Unsupervised Discovery of Firm-Level Variables in Earnings Call Transcript Embeddings

Daniel Edmiston* , **Ziho Park**
[1]University of Chicago, Chicago, IL, USA
{danedmiston, zpark}@uchicago.edu

Abstract

In this paper, we repurpose an algorithm from the computational biology literature to detect the extent to which modern document embedding methods are sensitive to financial and micro-economic variables. The contributions are two-fold. First, we provide a novel application of methods from an outside field and show its usefulness for unsupervised discovery of economic information in high-dimensional embeddings of financial documents. Second, we use the quantitative output of the algorithm to compare different embedding methods across different economic variables.

1 Introduction

The financial and economic worlds have never suffered for want of data, a fact which holds even more true in the modern era of big-data. In recent years, these worlds have seen a further explosion in the amount of data in the form of unstructured text. These new sources of data can be social-media interactions, internet news-stories, etc., in addition to the traditional text data available in the form of, for example, annual 10k filings and earnings call transcripts. Naturally, given the competitive advantage offered by the ability to better process data in all forms, considerable work in the economics and finance literature has gone into researching methods of processing these texts [Loughran and McDonald, 2016] to automatically detect and leverage the information contained therein, with a large portion of methods relying on discrete counts of words in documents.

In the somewhat far removed field of Natural Language Processing (NLP), researchers have gradually moved away from discrete, count-based methods since the early 2000's [Bengio *et al.*, 2003], all but eschewing them in modern research. Instead, virtually all current work in NLP is conducted representing linguistic entities (such as words, documents, etc.) with continuous, high-dimensional representations ($\in \mathbb{R}^n$), commonly referred to as embeddings. In addition to the advantages offered by the topology of the continuous space vs. a discrete one, researchers have also moved away from count-based methods altogether, instead favoring prediction-based approaches [Baroni *et al.*, 2014].

Given the applicability of modern NLP methods to text data in the financial world, it is perhaps surprising that these methods have not yet gained a foothold in mainstream finance and economics research. This is perhaps due to the fact that many of the variables (micro-)economists concern themselves with appear readily amenable to the simpler, and far-more interpretable count-based methods still popular in financial text-analytics. For example, [Hassan *et al.*, 2019] show that simple word counts from conference call transcripts provide an adequate proxy for political risk and associated sentiment, well correlated with other micro-economic variables. Conversely, in spite of the effectiveness of high-dimensional continuous embeddings for traditional NLP tasks, these representations remain opaque, and it is not clear how to mine them for economic information.

This paper hopes to provide a first step towards the discovery of economic information in high-dimensional embeddings, suggesting that it is indeed possible to discover information related to micro-economic variables within the seemingly opaque representations common in NLP. To do this, we build a graph structure based on the topology of embeddings built from earnings call transcripts, and apply the Spatial Analysis of Functional Enrichment (SAFE) algorithm [Baryshnikova, 2016] to this graph structure. The SAFE algorithm, originally developed in the computational biology literature, detects regions in a graph which show statistically high concentrations of values for a variable. The presence of high concentrations of values in a graph built on the topology of embeddings suggests that the embedding method used to create the graph is sensitive to that variable, and this is reflected in the partitioning of the space.

This paper then has the following as contributions. First, we repurpose an algorithm from the computational biology literature to show that the kind of firm-level variables (micro-)economists care about are reflected in the high-dimensional embeddings produced by modern NLP methods. This takes us a step closer to the interpretability of count-based methods, with the added advantages of a rich, continuous embedding space. Second, by using the quantitative output of the algorithm, we compare different document embedding strategies, showing that some methods capture certain economic variables better than others.

*Contact Author

It is our hope that work in this vein will serve as an encouragement to those in the finance/economic fields that modern document representation methods have value for their field, and we also hope to show the potential of the SAFE algorithm to the NLP field at large for the unsupervised discovery of information in high-dimensional embeddings.

2 Related Work

2.1 Text Mining in Economics

Text analysis in Finance/Economics has a long history, and a useful summary of methods in the field can be found in [Loughran and McDonald, 2016]. Here, we focus on the recent trend of mining textual data and treating the result as a proxy for economic variables. Such a method is exemplified in e.g. [Baker *et al.*, 2016], where the authors measure economic policy-related uncertainty via a word count-based index over newspaper articles. This proxy correlates with measured economic variables, such as observed political instability, and the proxy count variable is shown to peak at times of instability in the world at large, for example Brexit, and the election of Donald Trump as president of the United States.

