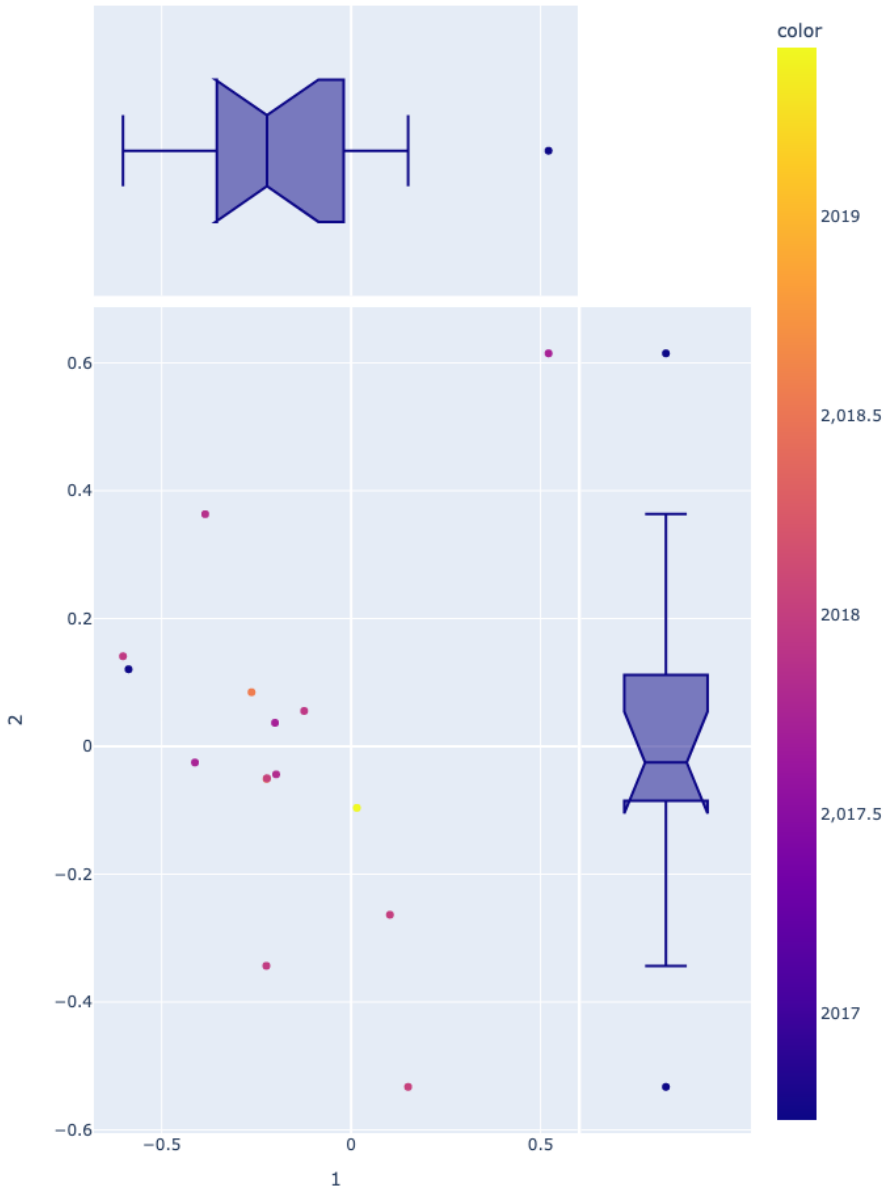
seemed to include words that add more details such as the adverbs: specifically, supposedly, and importantly.

