Capstone Project 2: Topic Model Analysis of Berkshire Hathaway's Annual Letters to Shareholders

Tom Halloin

SPRINGBOARD DATA SCIENCE CAREER TRACK

Problem Statement

“Read 500 pages every day. That's how knowledge works. It builds up, like compound interest. All of you can do it, but I guarantee not many of you will do it.”

- Warren Buffett, when asked how to prepare for an investing career.

Most of the general public considers investment websites and annual reports as long, boring, and difficult to read. The average investment website is more complex than the novel Moby Dick, and few people understand the overly complex reports their investment managers send. Warren Buffett became one of the most successful investors of all time because of his ability to comprehend thousands of pages of data. Buffett is the exception to the rule. Most people don’t have the patience to read 500 pages a day, let alone comprehend it. Instead, I am going to use machine learning to summarize the equivalent of reading at least 500 pages, Berkshire Hathaway’s annual letters to shareholders since 1977. Using Natural Language Processing, I will try to summarize the letters and create a topic model to discover hidden themes that are present across the letters. This model will try to extract meaning from these letters by identifying recurring themes or topics and the change of these themes over time.

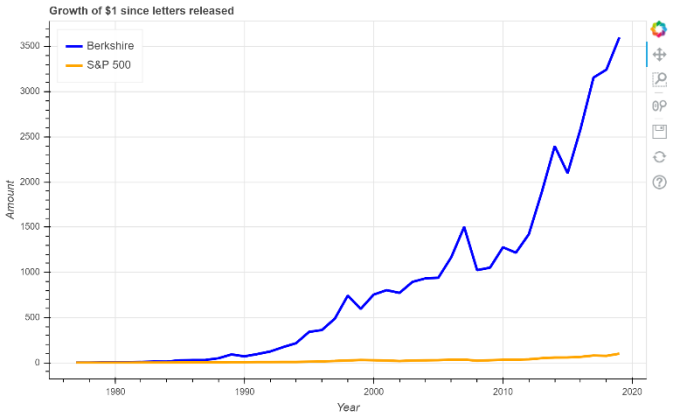
Acquiring, Cleaning, and Wrangling the Data

The data comes from Berkshire Hathaway's shareholder letters available at the link https://www.berkshirehathaway.com/letters/letters.html. The letters come in both HTML and PDF format, so part of the challenge came from scraping data from both formats into a feasible data set. For the HTML files (up to 1998), I used the packages urllib and Beautiful Soup to establish a secure internet connection, obtain the HTML file for each year, and save these files to a local connection. Eventually, Berkshire Hathaway thought that I was a bot and denied me to scrape multiple letters at once. The code still works, but it is something to watch out for if planning to run this code on your machine. The PDF files were trickier to parse. I used the libraries PyPDF2 and tika to download each PDF from the website. From there, I had to split all of the PDFs into individual pages to parse each page. For each page, I looped through to parse the text and added the text to a dictionary containing the text for each year.

The data required cleaning coming from the website and PDF files. The documents from the website still had remnants of HTML code and the PDFs were illegible. To make the letters legible, I used the Textacy package. Textacy is built on SpaCy and does pre- and post- processing for SpaCy, including cleaning text, generating text statistics, and creating n-grams. SpaCy is an advanced Natural Language Processing library used to break text into tokens, tag the tokens with their part of speech, and compare the similarity of two documents, among other things. Together, not only can these packages clean the texts so that they are the same as they were on the website, but also prepare the documents for topic modeling. Some of

the cleaning steps included getting rid of the aforementioned HTML, removing newlines and tabs, and normalizing Unicode.

The first page of the 2019 annual report includes a table showing investment performance of Berkshire Hathaway compared to the S&P 500. Using the read_pdf module from the Tabula package, I could create the following chart visualizing these data.