Other exemplars of the kind of work often done in financial/economics text-analytics are [Hassan *et al.*, 2019] and [Hassan *et al.*, 2020], wherein the text-based variables the authors compute not only exemplify modern text-analysis research in the economics literature, but also use the same dataset as this study.[1] These studies employ count-based methods on bag-of-words text-representations to create proxy variables for risk/uncertainty and sentiment with relation to politics and the COVID-19 pandemic. As an illustration of their methods, the authors of [Hassan *et al.*, 2020] use the following formula to measure firm-level exposure to a disease (e.g. COVID-19).

$$Risk^d_{i,t} = \frac{1}{B_{i,t}} \sum_{b=1}^{B_{i,t}} \{\mathbb{1}[b \in Disease_d] \times \mathbb{1}[|b - r| < 10]\}$$

Here, $Risk^d_{i,t}$ measures the risk associated with disease d for firm i at time t, as measured by parsing firm i's earnings call transcript at time t. $B_{i,t}$ is the length of firm i's transcript at time t, and b ranges over the unigrams in the transcript. $\mathbb{1}$ is the indicator function, and $Disease_d$ is a dictionary of synonyms for disease d.[2] Finally, r is the nearest position of a word deemed synonymous with 'risk' or 'uncertainty', as determined by Oxford English Dictionary. The term $|b - r|$ then signals how close the nearest risk-meaning word is to current unigram b. In words, $Risk^d_{i,t}$ counts the number of occurrences of a unigram denoting virus d within a window of 10 words of a word denoting risk or uncertainty, and then normalizes for the size of the transcript.

In spite of the simple bag-of-words based methods employed by the authors, it is shown that these variables serve as reliable proxies correlating well with economic variables. Given the effectiveness of these methods for the variables in question (e.g. exposure to political risk), economics researchers can perhaps feel vindicated in their continued use of less linguistically sophisticated techniques.

While a large portion of mainstream literature in financial/economics text analytics does proceed with bag-of-words, count-based methods, there has been slight movement towards adopting the more modern methods as studied in the NLP literature. For example, [Araci, 2019] makes use of the recent trend of leveraging pre-trained language models [Devlin *et al.*, 2018] and applying them to sentiment analysis in the financial domain with encouraging results. [Hiew *et al.*, 2019] similarly use a pre-trained language model for sentiment analysis, further using the output for the task of stock-price prediction.

While there is no question that modern, continuous representations of linguistic units are superior for practical downstream tasks, it has yet to be shown that they are useful for the sort of text analytics e.g. [Baker *et al.*, 2016] practices. We reiterate our hope with this work to take the first step of showing that these continuous representations can also be interpreted as reflecting micro-economic variables.

2.2 Document Embeddings

As mentioned in the introduction, modern NLP research has all but abandoned discrete, count-based methods for representing linguistic entities in favor of continuous embedding methods, most often trained by neural networks. This holds true for representation of linguistic entities from the character level through to the document level. In this study, we examine three methods of embedding earnings call transcripts into continuous high-dimensional space, meaning that embeddings at the level of document are the focus of this study. While the literature on document embeddings has grown vastly in recent years, we restrict ourselves to discussing only the models relevant to this study. Specifically, we look at one count-based method of high-dimensional embedding, and two neural network-based methods.

The first method is Latent Semantic Analysis (LSA) [Deerwester *et al.*, 1990], which is based on the truncated Singular-Value Decomposition (SVD) of a matrix consisting of TF-IDF values [Jones, 1972]. This is the only count-based method investigated here, but due to the dimensionality reduction of the truncated SVD, the result is opaque continuous embeddings for each document in the corpus. TF-IDF/LSA can often serve as a strong baseline for document representation [Wang and Manning, 2012], and has the principled interpretation of document embeddings being projections of their weighted unigram word counts onto the principal directions of variation in the corpus.

The second method we examine is the Doc2Vec (D2V) method of [Le and Mikolov, 2014], which trains a document embedding by optimizing a vector to predict the words which appear in that document. This method, which can be considered a document-level extension of Word2Vec [Mikolov *et al.*, 2013], comes in two varieties, one which trains word-embeddings in tandem with the document embedding, and one which trains only the document embedding. The former generally shows better results, however the concatenation of

[1] Strictly speaking, we make use of only a subset of their dataset.

[2] For example, *coronavirus* and *COVID* would be treated as synonyms for the COVID-19 virus.

two embeddings (one from each variant) consistently showed the best results per [Le and Mikolov, 2014].

