Categorizing Companies by Stated Risk Factors in 10-K Filings

Evie Wan, Nick Mosca, Eric South

CS506 Final Report

Introduction

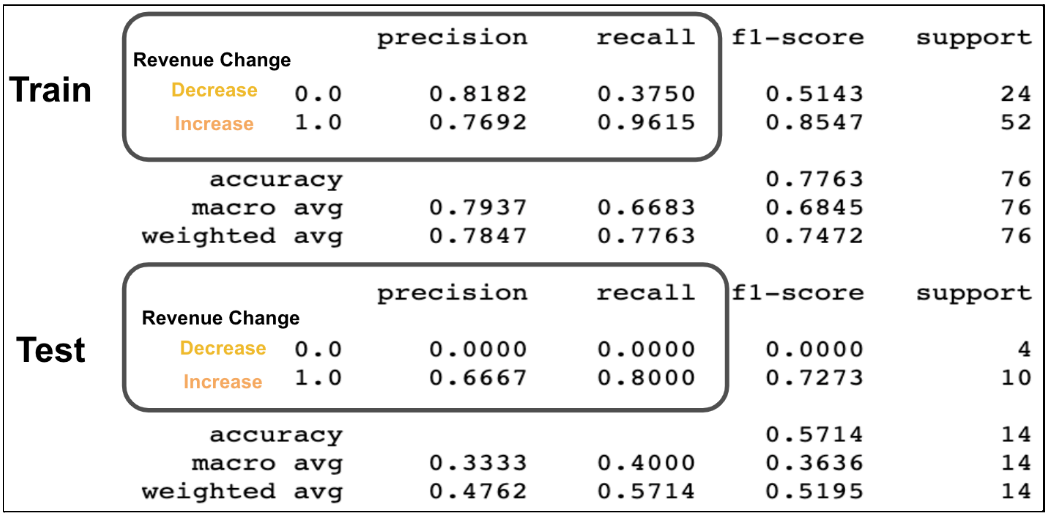
We developed a web-scraping tool that extracts financial documents from EDGAR, a database hosted by the Securities and Exchange Commission (SEC). Publicly traded companies in the United States must annually submit 10-K filings to the SEC. These 10-K documents are comprehensive reports for current and potential investors, where financial performance, forward looking statements, and risk factors are detailed across multiple sections. Each 10-K document includes a risk section, often labeled as “Item 1A Risk Section,” which details distinct vulnerabilities a company is working to mitigate. 10-K risk sections often consist of multiple paragraphs, and may emphasize technical, market, or supply chain risk, among other factors.

We hypothesized that vulnerabilities described in Item 1A could be used to categorize companies into distinct groups, where underlying sentiments across paragraphs of text would be dependent on a company’s general risk profile. We assumed that risk profiles were not made equal, and that different company vulnerabilities were sensitive to different external factors. Given these assumptions, we were curious whether the economic disruption that followed COVID-19 had a disproportionate impact on biotechnology companies with certain risk profiles. Although biotechnology companies in general have performed well during the pandemic, we asked whether a company’s financial performance (e.g., fold change in revenue between 2019 and 2020) could be linked to their self-stated risk factors (found in their 2019 10-K filings). We gathered 10-K filings for companies within the iShares Nasdaq Biotechnology ETF (IBB), an index fund consisting of over 200 biotechnology companies.

Our project explored whether natural language processing techniques, such as topic modelling, could be used to differentiate companies by their stated risk factors. We developed a corpus of 10-K risk sections for hundreds of biotech companies by both adapting an API that interacts with the EDGAR database and developing an HTML web scraper that identified, cleaned, and aggregated paragraphs from Item 1A subsections. Aggregated risk texts were then subject to Latent Dirichlet Allocation (LDA), an unsupervised learning, probabilistic algorithm which represented a corpus of text as an underlying set of topics (**Figure 1**). Our LDA model recognized at least two distinct topics: navigating the FDA regulatory landscape and monetary themes related to company evaluation and cash flow.

To better understand the sentiment context of 10-K documents, we referenced the Loughran-McDonald Sentiment Word Lists, a dictionary published in the Journal of Finance (Loughran, 2011), which associated common words in 10-K filings with sentiment categories (negative, positive, uncertainty, litigious, strong modal, weak modal, constraining). A growing literature finds significant correlations between sentiment categories and stock price reactions (Lougran, 2011). Using the sentiment vocabulary identified from historic 10-K filings, we engineered a range of numeric features based on word counts to quantify the sentiment of financial reports. In other words, 10-K documents were viewed as a set of words distributed across different sentiment categories. We then used logistic regression to model sentiment features against change in revenue. This will determine whether document sentiment had any predictive power on future earnings. For each company, independent variables were a collection of ‘word counts’ that described the prevalence of intra-document sentiment categories. These ‘sentiment profiles’ were then juxtaposed against whether a company had a positive or negative fold change in revenue between 2019-2020. Logistic regression was chosen as the modeling method for the following reasons. First of all, different from linear regression, logistic regression can be used to examine the association of independent variables and a dichotomous dependent variable, in this case, 0 or 1 with 0 representing decrease in revenue and 1 representing increase in revenue. Second of all, the logistic regression model can be extended to include several explanatory variables, allowing the multiple sentiment categories being modeled against revenue change. Model result shows that sentiment vocabulary from 10-K files is able to predict whether a company had a positive fold change in revenue with 0.769 train accuracy and negative fold change in revenue with 0.818 train accuracy.

