Categorizing Companies by Stated Risk Factors in 10-K Filings

Evie Wan, Nick Mosca, Eric South

[Draft v2] 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**).

In parallel, we referenced the **[name of resource]**, a dataset which associated common words in 10-K filings with sentiment categories (e.g., positive, negative, litigious). We used **[this resource]** for feature engineering, where companies were annotated with a mix of numeric features which described the sentiment across 10-K text. These processed datasets then fed into a logistic regression model, where we could predict whether a company had a positive or negative fold change in revenue between 2019-2020 with **[XX]** accuracy (**Figure 1**).

Approach

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 **[XXX]** 10-K filings from 2019 and 2020, which detailed financial performance among most 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. 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 generate new features for companies.

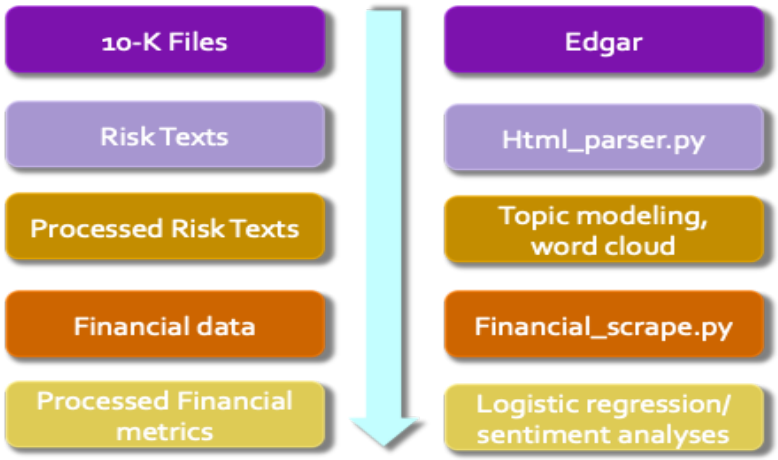

Figure 1. Overview of our analytical pipeline. After collecting 10-K filings we developed both supervised and unsupervised models to understand our data.

Unsupervised Model: Latent Dirichlet Allocation

To expedite the parsing of **[XXX, actual number**] HTML documents that were spread across different folders, we developed **[function name]** 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 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.

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_parset.py module) were broken down into 1) a corpus dictionary (i.e. unique set of words) and 2) 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. According to Investopedia, 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 generated by our LDA model were then visualized as word clouds, which included the top 15 words associated with each topic (**Figure 2**).

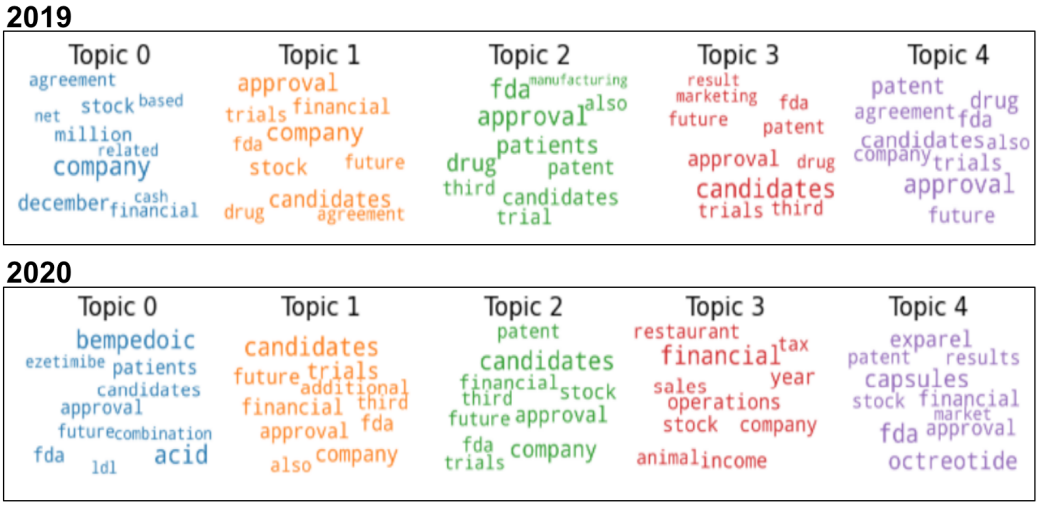

Figure 2. Top 15 words for each of the 5 topics generated by latent dirichlet allocation (LDA). Our LDA model designated a dominant topic to each document in our corpus (paragraphs of risk text subsections found among hundreds of company 10-K filings).