Comparing Different Summarization Techniques

There are two ways to summarize data: *extractive* summarization and *abstractive* summarization. *Extractive* summarization selects the most important phrases from the text and extracts it to get the most important phrases. *Abstractive* summarization creates sentences from the text using deep learning and returns these new sentences as the summary. There have been attempts with deep learning to create abstractive summaries, but extractive summaries have been proven more effective when summarizing documents. This is because challenges such as interpreting meaning of words, making inferences, and generating natural language are more difficult than a brute force approach found using extractive summarization. I will try three different extractive summarization methods: Lexrank, Textrank, and LSA. Lexrank and Textrank are graph-based algorithms based off of Google's PageRank. A good example of a graph-based algorithm is a social network, where each person is a node and the

strength of the friendship is the edge connecting the points. In document summarization, each sentence is a node, and each edge is calculated using a similarity score. Lexrank uses a method of cosine similarity weighted by term frequency – inverse document frequency, otherwise known as TF-IDF, to weight terms that have more meaning instead of just looking at document frequencies, while Textrank uses a log-based formula. The end result of applying each algorithm to a series of documents is a graph similar to the picture on page 3.

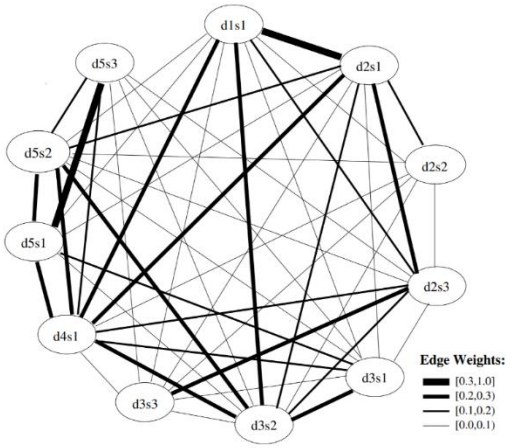
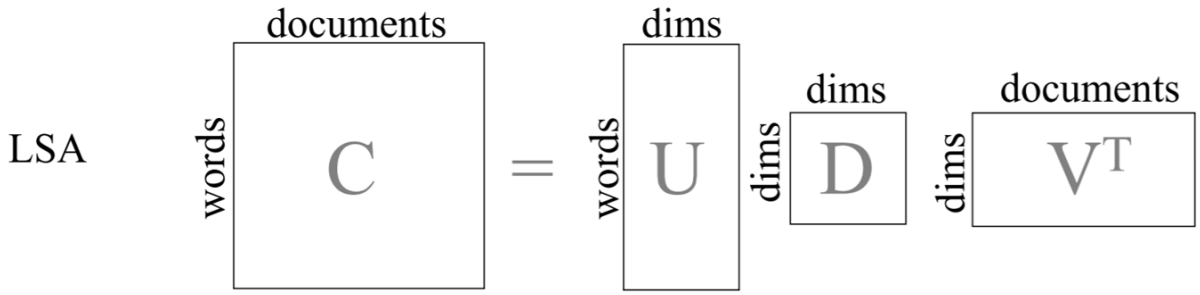


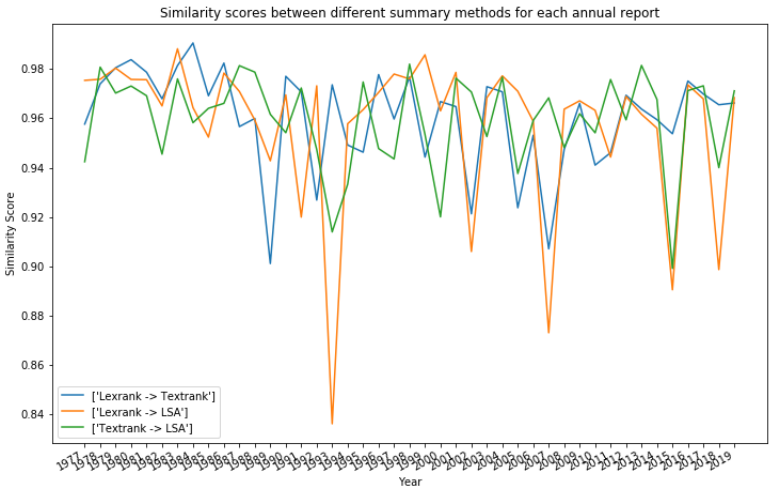
Figure 2: Weighted cosine similarity graph for the cluster in Figure 1.