Table 1. Logistic model performance. Top bracket represents training precision and bottom bracket represents test precision.



```
                                precision    recall  f1-score   support
       Revenue Change
Train    Decrease   0.0            0.8182    0.3750    0.5143        24
         Increase   1.0            0.7692    0.9615    0.8547        52

                accuracy                               0.7763        76
               macro avg           0.7937    0.6683    0.6845        76
            weighted avg           0.7847    0.7763    0.7472        76

                                precision    recall  f1-score   support
       Revenue Change
Test     Decrease   0.0            0.0000    0.0000    0.0000         4
         Increase   1.0            0.6667    0.8000    0.7273        10

                accuracy                               0.5714        14
               macro avg           0.3333    0.4000    0.3636        14
            weighted avg           0.4762    0.5714    0.5195        14
```

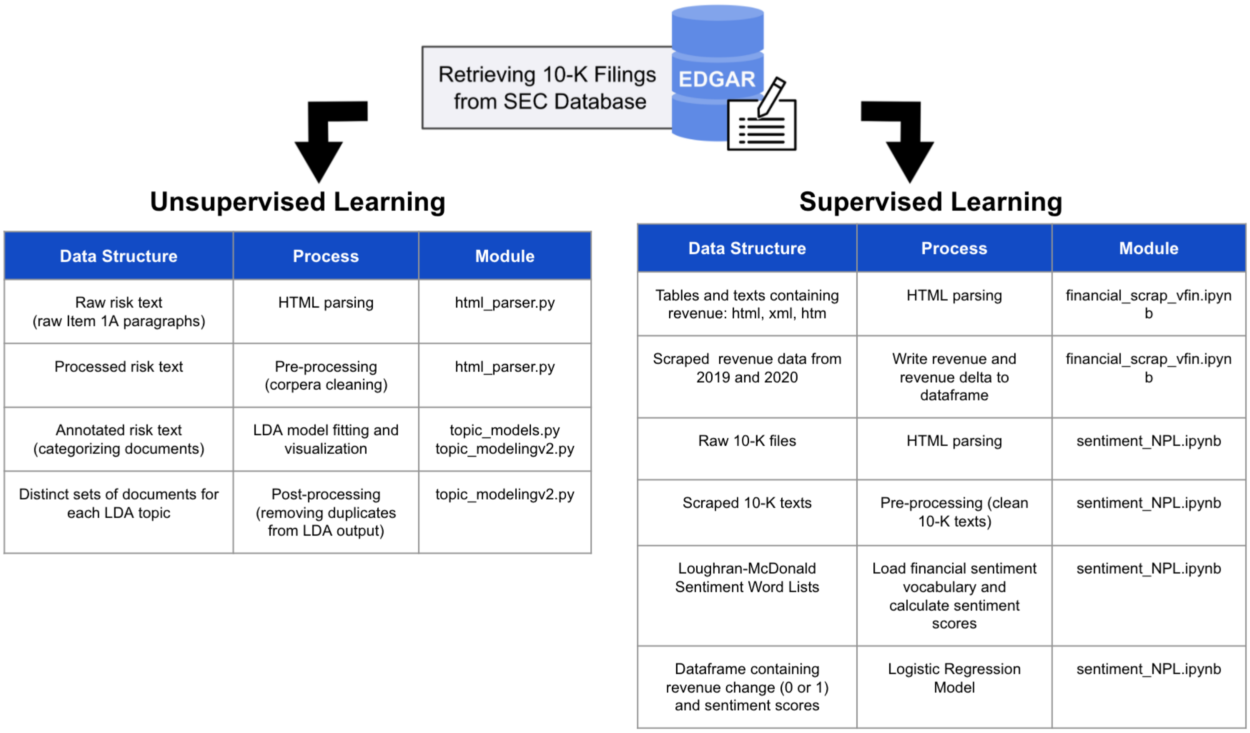
<u>Data Collection, Preparation, and Cleaning</u>

To conduct sentiment analysis on hundreds of financial documents, we first developed a function to bulk download 10-K filings directly from the EDGAR database (hosted by Securities and Exchange Commission). Our bulk_extraction function was designed to return 10-K documents when given a list of ticker symbols (i.e., shorthand notations for specific companies) along with a specific year of interest. Using this function, we downloaded 281 10-K filings from 2019-2020, which detailed financial performance among companies in the IBB index. 10-K filings were downloaded onto our local machines as HTML files, and subsequent parsing modules were designed to navigate our directory systems, locate 10-K filings of interest, and scrape relevant subsections.