The third and final method of embedding documents is a variant of the Transformer architecture [Vaswani *et al.*, 2017], specifically Longformer [Beltagy *et al.*, 2020], which is an extention of the popular BERT model [Devlin *et al.*, 2018]. The principal modification of Longformer is a more memory efficient self-attention mechanism, allowing the model to more easily handle long documents such as the ones we deal with here. In this paper, we make use of two variants of Longformer. The first is a pre-trained model *as-is*, meaning we simply use the model with its pre-trained model weights and no additional fine-tuning. The second is the same model further fine-tuned on the TriviaQA dataset [Joshi *et al.*, 2017].

While BERT-style models have set the standard in recent years with regard to virtually every downstream task in NLP, one significant limitation of these models is their restriction to modeling sequences of fixed length. As this fixed length is typically 512 tokens,[3] BERT-style models are generally regarded as being less suitable for modeling long-form documents. Efforts to extend the fixed length are hampered by the fact that the self-attention mechanism on which the model relies is quadratic in both time and memory with respect to sequence length. To get around this, rather than attending to the entire sequence as self-attention typically does, the Longformer model uses a sliding window approach (*cf.* [Child *et al.*, 2019]) in addition to using sparse global attention. This way, longer sequences can be modeled, while no individual applications of self-attention extend past a reasonable limit (further details on the embeddings follow in Section 3.2).

Using these four methods to embed our corpus of earnings call transcripts, we hope to use the algorithm described in Section 3.1 to (i) show that micro-economic level variables can be discovered in them, and (ii) compare results between models and across variables as a first approximation of what kind of information might be available. Insofar as we are successful on these fronts, we will have shown that we can have the benefits of high-dimensional continuous embeddings (e.g. not requiring hand-crafted dictionaries [Loughran and McDonald, 2011], capturing rich semantic information, etc.), while maintaining some interpretability with regard to economic variables.

3 Experiment Preliminaries

3.1 SAFE Algorithm

In this paper, we make use of the Spatial Analysis of Functional Enrichment (SAFE) algorithm [Baryshnikova, 2016]. This graph algorithm was originally designed to detect the functional organization of large biological networks such as those representing relationships between genes. Here, we repurpose it to work over graphs constructed from document embeddings so as to ascertain the extent to which these embedding methods are sensitive to the information we're interested in.

[3]Tokens are roughly equivalent to words, though uncommon words are sometimes broken into multiple tokens.

The input to the SAFE algorithm is a graph $\mathcal{G} = (V, E)$ wherein each node $v_i \in V$ has associated with it a set of variables $\mathcal{X} = \{X_k\}$; write $X_{k,j}$ as the value of the k^{th} variable on the j^{th} node. Given such an input, the SAFE algorithm provides a means of identifying neighborhoods in the graph with statistically high concentrations of either low-values or high-values for variables X_k.

The node-level output of SAFE for variable X_k is a "neighborhood enrichment score"($\in \mathbb{R}$) which is calculated in the following way. For node v_i, define the neighborhood of v_i as $N(v_i) = \{v_j \in V \mid d(v_i, v_j) < \varepsilon\}$, where $\varepsilon \in \mathbb{R}^+$, and $d(\cdot, \cdot)$ is a distance metric on $\mathcal{G}$, e.g. shortest path. Define the observed score for variable X_k at node v_i as $O_k(v_i) = \sum_{v_j \in N(v_i)} X_{k,j}$. That is, $O_k(v_i)$ is the sum of values of variable X_k at nodes v_j for each node v_j in the neighborhood of v_i. Compare $O_k(v_i)$ with $n = 1,000$ random shufflings of the values of X_k across the nodes in the graph, thus producing a p-value for $O_k(v_i)$, call it $P_k(v_i)$. The neighborhood enrichment score of node v_i, denoted $NES_k(v_i)$, is the negative log transform of this p-value, i.e. $NES_k(v_i) = -log_{10}(P_k(v_i))$. Intuitively, nodes whose neighborhoods have scores for a variable resembling a random distribution will have high p-values, and accordingly low neighborhood enrichment scores. Alternatively, nodes whose neighborhoods commonly have very high (or very low) values will have relatively low p-values, and therefore high neighborhood enrichment scores.