Beyond grouping companies into risk categories, financial performance (e.g., fold change in revenue) was compared pre- and post- COVID-19. Among biotechnology companies, we are interested in the relationship between 1) types of business model risk (as stated in SEC filings) and 2) financial growth during a pandemic. Our devised pipeline processed over 100 companies from the IBB index for both 2019 and 2020.
Column fold change, which indicates fold change [(new_revenue - old_revenue) / old_revenue)].

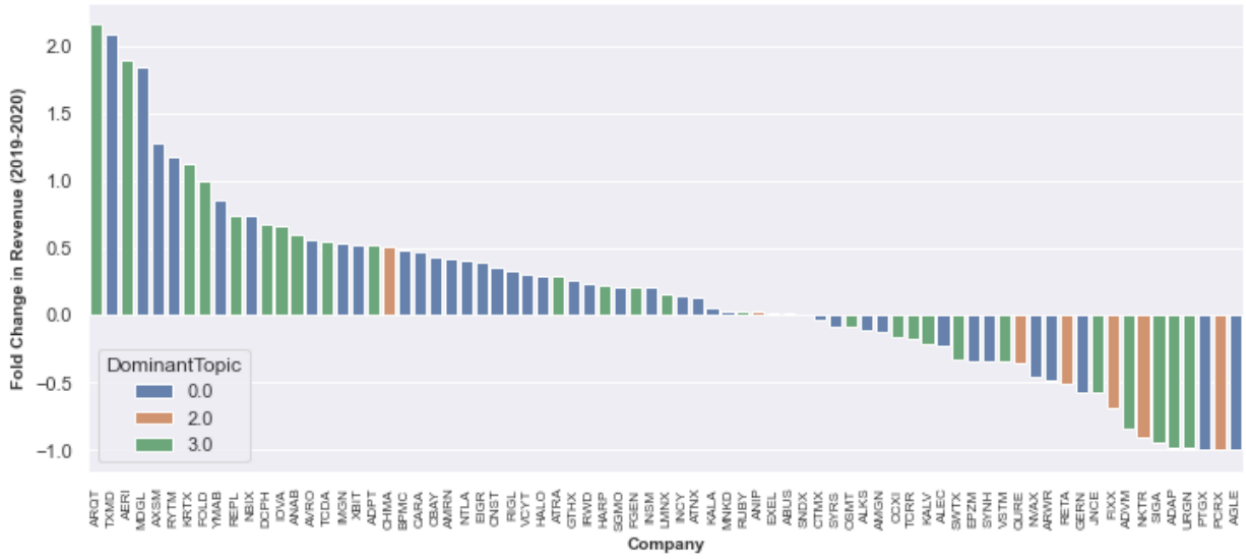

Figure 4. Incorporating 10-K revenue data for each company with dominant topics produced by the LDA model.

[paragraph on LDA struggles, which will segway into Nick's updated model]
Despite specifying 5 distinct topics in the LDA, only 3 topic groups (groups 0, 2, or 3) were ever dominant in a document. Oddly, LDA results led to extreme data skew towards one dominant topic (document's often had one topic predominant, opposed to an even distribution of 'dominant topics' across the entire corpus of documents. The problem is that uneven labelling of topics makes revenue comparison difficult. To achieve more 'even' outcomes…