To expedite the parsing of 281 HTML documents that were spread across different local directories, we used Pathlib’s .glob method to generate absolute file paths when provided a list of companies and years. We implemented a function called file_paths within our path_mover.py module, which utilized both regular expressions and python's built-in pathlib package to generate directory paths based on file types. File type extensions were coupled with ticker symbols embedded inside the file, which allowed us to generate file paths for every downloaded 10-K document. We then connected our web scraping module (10K_extraction.py) to our HTML parsing module (html_parser.py), which enabled the scanning and extraction of hundreds of 'risk sections' (i.e. paragraphs of strings found between Item 1A and Item 1B from SEC 10-K filings. Lists of file paths then served as inputs for our preprocessing grab_section_text function, which used the BeautifulSoup package to navigate HTML trees and isolate paragraphs of interest.

The format of a 10-K document is similar at first glance but differs in terms of HTML structure. These discrepancies in HTML tree structuring were non-trivial, as bespoke or non-generalizable scraping algorithms would fail to robustly identify Item 1A subsections among hundreds of 10-K filings. Despite these technical challenges, our html_parser.py module was able to recognize the majority of Item 1A subsections among hundreds of company filings. The isolated raw 10-K risk sections were then processed in our clean_strings function, where persisting html tags, whitespaces, and end of line characters were removed. Furthermore, risk texts were tokenized, lemmatized, and stripped of common stopwords using the nltk package.

Our HTML parsing module could produce a CSV which contained both company ID and its associated (cleaned risk) text. Similarly, a different HTML parser was used to scrape company revenue of 2019 and 2020 from 10-K documents. First, regular expression was used to isolate companies' central key and accession number. These elements were then concatenated to mass produce the html link, from which the .htm link containing table with revenue information will be extracted. This method was developed as a workaround after several attempts with different methods such as extracting Json and .xml contents and store values in dictionaries. The workaround was able to be generalized to most companies (177 out of 281 companies). We then differentiated hundreds of biotechnology 10-K filings by implementing both supervised and unsupervised learning models, where underlying sentiments among textual documents served to either categorize or engineer new features for companies.



Unsupervised Learning

Data Structure	Process	Module
Raw risk text (raw Item 1A paragraphs)	HTML parsing	html_parser.py
Processed risk text	Pre-processing (corpera cleaning)	html_parser.py
Annotated risk text (categorizing documents)	LDA model fitting and visualization	topic_models.py topic_modelingv2.py
Distinct sets of documents for each LDA topic	Post-processing (removing duplicates from LDA output)	topic_modelingv2.py

Supervised Learning

Data Structure	Process	Module
Tables and texts containing revenue: html, xml, htm	HTML parsing	financial_scrap_vfin.ipynb
Scraped revenue data from 2019 and 2020	Write revenue and revenue delta to dataframe	financial_scrap_vfin.ipynb
Raw 10-K files	HTML parsing	sentiment_NPL.ipynb
Scraped 10-K texts	Pre-processing (clean 10-K texts)	sentiment_NPL.ipynb
Loughran-McDonald Sentiment Word Lists	Load financial sentiment vocabulary and calculate sentiment scores	sentiment_NPL.ipynb
Dataframe containing revenue change (0 or 1) and sentiment scores	Logistic Regression Model	sentiment_NPL.ipynb

Figure 1. Overview of our analytical pipeline which aimed to explore whether COVID-19 had a disproportionate impact on biotechnology companies with certain risk profiles. 10-K filings for hundreds of biotechnology companies were collected and fed into both supervised and unsupervised models.

Analysis

Unsupervised Modeling: Latent Dirichlet Allocation

Once Item 1A documentation was isolated for all companies, we categorized risk section text by applying Latent Dirichlet Allocation (LDA). LDA is a generative, hierarchical probabilistic model for discrete data, where words, documents, and corpera serve as model parameters (Blei, 2003). LDA adds intra-document statistical structure, where documents are represented as a mixture of gaussian distributions, or topics, which are each defined as a subset of words in the corpus (Blei, 2003). Each Item 1A document was assigned a latent, dominant topic, which then served as a feature to help categorize companies within our dataset. We developed our LDA model in topic_model.py, where lists of companies (organized as a CSV file; obtained from our html_parser.py module) were broken down into a corpus dictionary (i.e. unique set of words) and corpus of text (i.e. term-frequencies for each risk document). Initially, we specified 5 *a priori* topics within the LDA, based on the literature surrounding business risk theory. We surmised that business risks can be categorized as either: market, liquidity, credit, or operational. Our LDA model assigned a mixture of probabilities for each company document, where each probability represented the prevalence of (1 of 5) topics in the risk text. Companies were then labelled by which LDA topic was most dominant among Item 1A documentation, which enabled us to categorize companies into discrete groups. Topic groups were then visualized as word clouds, which included the top 15 words associated with each topic (**Figure 2**).