See LexRank: Graph-based Lexical Centrality as Salience in Text Summarization by Gunes Erkan and Dragomir R. Radev for graph.

LSA, or *Latent Semantic Analysis*, analyzes similarities between the meanings of a series of documents and produces a series of concepts related to the documents. Like Lexrank and Textrank, LSA creates a term-document matrix consisting of word frequencies for each term in each document. Each word is a row while each document is a column. After creating this matrix, a linear algebra technique called singular value decomposition reduces the number of words, while keeping the same meaning in each of the documents. By reducing the number of words in each document, it is easier to



find similarities between different documents. Sentences that keep the most words from the reduction are considered the “most important”. These reduced words and phrases also generate topic models that will be shown later.



For each annual report, I used each of these three techniques to summarize each year in the corpus by extracting the five most meaningful sentences according to its method. Then, I used the SpaCy library to create a summary score to see how similar the summaries are to each other. SpaCy has a large vocabulary of words that can capture similar phrases even if the words do not match, so the techniques should produce similar summaries according to SpaCy. The graph to the right plots these similarity scores between the three different methods. Each line is a comparison between two of the methods for each year in the annual report. The divergences, as shown in the year 1993, provide a good contrast to see the different sentences each technique deems as important. Here is a sentence that Lexrank deemed “important” in 1993 that is actually a terrible summary of the document.

Berkshire's Share of Undistributed Berkshire's Approximate Operating Earnings Berkshire's Major Investees Ownership at Yearend (in millions) 1993 1992 1993 1992 Capital Cities/ABC, Inc. 13.0% 18.2% $ 83(2) $ 70 The Coca-Cola Company 7.2% 7.1%

94 82 Federal Home Loan Mortgage Corp. 6.8%(1) 8.2%(1) 41(2) 29(2) GEICO Corp. 48.4% 48.1% 76(3) 34(3) General Dynamics Corp. 13.9% 14.1% 25 11(2) The Gillette Company 10.9% 10.9% 44 38 Guinness PLC 1.9% 2.0% 8 7 The Washington Post Company 14.8% 14.6% 15 11 Wells Fargo & Company 12.2% 11.5% 53(2) 16(2) Berkshire's share of undistributed earnings of major investees $439 $298 Hypothetical tax on these undistributed investee earnings(4) (61) (42) Reported operating earnings of Berkshire 478 348 Total look-through earnings of Berkshire $856 $604 (1) Does not include shares allocable to the minority interest at Wesco (2) Calculated on average ownership for the year (3) Excludes realized capital gains, which have been both recurring and significant (4) The tax rate used is 14%, which is the rate Berkshire pays on the dividends it receives We have told you that we expect the undistributed, hypothetically-taxed earnings of our investees to produce at least equivalent gains in Berkshire's intrinsic value.

Both Lexrank and Textrank picked sentences that contain many key terms throughout the document, such as the company names, but I would not pick those sentences if I wanted to summarize the entire annual report. The sentences in the LSA summaries, in contrast, are much cleaner. Here are the five most relevant sentences of the most recent annual letter, using LSA:

1.) Nevertheless, when business ownership was sliced into small pieces stocks buyers in the pre-Smith years usually thought of their shares as a short-term gamble on market movements.

2.) Sometimes this job has been easy at other times, more than difficult, particularly when we began working with huge and ever- growing sums of money.

3.) Today, we have most of your money deployed in controlled businesses that achieve good-to-excellent returns on the net tangible assets each requires for its operations.