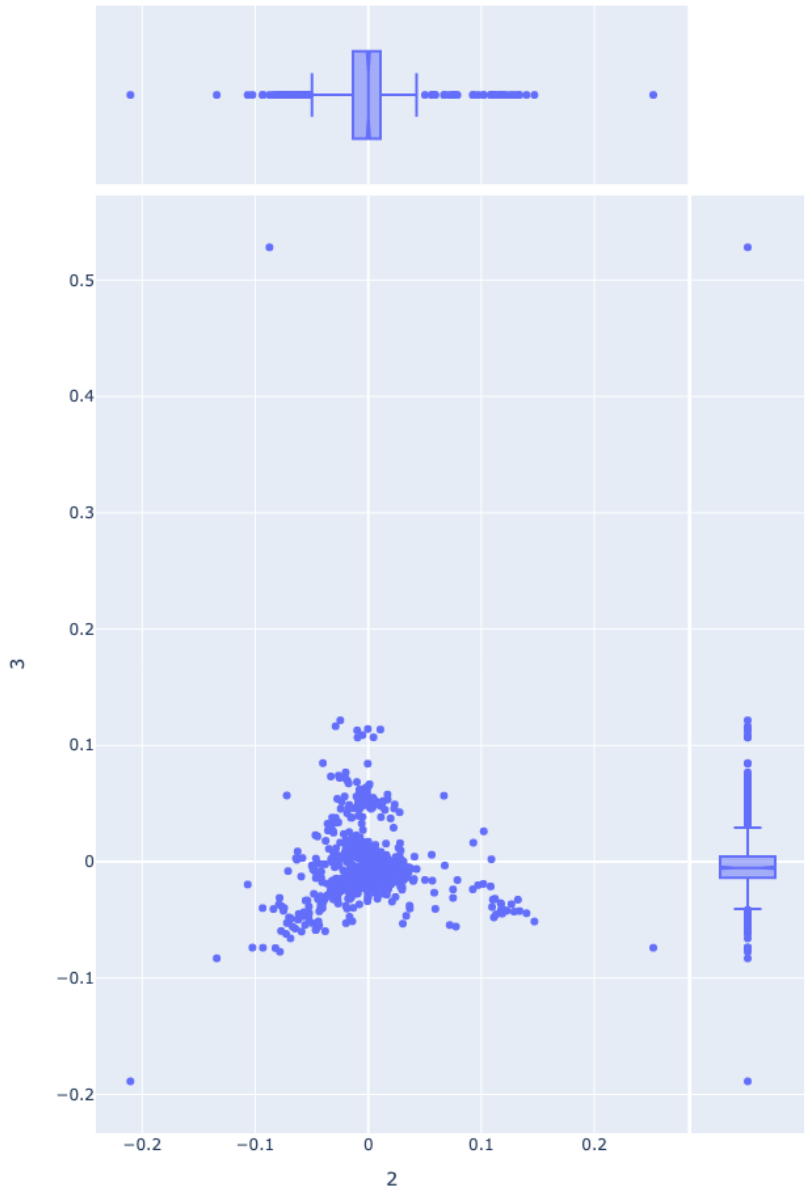
PCA Visualization 2 (4)

Include a scatterplot of documents in the space created by the second two components.

Color the points based on a metadata feature associated with the documents.

Also include a scatterplot of the loadings for the same two components. (This does not need a feature mapped onto color.)


color
2019
2,018.5
2018
2,017.5
2017
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
2
−0.5
0
0.5
1



Briefly describe the nature of the polarity you see in the second component:

The poles for the first component seemed to include words involving words that might describe something with negative connotation on one end includ-

ing: overshadowed, incomplete, and withdrew, while the other end seemed to include similar types of verbs including: circulated, objected, coordinated, and fractured.

LDA TOPIC (4)

- UVA Box URL: https://virginia.box.com/s/oeqursxuzjc4nfjrdre7y0q5key7r0q1
- UVA Box URL of count matrix used to create: https://virginia.box.com/s/htu2lnulet7n92ryddket3rhhjlnxinc
- GitHub URL for notebook used to create: https://github.com/rkim422/DS5001_Final_Project/blob/main/lda_models.ipynb
- Delimitter: ,
- Libary used to compute: sklearn
- A description of any filtering, e.g. POS (Nouns and Verbs only): Nouns only
- Number of components: 20 topics
- Any other parameters used: n_features: 4000, lda_max_iter: 5, lda_n_top_terms: 7
- Top 5 words and best-guess labels for topic five topics by mean document weight:
  - T00: deal, game, trial, points, coverage: games
  - T01: man, police, authorities, death, woman: crime
  - T02: government, border, state, budget, wall: politics of border security
  - T03: news, media, investigation, articles, security: journalism
  - T04: story, link, column, advertise, second: online content

LDA THETA (4)

- UVA Box URL: https://virginia.box.com/s/ws60hvcdlykaybbdv0lmcgdnccq5qlhv
- GitHub URL for notebook used to create: https://github.com/rkim422/DS5001_Final_Project/blob/main/lda_models.ipynb
- Delimitter: ,

LDA PHI (4)