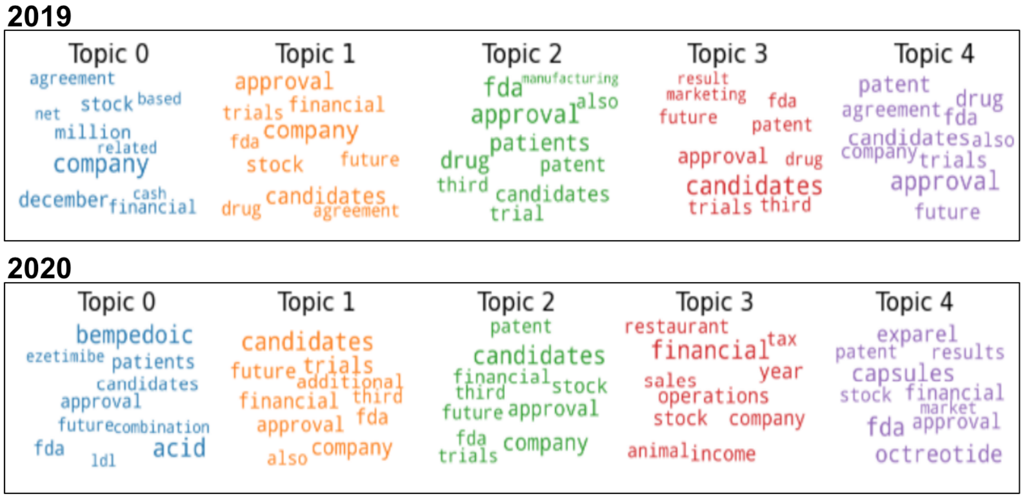


Figure 2. Word clouds for each of the 5 topics generated by latent dirichlet allocation (LDA). Topics observed considerable overlap, with themes related to FDA regulatory approval appearing in all groups.

We then analyzed whether financial performance pre- and post- COVID-19 was dependent on a company's general risk profile. Since each document in our corpera was characterized as a mixture of topics, we labeled companies by the dominant topic in their 10-K filings. These 'risk labels' were compared with each company's fold change in revenue between 2019-2020. Fold change in revenue was calculated as follows:

$$(2020\,revenue\ -\ 2019\,revenue)\ \div\ 2019\,revenue$$

We looked for trends among topic groups by plotting the fold change in revenue for each company, and then checked if any topic group was disproportionately present among extreme positive or negative growth (**Figure 3**).

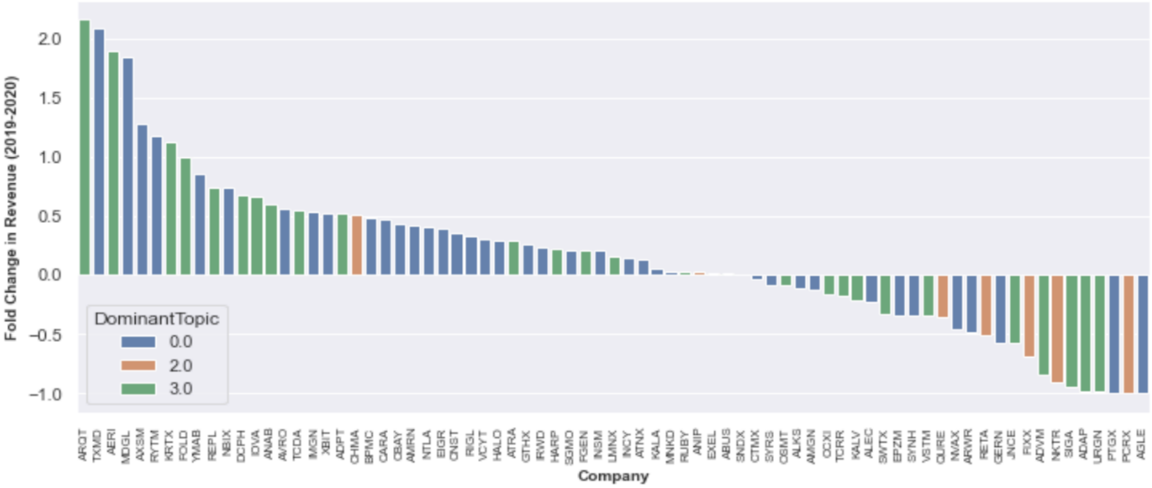


Figure 3. Fold change in revenue among 75 biotechnology companies. Plot colors and dominant topics (i.e., legend labels 0, 2, and 3) correspond to assigned topic groups from the preliminary LDA model (see **Figure 2**). Results were inconclusive due to an imbalanced allocation of topic groups (see main text).