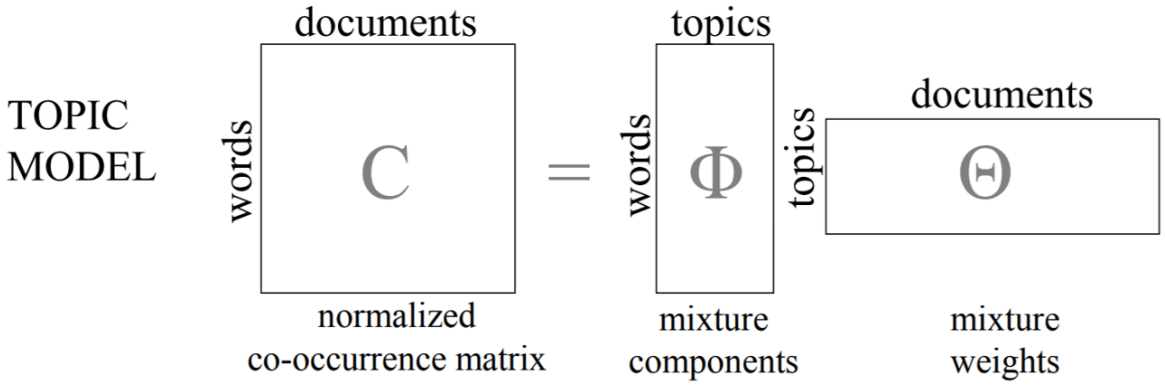
4.) We exclude our Kraft Heinz holding 325,442,152 shares because Berkshire is part of a control group and therefore must account for this investment on the equity method.

5.) First, Berkshires assets are deployed in an extraordinary variety of wholly or partly-owned businesses that, averaged out, earn attractive returns on the capital they use.

Selecting Appropriate Algorithms and Applications

Most machine learning focuses on supervised data – that is, data that has a particular label attached to it and building a model trying to predict the label on future data. Natural Language Processing has its own versions of supervised learning, mostly tied to sentiment analysis. A common example is predicting the “freshness” of a movie based on the text of the reviews. These annual reports do not have a similar label. Likewise, predicting returns based on the words in the report is possible, but it does not address the main problem: figuring out the context of the annual letters without actually having to read them.

To figure out this context, the model will examine the texts and output clusters of words that appear together. These clusters of words are potential topics. The best way to find these topics is to build a topic model. To be clear, the topic model is built off of the documents themselves, not the individual summaries. Each document consists of different topics, and each document is a weighted mixture of these topics. The order of the words in each of the documents does not matter. The end goal is to have a list of interpretable phrases for each topic.