- UVA Box URL: https://virginia.box.com/s/tzrsdbm8vmrhhwomk0ilpdqr1spmoesy
- GitHub URL for notebook used to create: https://github.com/rkim422/DS5001_Final_Project/blob/main/lda_models.ipynb
- Delimitter: ,

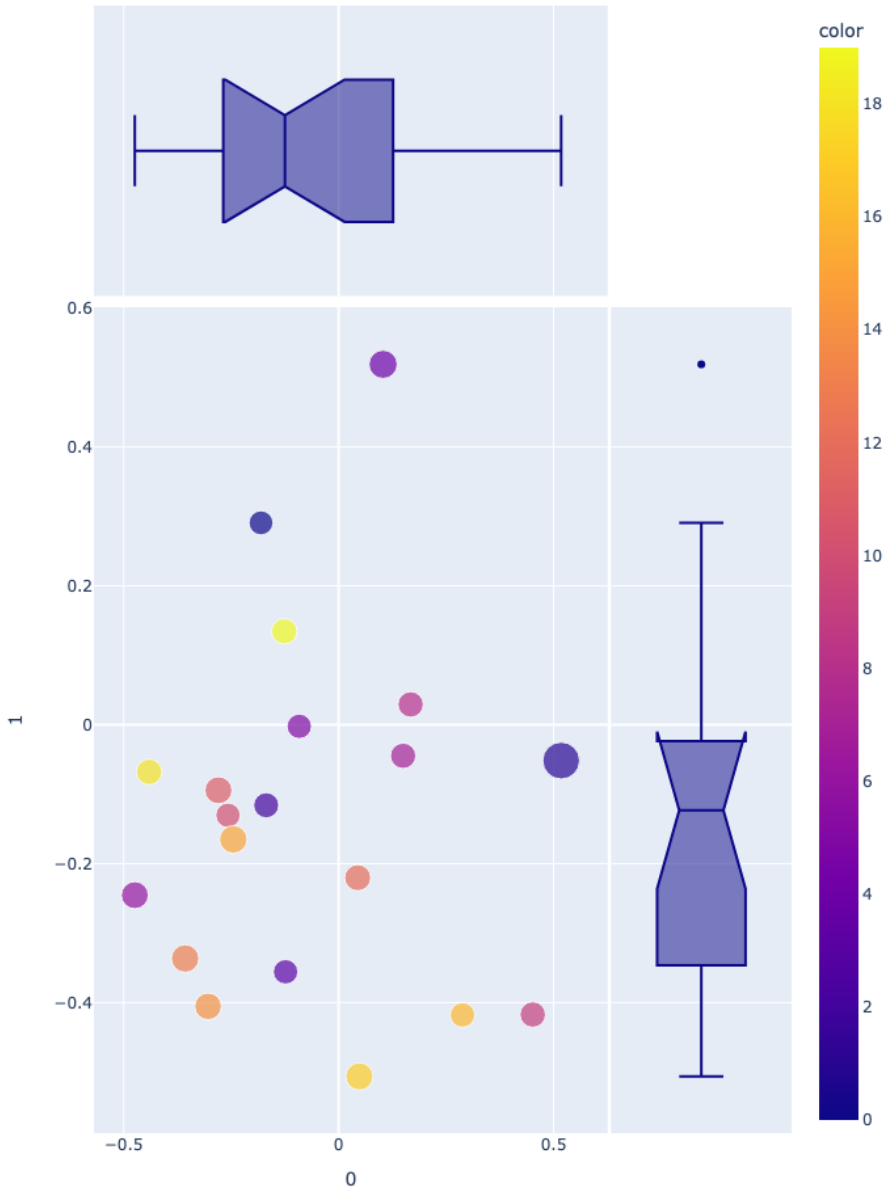
LDA + PCA Visualization (4)

Apply PCA to the PHI table and plot the topics in the space opened by the first two components.

Size the points based on the mean document weight of each topic (using the THETA table).

Color the points basd on a metadata feature from the LIB table.

Provide a brief interpretation of what you see.



Most of the topics seem to be similarly sized aside from topic 1 and 4, 1 being larger, and 4 being smaller. This indicates that topic 1 is likely more prevalent in the documents while topic 4 is less prevalent. This makes sense because topic 1 included lots of words involving people, which can be related to a wide

variety of documents. Topic 4 had more words that seemed less likely compared to the others to be as influential. Aside from the size of the points, no much seemed significant because the points were relatively clumpted together with few outliers. The points that were closer together suggest that these topics are more similar in word distribution.