Although our LDA model could assign topic mixtures to documents, our initial groups failed to represent distinct subject matters. Words related to company evaluation, FDA approval, initial public offerings, and miscellaneous drugs were scattered across topics (**Figure 2**). Furthermore, despite specifying 5 topics *a priori* during LDA, only 3 topic groups were ever dominant in a document (i.e., the LDA always recognized 2 minor, non-predominant topics among risk text) (**Figure 3**). We figured this imbalanced intra-document statistical structure was due to an incorrect specification of topic groups *a priori*, where our risk texts simply didn't have 5 underlying topics to differentiate. Moreover, certain words such as 'approval' or 'financial' were labelled as topic-defining words for multiple groups (**Figure 2**). Given these drawbacks, we decided to revamp our LDA model with improved pre-processing, hyperparameter tuning, and post-processing.

Revisiting our LDA Model

Our initial topic groups observed considerable overlap, which suggested that a need to either tune our model or refine preprocessing steps. To achieve more distinct topics we refined our preprocessing steps by adding additional stopwords that added little to no sentiment to each topic. We added additional 'financial based' stop words (e.g., 'company', 'december') that were not included in the NLTK package.

In terms of LDA hyperparameter tuning, we adjusted the number of iterations through our corpus when generating topic distribution. Increased iterations through the corpus allows better topic inference, resulting in more defined topics. During model training we also increased the number corpus iterations from 10 to 50. The threshold for minimum probability was also increased to further increase differentiation between topics. The results of **Figure 4** indicated that only 3 of our 5 topics contributed to the topic distribution. In our updated model we specified only 3 topics.

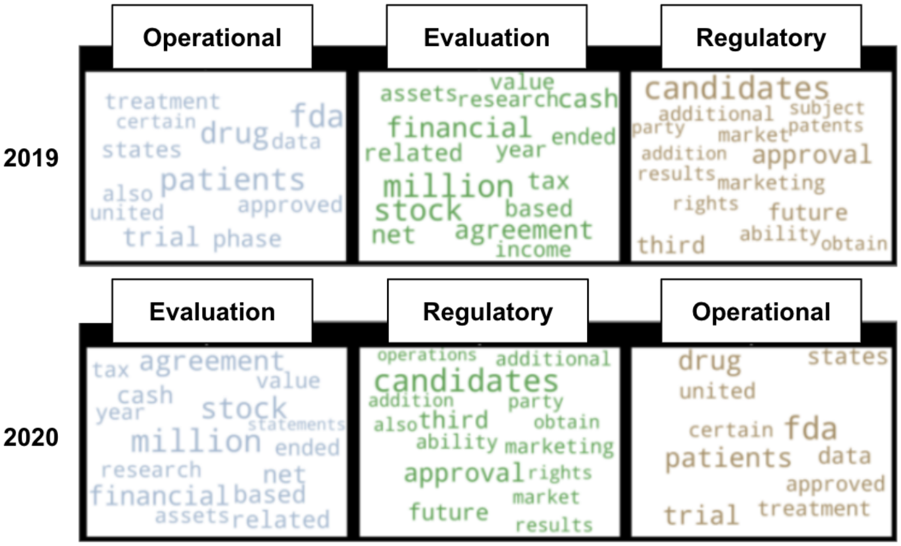


Figure 4. Word clouds for each of the 3 topics generated by latent dirichlet allocation (LDA). Topics observed loose genres of FDA regulation, operational logistics, and market evaluation. LDA was coupled with our post processing function clean_model_topics. Additional preprocessing was also incorporated prior to model construction.

We revised our LDA model with additional pre- and post-processing within topic_modeling_v2.py. Part of our LDA output was a nested list of ‘topic groups,’ which consisted of both topic-specific words and associated weight. For each word, numeric weights represented the degree of association to a particular topic (i.e, the higher the weight, the stronger association to topic *X*). Our initial LDA model had little post-processing, and thus did not utilize this word-to-weight output (which contributed to particular words being represented among multiple word clouds). However, in our refined model, we wrote the clean_topics_modeling function, which removed duplicate words that appear in multiple topics. To retain topic-defining words, the weights of duplicate entries were compared. A word was not dropped from a topic group if said word had the highest weight compared to its duplicates. We then generated 3 word clouds that each represented distinct subject matters (**Figure 4**). Topics observed loose themes of FDA regulation, operational logistics, and market evaluation.