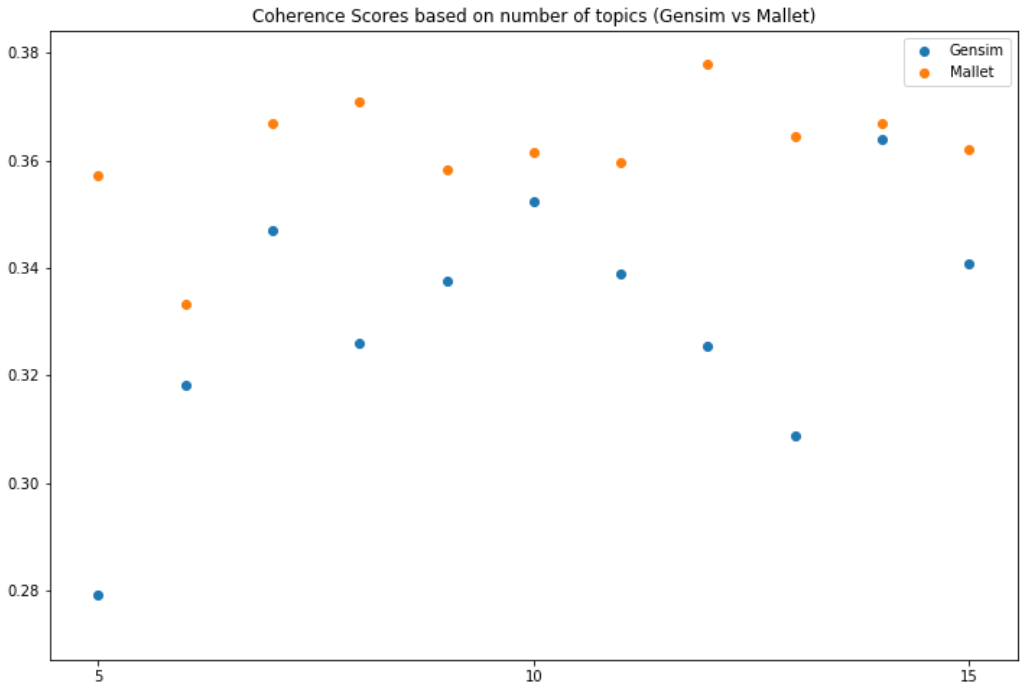


Machine Learning and Evaluation Techniques

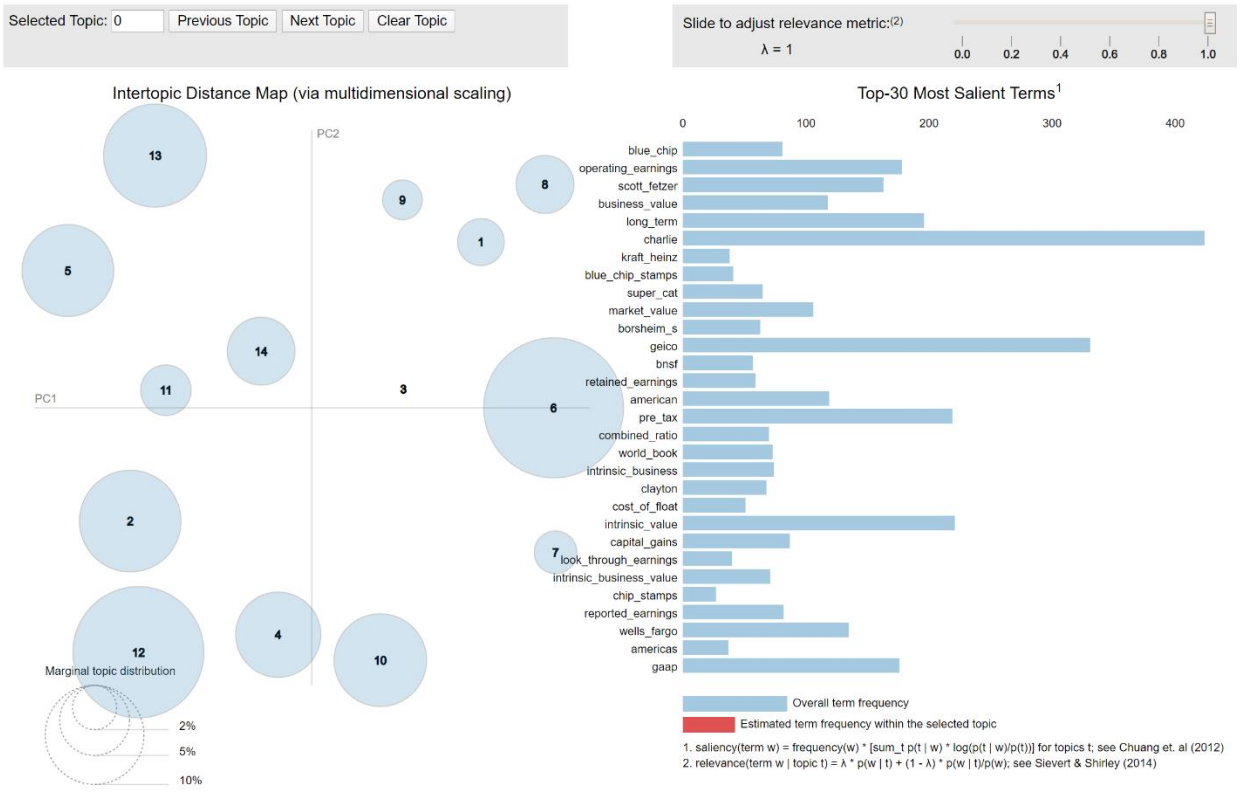
I will apply two different topic modeling algorithms: Gensim LDA and Mallet LDA. LDA stands for *Latent Dirichlet Allocation*, a technique similar to Latent Semantic Analysis covered earlier. Instead of trying to find the hidden meaning of the documents as done in LSA, we're trying to find the hidden distribution of words amongst the topics. The Dirichlet in Latent Dirichlet Allocation is a smoothing parameter. LDA is a computationally expensive algorithm. Gensim tries to get around this by providing a close approximation, while Mallet uses a Java library to speed up computation. The metric I'm going to use to measure the quality is a coherence score. Topic coherence measures the degree of similarity between high scoring words in the topic. A topic model with high coherence produces more interpretable topics. I will be using the c_v measure from Gensim, which scores cohesion from 0 (meaning the words in the topic do not go together at all) to 1 (all of the words are the same). The number of topics is a hyperparameter I will modify to find the model with the highest coherence score.