Sentiment VOCAB_SENT (4)

Sentiment values associated with a subset of the VOCAB from a curated sentiment lexicon.

- UVA Box URL: https://virginia.box.com/s/mx017vcymetl7hbuthaq5hw2tnbti7wr
- UVA Box URL for source lexicon: https://virginia.box.com/s/5vp6u55g3lbe8hm9y3rghmt54qns1kv5
- GitHub URL for notebook used to create: https://github.com/rkim422/DS5001_Final_Project/blob/main/sentiment_models.ipynb
- Delimitter: ,

Sentiment BOW_SENT (4)

Sentiment values from VOCAB_SENT mapped onto BOW.

- UVA Box URL: https://virginia.box.com/s/h4lgzwkxqdfsdtbi3emwelld0bcgn5o5
- GitHub URL for notebook used to create: https://github.com/rkim422/DS5001_Final_Project/blob/main/sentiment_models.ipynb
- Delimitter: ,

Sentiment DOC_SENT (4)

Computed sentiment per bag computed from BOW_SENT.

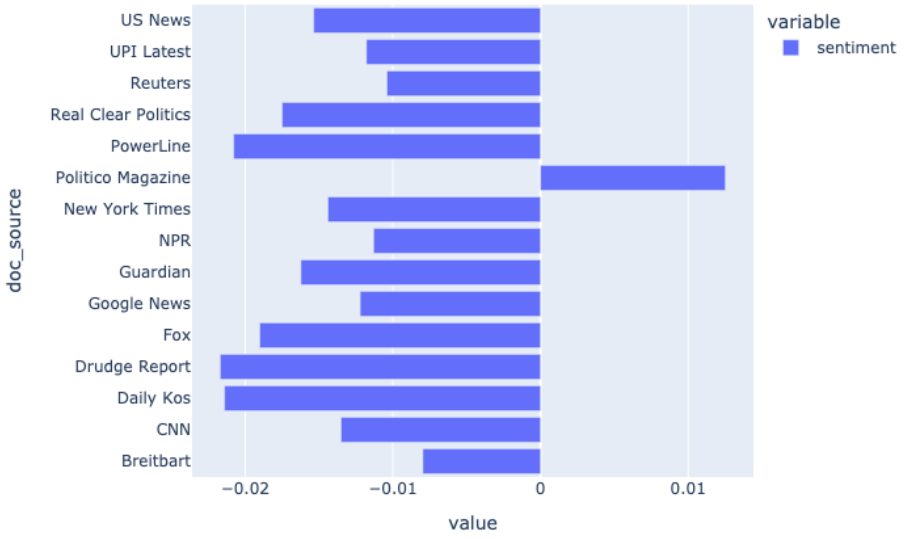
- UVA Box URL: https://virginia.box.com/s/xjets4b2jjq2ahrklg4rhz1frw5f3p5t
- GitHub URL for notebook used to create: https://github.com/rkim422/DS5001_Final_Project/blob/main/sentiment_models.ipynb
- Delimitter: ,
- Document bag expressed in terms of OHCO levels: [doc_source]

Sentiment Plot (4)

Plot sentiment over some metric space, such as time.

If you don't have a metric metadata features, plot sentiment over a feature of your choice.

You may use a bar chart or a line graph.



VOCAB_W2V (4)