Given node-level scores, one can describe a variable's distribution throughout the network with the score described in Equation 1. Total SAFE score, as a sum of all nodes' neighborhood enrichment scores, measures the extent to which a sample variable has high values or low values concentrated in specific neighborhoods in a graph. One refers to a variable with high (*resp.* low) SAFE scores as being highly (*resp.* poorly) enriched in a network. We interpret an embedding method as sensitive to a particular variable when that variable is highly enriched, and not so otherwise.

$$\text{Total SAFE Score}_k = \sum_{v_i \in V} NES_k(v_i) \tag{1}$$

3.2 Embedding Details

Given a corpus of 1,408 earnings call transcripts from the first quarter of 2020, we use the four models discussed above for embedding these documents into continuous, high-dimensional space. In the case of LSA, the TF-IDF vectors are calculated and then reduced via truncated SVD to a dimension of 800. For the Doc2Vec vectors, each document is trained for 10 epochs for each of the two variants of Doc2Vec (see Section 2.2), each in 400 dimensions. Each document is then represented as the concatenation of the resulting vectors from these variants, resulting in 800 dimensional embeddings, per the suggested specifications of [Le and Mikolov, 2014]. Finally for Longformer, both the pre-trained and fine-tuned models are BERT-Large variants, meaning they embed into 1,024 dimensions. To do the embedding, documents are split into chunks $c_1, ..., c_t$, each of 4096 tokens (except perhaps the last) with a stride of 500. Global attention is placed on the special [CLS] for each chunk, and the final embedding is the average of these [CLS] tokens.

For each of the models, we also compare a random baseline where the variables are permuted randomly throughout the network. It is sometimes common to evaluate models against a random baseline in which the embeddings themselves are randomized. However, as the output of SAFE is sensitive to graph structure, we use the same embeddings to ensure the same graph, and only randomly permute the variable values.

4 Experiment Design

4.1 Graph Construction

Given embeddings of the corpus documents, the first step is to construct a graph which reflects the topology of the distribution of the embeddings in the continuous space. Graph construction is done via an ϵ-radius graph; i.e. given document embeddings $X = \{x_1, ..., x_{1,408}\}, x_i \in \mathbb{R}^d$, construct graph $\mathcal{G} = (V, E)$ such that there is a bijection $f : V \to X$, and for $v_i, v_j \in V$, $(v_i, v_j) \in E$ *iff* $d(f(v_i), f(v_j)) < \epsilon$, where $d(\cdot, \cdot)$ is Euclidean distance in $\mathbb{R}^d$. In other words, there is a one-to-one correspondence between vertices in the graph and document embeddings, and there is an edge between vertices just in case their representative document embeddings are closer in $\mathbb{R}^d$ than ϵ.

The choice of ϵ is not arbitrary, but is again chosen to reflect the topology of the underlying embeddings. Specifically, ϵ is chosen such that resulting graph $\mathcal{G}$ has the "correct" number of connected components, where the number of connected components is decided by the *eigengap* heuristic [Von Luxburg, 2007].[4] The graph $\mathcal{G}$ is fed as input to SAFE for experimentation.

4.2 Discovering Micro-economic Variables

The experiment we describe here makes use of data collected from Computstat for the first quarter of 2020, in line with the timing of our earnings call transcripts. While the number of potential variables to experiment with from this data is large (in the thousands), here we focus on a subset of the variables, as listed in Table 1.

Along with the graph output discussed above, the values for these variables serve as input to the SAFE algorithm. The task is then to obtain $NES_k(v_i)$ for each variable k and transcript embedding represented by v_i, allowing us to calculate the Total SAFE score discussed in Section 3.1. This score can be taken as a measure for how much an embedding model has a tendancy to partition its embedding space by value of the respective variable. A low SAFE score indicates that the dispersion of the values of the variable is close to what would be expected by random assignment of values to nodes, whereas a high SAFE score indicates that certain nodes have neighborhoods which have significantly higher (or lower) values than would be expected by random assignment.

As a simple coarse measure of how much different embedding models reflect microeconomic information (at least for the variables chosen), we can average the Total SAFE scores for all relevant variables. The results of this experiment are in Table 2, along with the random baselines discussed above which serve as a control. For a more fine-grained analysis, we examine the Total SAFE scores for each of the individual variables. These results are in Table 3.

[4]That is, given the eigendecomposition of the laplacian of an affinity matrix (e.g. as given by a Gaussian kernel), sort the eigenvalues in ascending order and determine the number of clusters—or in this case, connected components—as k such that the gap between eigenvalues k and $k + 1$ is large.