Once we generated word clouds that could be distinguished by eye, we revisited our 10-K corpora (i.e., risk text from 170 companies) and measured the prevalence of topic-specific words in each document. Elements from each word cloud were counted in each document. These word counts were then normalized into percentages, which resulted in a data frame where each row was a document, each column was a topic, and values were the proportions of topic-specific words. These proportions were then plotted as population distributions to see how topics were represented across all documents between 2019 and 2020 (**Figure 5**).

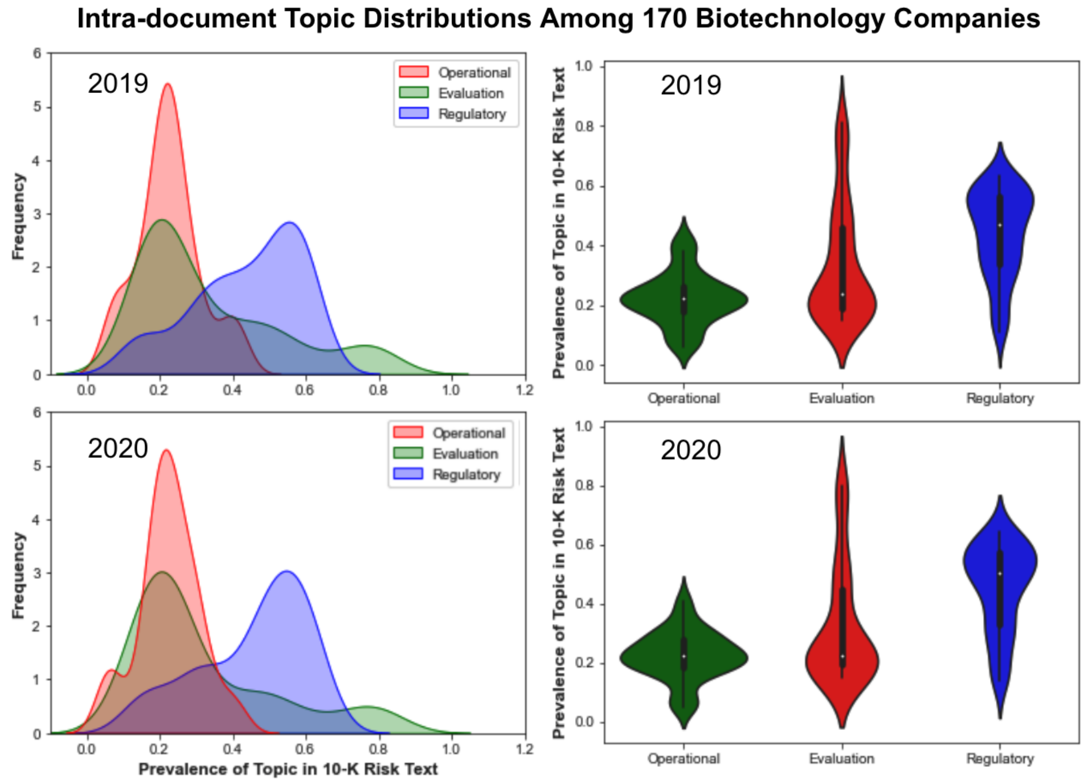


Figure 5. Intra-document topic distributions among risk text corpora. Documents were measured based on the prevalence of words found in **Figure 4**. Between 2019 & 2020, the predominance of ‘regulatory’-, ‘operational’, and ‘evaluation’-themed words remained constant among Item 1A risk sections.

Supervised Modeling: Logistic Regression

Logistic regression was used to model sentiment scores against outcome variables (change in revenue). The objective is to extract sentiment features from 10-K texts and predict change in revenue in terms of increase vs. decrease in revenue. Texts from all sections of 10-K documents will be included to examine sentiments holistically. After scraping texts from 10-K documents, the texts were cleaned by expanding contractions, removing stop words, html tags, and words shorter than four letters. Texts were the tokenized and unique tokens were extracted and stored in a dictionary. Functions were then defined to calculate term frequency based on predefined vocabulary from Lougran and McDonald Word List. This will compute sentiment features, which will be scaled before feeding into the logistic regression model. The final sentiment features are the following: positive score, negative score, modal strong, modal weak, uncertainty words, litigious score, polarity score, complex words, complex word count, word count, constraining score, positive word proportion, negative word proportion, constraining word proportion. Positive and negative scores demonstrate the extent of positive and negative sentiments. Modal strong and modal weak demonstrate the tone of the text, for example, “definitely” or “always” are strong modal, while “could” or “seldom” are weak modal. Polarity score was calculated using the following method: (positiveScore - negativeScore) / ((positiveScore + negativeScore) + 0.000001). The rationale is to show sentiment on a scale from negative to neutral to positive. Moreover, uncertainty score quantifies the ambiguity of the texts and contains vocabulary such as “appears” or “could”. Litigious words contain legal related terms such as “absolve” and “regulatory”. Complex words are compound words that might contain certain finance jargons such as “drawdown” and constraining words include “bounded”, “confined”, “forbid”, etc.