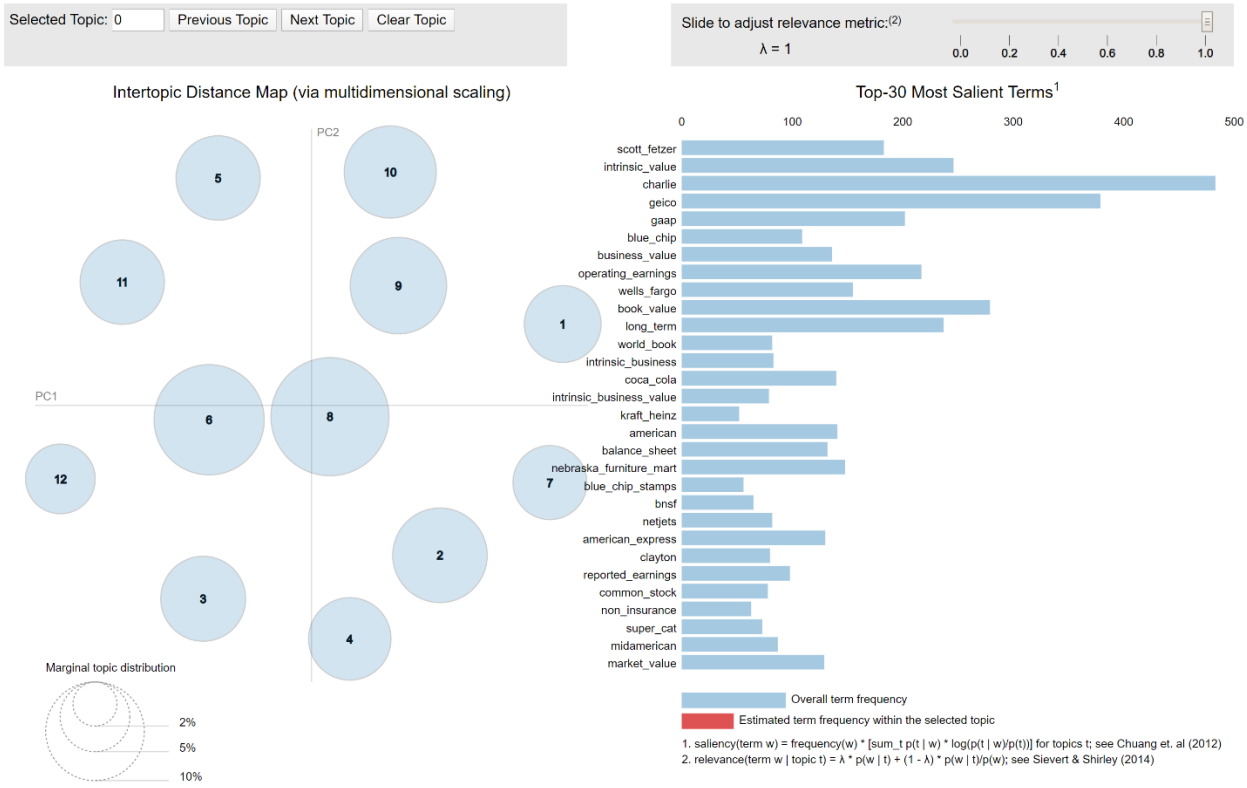
The first step to running the models is cleaning the documents a second time, this time to remove stop words, punctuation, numbers, and dates. Stop words such as "the" are so common that they do not belong to a particular topic. Punctuation does not fit in a group of words well, either. Keeping numbers might make sense in other applications, but they do not add much to a topic model. I removed dates and general terms such as "last year" because of generality. I also added bigrams and trigrams to each document. A bigram is a common two-word phrase, such as "Warren Buffett". Trigrams are three-word phrases, such as "Berkshire Hathaway, Inc.". These might catch phrases that single words alone would miss.