Compustat abbr.	Description
actq	Current assets
altoq	Long-term Assets
chq	Cash
ciderglq	Derivatives gains/losses
cshtrq	Common shares traded
dlcchy	Changes in current debt
dlttq	Long-term debt
epsf12	Earnings per share
fincfy	Net cash flow
ivstchy	Short-term investments (change)
revtq	Total revenue

Table 1: Variables of interest for experiment. Abbreviations used as in Compustat database.

We stress that due to the lack of context with regard to SAFE scores—this being the first application of the method outside of biology, to the authors' knowledge—it is best to interpret model scores relative only to how they fare when compared against their random baselines. This is especially the case because SAFE scores are sensitive to the graph structure they're calculated on, and thus only models which share the same graph structure are directly comparable. As such, absolute SAFE scores are less informative for our purposes than the ratio of trained model performance to random model performance, as indicated in Tables 2-3. We interpret scores significantly higher than the random baseline as evidence of the information being reflected in the embeddings.

5 Results

Table 2 houses the scores for each model, along with its random baseline control where the values of the variables are permuted as discussed above. The results in the table are the Total SAFE Scores averaged over all variables. Again, this can be taken as a coarse measure of the embedding models' sensitivity to the variables.

Model	Trained	Random	Ratio
Latent Semantic Analysis	841	578	1.46
Doc2Vec	1909	751	2.54
Longformer	569	613	0.93
Longformer-finetuned	522	473	1.10

Table 2: Average Total SAFE Score across all considered variables. SAFE scores rounded to nearest integer, ratios to two decimal places. Red score indicates trained model below random baseline.

In three of the four cases, the trained model outscores its random baseline, but in the case of the non-finetuned Longformer, this is not the case. Doc2Vec, meanwhile, performs the best. Speculation as to why Doc2Vec performed the

Model	LSA			D2V			LF			LF-finetuned		
Variable	Trained	Random	Ratio	Trained	Random	Ratio	Trained	Random	Ratio	Trained	Random	Ratio
actq	970	558	1.74	2480	867	2.86	352	519	0.68	675	454	1.49
altoq	1246	614	2.03	2396	387	6.19	811	644	1.26	466	534	0.87
chq	1029	589	1.75	1663	945	1.76	567	599	0.95	683	286	2.39
ciderglq	420	491	0.86	1430	614	2.33	338	445	0.76	708	617	1.15
cshtrq	785	556	1.41	3360	583	5.76	641	755	0.85	711	665	1.07
dlcchy	351	643	0.55	1213	919	1.32	560	401	1.40	355	79	4.49
dlttq	914	659	1.39	1498	1125	1.33	420	636	0.66	211	200	1.06
epsf12	1115	534	2.09	2651	632	4.19	487	864	0.56	294	258	1.14
fincfy	412	619	0.67	828	498	1.66	916	602	1.52	171	125	1.37
ivstchy	709	510	1.39	470	386	1.22	686	635	1.08	843	1674	0.50
revtq	1295	593	2.18	3004	1302	2.31	478	643	0.74	629	306	2.06

Table 3: Total SAFE Score for each of the eleven variables of interest. All scores rounded to the nearest integer, ratios to two decimal places. Red denotes trained score for a model is below random baseline.

best and why the non-finetuned Longformer underperforms is withheld till Section 6.

In Table 3 are housed the Total SAFE Scores for each model and for each variable. This more fine-grained view of the scores shows that non-finetuned Longformer consistently produces scores near its random baseline. The LSA model and finetuned-Longformer models perform near-random on some variables, while showing strong results on others (e.g. *dlcchy*, 'Changes in current debt' for finetuned-Longformer). Doc2Vec on the other hand, outperforms its random counterpart on every variable, and significantly so in many cases, appearing particularly sensitive to *cshtrq*, 'Common shares traded.'

6 Discussion

Though the results are preliminary, it would appear that at least in some cases it is possible to show that high-dimensional embeddings have distributions correlated with micro-economic variables. For example, the topology of Doc2Vec embeddings seems to reflect the distribution of variables like *epsf12*, 'Earnings per share.' It stands to reason that if any type of economic information would be identifiable in these embeddings it would be the sort of variable referenced in conversation between shareholders and management. The issue of earnings per share for the quarter, along with topics like total revenue (variable *revtq*), are likely to be broached in earnings calls. Note that we do not interpret this as Doc2Vec being directly sensitive to values of variables like the number of shares traded, rather we interpret this as Doc2Vec (and similarly well performing models) being sensitive to language which likely correlates with variables like the ones discussed here.