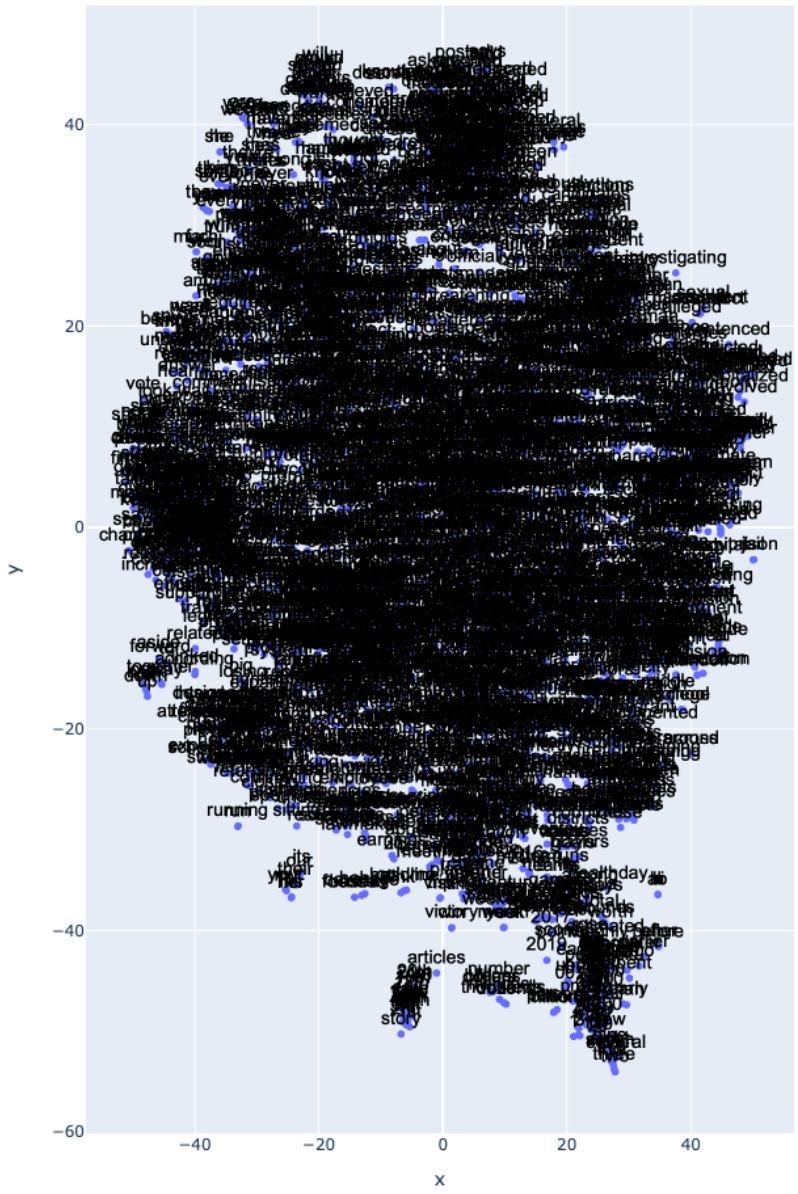
A table of word2vec features associated with terms in the VOCAB table.

- UVA Box URL: https://virginia.box.com/s/gnosz0x6lou6gw1l11yc9bir9di31if7
- GitHub URL for notebook used to create: https://github.com/rkim422/DS5001_Final_Project/blob/main/w2v_models.ipynb
- Delimitter: ,
- Document bag expressed in terms of OHCO levels: ['doc_source', 'doc_id']
- Number of features generated: 256
- The library used to generate the embeddings: gensim

Word2vec tSNE Plot (4)

Plot word embedding featues in two-dimensions using t-SNE.

Describe a cluster in the plot that captures your attention.



One cluster that captures my attention is at around 30, -20 with words such as: militants, civilians, victims, firefighters, soldiers, and officers. This cluster is likely for the various types of people that might be talked about in the news article. The types of people seem to trend towards those who are involved in

conflict or emergency response, which makes sense because those are typically the types of people who are written about in news reports.

Riffs

Provde at least three visualizations that combine the preceding model data in interesting ways.

These should provide insight into how features in the LIB table are related.

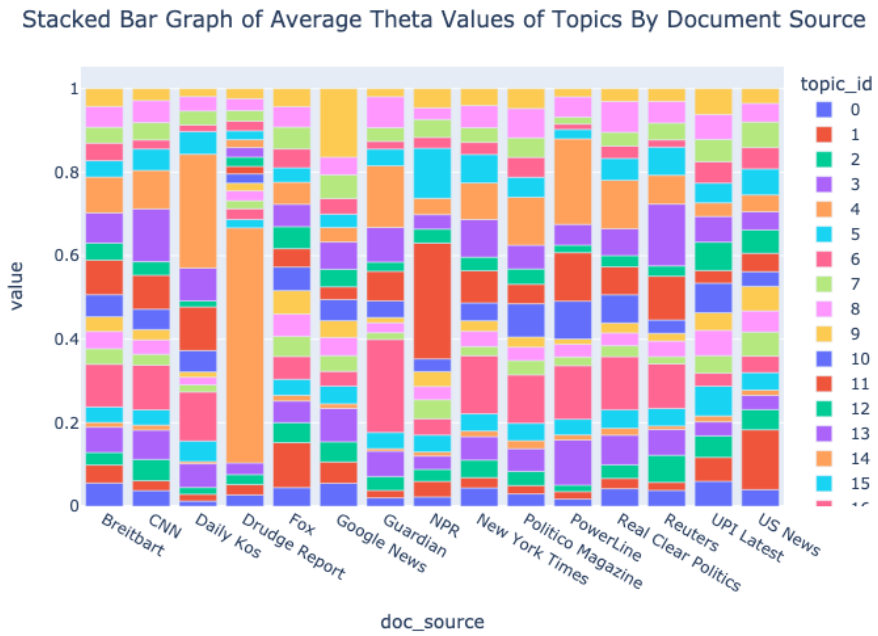
The nature of this relationship is left open to you -- it may be correlation, or mutual information, or something less well defined.