The logistic regression model had a high train precision of 0.818 for 0 (decrease in revenue) and 0.769 for 1 (increase in revenue). In comparison, the model had a lower test precision with 0.0 for decrease in revenue and 0.67 for increase in revenue. The low test precision is largely due to the small number of companies in the testing set (20). Overall, our model was able to predict revenue change with high train precision using sentiment features defined by Loughran-McDonald Sentiment Word List.

Discussion

We developed web-scraping, text parsing, and sentiment analysis tools that operated on documents from EDGAR, a database hosted by the Securities and Exchange Commission (SEC). Our analytical pipeline could aggregate text sections from 10-K filings, which enabled the development of both supervised and unsupervised learning algorithms. We analyzed whether underlying sentiments in 10-K paragraphs could indicate a company’s general risk profile, and thus predict how companies were impacted by the COVID-19 pandemic.

We developed an HTML 10-K parser that was generalizable (i.e. it could successfully extract risk sections from a heterogeneous mix of 10-K HTML tree hierarchies). Although 10-K filings are purportedly standardized, our database of underlying HTML trees varied, and thus developing a scalable method for isolating specific text sections was non-trivial. Our html_parser.py returned Nan values for a proportion of 10-K filings (~30%), which indicated unbeknownst bugs in our scraping algorithm. Despite these issues, our web scraper could return over two hundred ‘Item 1A Risk Sections’ for companies between 2019 and 2020. Developing a generalizable HTML parser enabled our pipeline to be scalable. Sentiment analysis modules were set up to accommodate any set of 10-K documents downloaded locally, which could then be synced to downstream models.

Our latent dirichlet allocation model could identify intra-document topic distributions among risk text corpora. Our initial LDA results had 5 topics set *a priori* (**Figure 2**) which were relatively undefined and had considerable overlap between topics. After tuning our model and designating 3 topics (**Figure 4**) we identified risk topics associated with financial evaluation, clinical trial operations, and FDA approval for patients. Unsurprisingly, these findings mirror the standard biotechnology business model, where risk often involves drug development and traversing the FDA regulatory landscape. We saw that biotechnology companies within the IBB index faced similar challenges before and during the pandemic.

Once topic-specific words were identified, 10-K documents were revisited and measured based on the prevalence of words found in **Figure 4**. Between 2019 and 2020, the predominance of ‘regulatory’-, ‘operational’, and ‘evaluation’-themed words remained constant among Item 1A risk sections (**Figure 5**). These findings suggest that COVID-19 failed to alter the risk landscape for biotechnology companies, which could be illustrated in the wordage used in SEC filings. In addition, our supervised model shows that the sentiments of 10-K files are correlated to revenue change. Using the sentiment variables derived from Loughran-McDonald Sentiment Word Lists, we were able to predict revenue changes (increase vs decrease in revenue) with 0.818 and 0.769 train precision respectively. This suggests that the pre-defined sentiment categories and their corresponding vocabulary were able to effectively capture the sentiments of 10-K filings, which are an indicator of revenue change. Thus, sentiment analysis using texts in financial filings can be used to evaluate and forecast the performance of biotech companies and potentially companies of other industries.

Overall, our project successfully processed limited, heterogeneous data (e.g. unique company 10-K HTML file structures) and produced preliminary insights through multiple models. Our analysis revealed that biotech companies in 2020 faced similar self reported risks compared to 2019. Companies that mentioned COVID-19 in risk text were also associated with vaccine-based clinical trials involving FDA approval.

Limitations and Future directions

Despite accomplishing our goal, there are plenty of opportunities to improve our analytical pipeline and ML models. Our modelling results were likely constrained by the limited number of companies within the IBB index. There were 177 rows after scraping revenue and 124 rows after removing NA. This limitation was demonstrated as very low test precision in the logistic regression model, especially low test precision for companies that had decreased revenue as there were only 6 of such cases. Moreover, sentiment features were computed using term frequency instead of vectorization since the document length made vectorization too computationally heavy. Since term frequency is a relatively primitive method that is not able to address the semantics of the text, we’d like to further explore more computationally efficient vectorization methods or perform the analysis on a more powerful server/cluster. Furthermore, to improve our LDA model performance, we could have expanded document analysis to regions of 10-K filings because just the Item 1A risk sections. In addition, any alterations to a company's 'true risk profile' due to the economic disruption caused by COVID-19 may take more than one year to manifest, thus rendering our 2019-2020 dataset too narrow to capture significant changes in 10-K filing wordage.