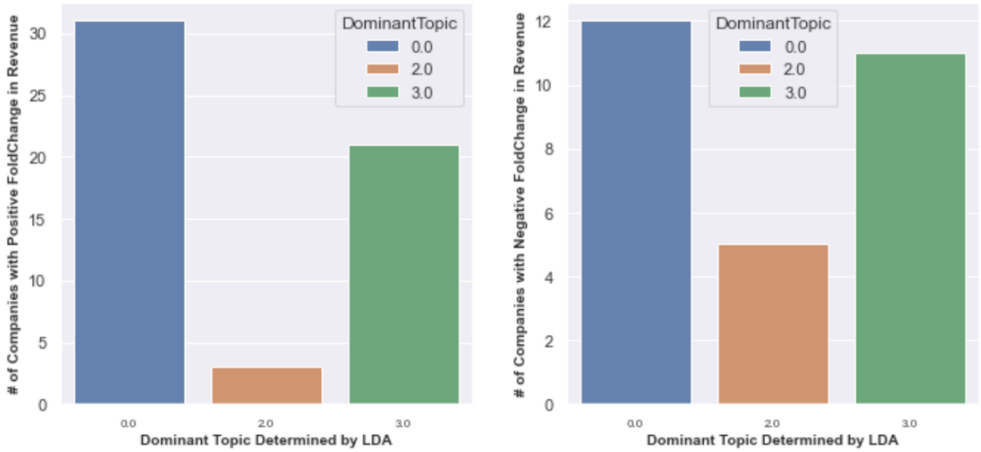

Figure 5. For each topic group, number of companies with positive fold change (i.e., how many companies performed well financially between 2019 and 2020, and are these numbers trending between topic groups?).

Refining our LDA (Nick)

Topic groups observed considerable overlap, which suggested that we needed to either tune our model or refine preprocessing steps (e.g. adding stopwords or improving upstream string cleaning).

(Nick's figure)
Figure X. Word clouds with alternative version of cleaned text

Supervised Model: Logistic Regression
Logistic regression was used to model sentiment variables against outcome variables (change in revenue). The objective is to predict increase or decrease in revenue based on 10-K texts. 10-k texts were first cleaned by expanding contractions, removing stop words, html tags, and words shorter than four letters. Sentiment scores were calculated using financial vocabulary from Lougran and McDonald Sentiment Word List. This dictionary was created specifically for financial textual analysis. Sentiment words are organized by category (negative, positive, litigious, strong, weak, constraining). Texts were tokenized and unique tokens were extracted and stored in a dictionary.

Model Performance:
The model had training precision of 0.81 for 0(decrease in revenue) and 0.77 for 1 (increase in revenue). Testing precision is 0.0 for decrease in revenue and 0.67 for increase in revenue.

Limitations:
Our data set is fairly small - 124 rows after removing NA. The data is also uneven - there are less cases of 0 (decrease in revenue) than 1 (increase in revenue). This largely contributed to the low test precision, especially low test precision for companies that had a decrease in revenue. There were only four cases of decrease in revenue and none of these were predicted correctly by the model.

Discussion

What did we accomplish?
What did we do well?
What could we improve upon?

Evaluating Performance of our HTML Parser

We've found that our HTML parser is fairly generalizable (i.e. can successfully extract risk sections from a heterogeneous mix of html data). Although 10-K filings are purportedly standardized, the underlying HTML tree can vary, and thus developing a scalable method for isolating specific text sections is non-trivial. We've found that our current html_parser.py returns nan values for a proportion of 10-K filings-- indicating unbeknownst bugs in our scraping algorithm. Despite these issues, our web scraper can return over 200 Risk Sections for companies between 2019 and 2020 (which we're satisfied with, as it'll provide an initial corpus for our topic modeling efforts).

Limitations

Although we were excited that our LDA model works, we were concerned that our compiled risk sections did not form distinct 'topic groups'. Upon initial analysis, many of the top words among topic groups were standard biotech business words (e.g., 'product', 'develop', 'regulatory', 'clinical'). We sought to refine our model to form more differentiable groupings.

Subject to change

If we cannot do this, we'll need to focus less on finding differences among documents and more on exploring trends among our risk corpus. Either way, the next phase of our project will predominantly focus on exploratory data analysis.

Navigating our Repository

Tools and Methods

Scraping: beautiful soup, Autoscraper, or EDGAR specific packages will be used to aggregate data from yahoo finance webpages.

- Scraping EDGAR with Python (https://doi.org/10.1080/08832323.2017.1323720)

Pandas supported cleaning and preprocessing efforts, scikit learn for cosine similarity and clustering, and both Matplotlib and Seaborn for data visualization.