In doing so, consider the following visualization types:

- Hierarchical cluster diagrams
- Heatmaps
- Scatter plots
- KDE plots
- Dispersion plots
- t-SNE plots
- etc.

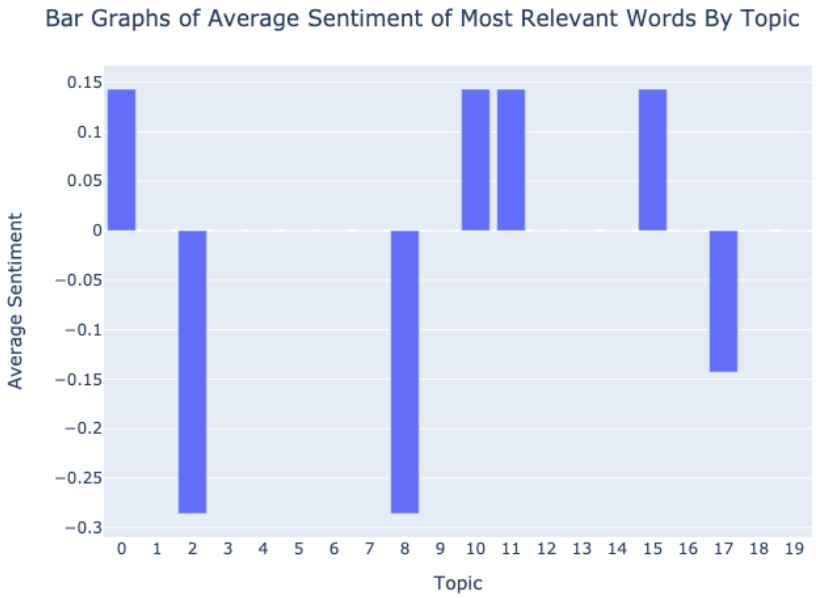
Riff 1 (5)

- GitHub URL for notebook used to create:



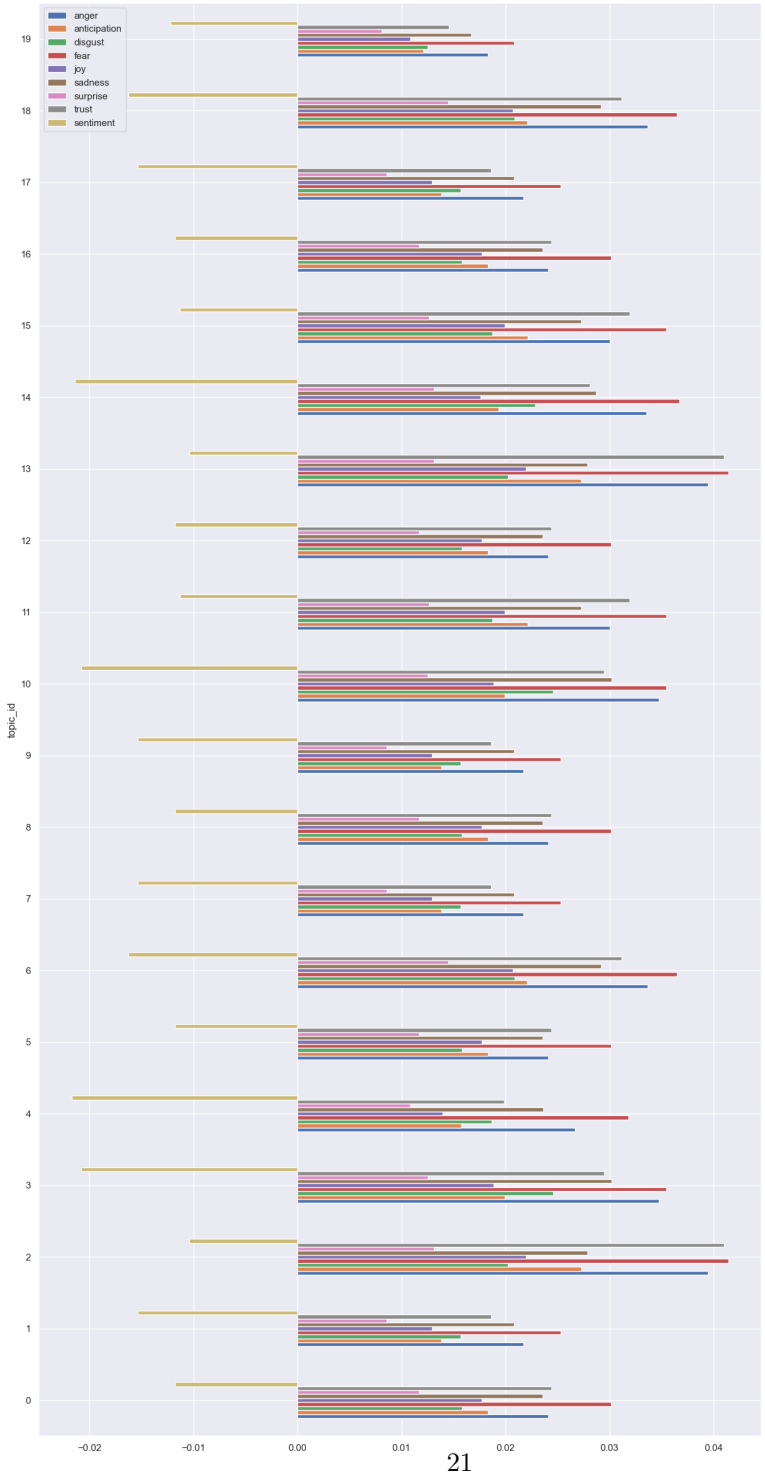
This graph displays stacked bar graphs of average theta values of topics by document source. With each source on the x axis, you can see how relevant each topic is to the documents from that source based on the size of the bar above. For example, for Drudge Report, topic 4 seems to be very prominant among documents from that source.

Riff 2 (5)



This graph displays bar graphs of average sentiment of most relevant words by topic. This graph was derived from the topic list, where each word was given its associated sentiment score. Based on the sentiment score of the 7 most relevant words to each topic, an average sentiment for each topic was determined. We can see that most of the topics had an average of 0 because overall, most words had a neutral sentiment score of 0.

Riff 3 (5)


anger
anticipation
disgust
fear
joy
sadness
surprise
trust
sentiment
topic_id
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
−0.02
−0.01
0.00
0.01
0.02
0.03
0.04

This graphs shows the specific array of emotion scores for each topic based on which document source it most closely related to. For this graph, the THETA, TOPICS, and BOW_SENT were required to compile the data.

Interpretation (4)

Describe something interesting about your corpus that you discovered during the process of completing this assignment.

At a minumum, use 250 words, but you may use more. You may also add images if you'd like.

Something I found interesting about my corpus was the distribution of emotion counts in the sentiment analysis modeling. Although the results are not especially surprising, I did find it interesting to see actual statistics to reinforce my preconceived notions. Specifically, I noticed that much more negative words were used than positive. This makes sense to me because bad news seems to catch more attention in the media. Additionally, the specific emotions of anger, fear, and sadness seemed to be especially prevalent in the documents. These emotions seem to be best at stirring up conversation in the media. While I have no statistics about the actual perception of the documents, I would like to continue using sentiment analysis on different news articles to learn more about how actual readers perceive various articles that deal with different emotions. If I were to find a news dataset that included some sort of human review score or something similar, I would like to compare my expected results with some actual calculated results. I also found it interesting that the most popular topic seemed to be about games. While I did not go much further into exploring the specificity of the topic labels, I would make a guess based on the top five words that the topic could be about sports. This also makes sense because sports journalism is a large market that reporters write about daily with new material happening daily. Although I found 1 of the top 5 topics to be somewhat political, I did find it surprising that politics were not more prevalent overall. From my personal experience, a large part of my news intake is related to politics; however, the results I found did not exactly meet my expectation of high political relevance.