It is also important to mention that 10-K financial documents are specifically written and designed to only reveal limited information while meeting requirements by the SEC. The unfortunate truth is that companies are likely vague when assessing risk because it would be a competitive disadvantage to disclose more information than required. In part our results may potentially lack due to the methodology behind constructing financial statements. Despite our LDA model recognizing distinct topics, topic groups such as 'operational' and 'regulatory' were somewhat arbitrarily assigned.

Future work would include more systematic topic recognition to avoid human bias. It would also be interesting to test different topic recognition algorithms and compare those results to our LDA model. The framework of our pipeline is able to support much more data. With that said, we could apply another index and either merge or compare those results to our IBB corpus. Beyond building a larger corpus, we could also have lemmatized our documents for additional preprocessing. Our LDA model did not include bigrams, which may have provided more context.

Most of the challenges we faced involved formatting our parsed data as an input to our LDA model. It was difficult to write a function that appropriately extracted all the risk sections regardless of the difference between formats. In addition our initial results indicated no clear distinction between topics. This told us that our companies were extremely similar and we would have to focus more on post processing and methods to differentiate our topics.

Overall, although we were able to answer our initial question through our results, we did not discover any significant nor surprising results. It would have been fascinating to find distinct language within the 10-k documentation that provided more of an indication of performance but that was not the case. It is possible that we need to look at a longer time frame (beyond 2019-2020) since it might take longer than one year for the impact of COVID to be shown as additional business risks.

Navigating our Repository

Project Modules

Interacting with the EDGAR Database, and downloading desired financial documents with our Bulk_extraction function

- bulk_10k_extraction.py
- 10k_extraction.py

Generative absolute paths for locally downloaded 10-K HTML files. Creating .csv files with ticker symbols and parsed risk sections.

- path_mover.py
- structuring_data.py

Parsing 10-K filings while performing preprocessing

- html_parser.py

Parsing 10-K filings and supervised learning models

- financial_scrape_v1.ipynb

Unsupervised learning models, additional preprocessing, post processing, and plotting.

- topic_models.py
- topic_modeling_v2_2019.py
- topic_modeling_v2_2020.py

Relevant Datasets

LDA_and_revenue_v0.csv

- Preliminary dataset which contains LDA output, topic assignments, and financial metrics for biotech companies.
    - DominantTopic: Numeric value which classifies the assigned distribution the LDA model assigned to that particular document
    - Topic Percentage: Details the mixture percentage of the dominant topic for a document
    - Keywords: Top words the LDA decided best represented that particular document
    - Text: Tokenized list of risk text for a document
    - TCKR/company: Ticker symbol or name of company
    - Old_revenue: Revenue for 2019
    - New_revenue: Revenue for 2020
    - fc: Fold change in revenue between 2019 and 2020 [(new_revenue - old_revenue) / old_revenue)]
    - Delta: Whether fold change was positive or negative

sentiment_NLP.ipynb

- Sentiment analysis using logistic regression model
    - Clean raw text functions:
      expand_contractions: eg, :I'll" to "I will"
      Remove_accented_chars: remove unicode
      scrub_words: remove html tags and ascii characters
    - Compute sentiment features functions
      positive_score: compute positive word frequency
      negative_word: compute negative word frequency
      litigious _score: compute litigious word frequency
      modal_strong_score: compute strong modal word frequency
      modal_weak_score: compute weak modal word frequency

polarity_score:(positiveScore - negativeScore) / ((positiveScore + negativeScore) + 0.000001)
percentage_complex_word: percentage of complex words
complex_word_count: compute complex word frequency
total_word_count: compute word count of individual document
uncertainty_score: compute uncertainty word frequency
constraining_score: compute constraining word frequency
positive_word_prop: compute positive word proportion
negative_word_prop: compute negative word proportion
uncertain_word_prop: compute uncertainty word proportion

- vector_train: dataframe containing all sentiment features and corresponding revenue change in terms of 0 for decrease in revenue and 1 for increase in revenue
- y_train: training set revenue
- x_train: training set sentiment features
- x_test: testing set sentiment features
- y_test: traintestinging set revenue
- log_train_pred:
- log_test_pred:

Utilities

1. Updated_requirements.txt
2. NLP.yml

Appendix

1. Scraping: beautiful soup, Autoscraper, SEC_edgar_Downloader
2. Scraping EDGAR with Python (https://doi.org/10.1080/08832323.2017.1323720)
3. Pandas supported cleaning and preprocessing efforts,Pathlib for file navigation, NLTK for modeling , and both Matplotlib and Seaborn for data visualization.
4. SEC information (https://www.sec.gov/fast-answers/answersreada10khtm.html)

References

1. David Blei, Michael I. Jordan, Andrew Ng. Latent dirichlet allocation. Journal of Machine Learning Research 3 (2003) 993-1022.
2. Tim Loughran, Bill McDonald. When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10‐Ks. The Journal of Finance. Volume 66, Issue 1. February, 2011.