After the second cleaning of the documents, I ran the coherence scores for both Gensim LDA and Mallet LDA. The highest coherence score for Gensim LDA was .36, while Mallet was .37. That is not significantly different, but Mallet's topics were far superior. Below are graphs of each of the topic distributions. The axes are determined using Principal Component Analysis, a technique that reduces the dimensionality of a dataset at the expense of having easily interpretable axes. The size of the circle relates to the size of the topic. Topics that overlap each other are similar in nature. In practice, the more evenly distributed the topics, the better, to get a better cross-section of terms across the document.



Below is a sample of the distance maps, first from Gensim, then from Mallet. The relevance metric on the right shows words that are distinguishable for each topic.



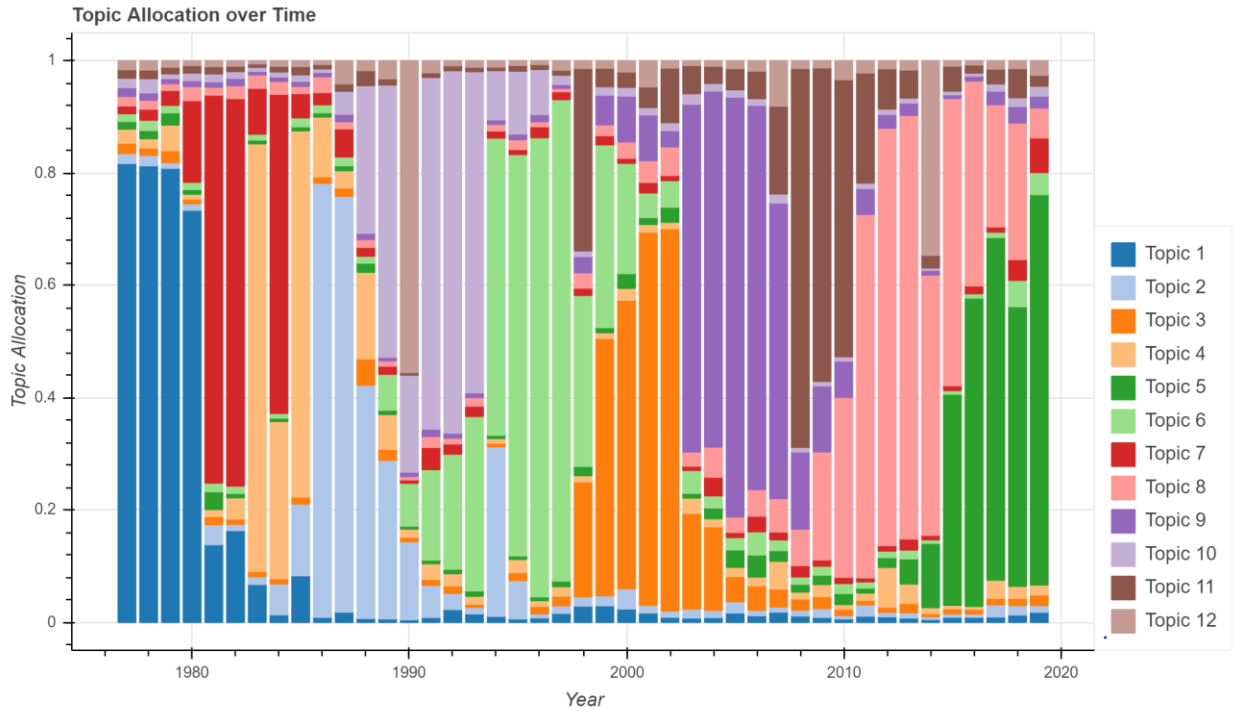


Research from the creators of the package suggest setting lambda at .6, but I found lower values gave better words for providing relevant and unique topics.

The words corresponding to each topic of the Mallet LDA model is shown on the next page. It is edited for interpretability purposes, such as removing underscores from bigrams. Below that is a chart showing the change in the distribution of topics over time. After some research, I would classify them as the following:

Topic 1: Early Holdings: Banks and Stamps
Topic 2: Scott Fetzer Company
Topic 3: Aviation Businesses
Topic 4: Bond Defaults and Newspapers
Topic 5: Massive Funds and Businesses
Topic 6: Insurance Underwriting
Topic 7: Insurance Companies
Topic 8: Industrial Holdings
Topic 9: Annual Meeting at the Qwest
Topic 10: Cowboy Boots, Junk Bonds and Mortgages
Topic 11: Electricity and Reinsurance
Topic 12: US Steel

Topic 1	Topic 2	Topic 3	Topic 4
Wesco Financial	Ralph Schey	Fractional Ownership	WPPSS
Illinois National	Scott Fetzer's	General Re	News
Earnings Per Share	Ralph	September 11th	Courier Express
Phil Liesche	Fechheimer	Executive Jet	The News
National Bank	Chuck	Owners Manual	Stan
Topic 5	Topic 6	Topic 7	Topic 8
Kraft Heinz	Major Investees	Non-controlled	BNSF
Hathaway Energy	Cat Business	Good Businesses	Marmon
Fund of Funds	Super Cat Business	Controlled Businesses	Operating Expenses
Hedge Funds	Geico's	Buffalo Evening	ton miles
Berkshire Hathaway Energy	Earnings Reported	Unusual Sales	Heinz
Topic 9	Topic 10	Topic 11	Topic 12
R.C. Willey	HH Brown	General Res	Stock Prices
New Jersey	Cost of Funds	Compounded annually	Contingency Reserve
Growth Rate	Preferred Stock	Black Scholes	Berkshire System
Electric Customers	H Brown	Kern River	Board of Directors
Qwest	RJR Nabisco	General Electric	Years Later



Conclusion and Areas of Further Research

This is an end-to-end project that attempted to describe the contents of Berkshire Hathaway's annual letters to shareholders without taking the burden to read them. Attempts included summarizing the letters year-by-year using three different summarization techniques and building two different topic models to represent the different topics the letters cover. These models revealed a list of relevant words corresponding to each topic.

Future edits might include incorporating the following ideas:

- Applying extractive summarization techniques to the dataset.
- Tracking sentiment analysis from letter to letter.
- Applying techniques used in this paper to other financial documents, such as such as 10-Ks and earnings conference call transcripts.