As for relative performance of the models, it is not surprising that Doc2Vec outperforms the others. First, that it would outperform LSA is expected, as it has been shown that models trained on prediction tasks (like Doc2Vec) generally outperform those based on counts [Baroni *et al.*, 2014]. That said, even more traditional methods like LSA show potential for mining economic variables directly from high-dimensional vectors, as it showed strong performance on variables *revtq*, 'Total revenue,' and *epsf12*, 'Earnings per share'.

With regard to the two transformer-based models, it is a noted weakness of these models that they are not ideal at representing long sequences without fine-tuning on a downstream task. Particularly, it has been noted that the [CLS] token is likely not a good representation of the entire sequence without further task-specific training. As such, that the non-finetuned model would underperform its finetuned counterpart is to be expected.

As for the finetuned Longformer, its better-than-random performace is encouraging. This is because even though this model was finetuned on a general language understanding dataset, it resulted in embeddings which showed increased sensitivity to economic variables; i.e. finetuning is an effective means of creating representations more sensitive to the types of variables economists care about. Furthermore, it is likely the case that if a Longformer model had the benefit of further in-domain pre-training, it would significantly enhance the quality of the embeddings for this task [Gururangan *et al.*, 2020]. As such, we caution the reader against treating Longformer's poor performance relative to Doc2Vec as an indictment against Transformer models for this sort of task. Transformers have shown themselves invaluable for virtually every downstream task NLP practitioners care about, and with the proper training regimen it is entirely possible models like Longformer would be more competitive; we leave this for future work.

7 Conclusion

In this paper, we have hoped to show that the high-dimensional continuous representations common in NLP have potential for mining the sort of variables researched in economics and its neighboring discplines. Specifically, with the results above we have shown that certain continuous embedding models appear to partition their spaces in a way that correlates with certain firm-level variables. We take this as evidence of success with regard to the modest goals set out for this paper. Specifically, we hoped to show that modern algorithms and their representational techniques are sufficiently powerful to reflect the correlations of language as found in financial documents with that of certain economic variables. Furthermore, we have presented an algorithm from an outside field to aid in the resultant representations' interpretation.

References

[Araci, 2019] Dogu Araci. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*, 2019.

[Baker *et al.*, 2016] Scott R Baker, Nicholas Bloom, and Steven J Davis. Measuring economic policy uncertainty. *The quarterly journal of economics*, 131(4):1593–1636, 2016.

[Baroni *et al.*, 2014] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, 2014.

[Baryshnikova, 2016] Anastasia Baryshnikova. Systematic functional annotation and visualization of biological networks. *Cell systems*, 2(6):412–421, 2016.

[Beltagy *et al.*, 2020] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[Bengio *et al.*, 2003] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[Child *et al.*, 2019] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

[Deerwester *et al.*, 1990] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[Gururangan *et al.*, 2020] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*, 2020.

[Hassan *et al.*, 2019] Tarek A Hassan, Stephan Hollander, Laurence van Lent, and Ahmed Tahoun. Firm-level political risk: Measurement and effects. *The Quarterly Journal of Economics*, 134(4):2135–2202, 2019.

[Hassan *et al.*, 2020] Tarek Alexander Hassan, Stephan Hollander, Laurence van Lent, and Ahmed Tahoun. Firm-level exposure to epidemic diseases: Covid-19, sars, and h1n1. Technical report, National Bureau of Economic Research, 2020.

[Hiew *et al.*, 2019] Joshua Zoen Git Hiew, Xin Huang, Hao Mou, Duan Li, Qi Wu, and Yabo Xu. Bert-based financial sentiment index and lstm-based stock return predictability. *arXiv preprint arXiv:1906.09024*, 2019.

[Jones, 1972] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 1972.

[Joshi *et al.*, 2017] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

[Le and Mikolov, 2014] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196, 2014.

[Loughran and McDonald, 2011] Tim Loughran and Bill McDonald. When is a liability not a liability? textual analysis, dictionaries, and 10-ks. *The Journal of Finance*, 66(1):35–65, 2011.

[Loughran and McDonald, 2016] Tim Loughran and Bill McDonald. Textual analysis in accounting and finance: A survey. *Journal of Accounting Research*, 54(4):1187–1230, 2016.

[Mikolov *et al.*, 2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[Von Luxburg, 2007] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[Wang and Manning, 2012] Sida Wang and Christopher D Manning. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th annual meeting of the association for computational linguistics: Short papers-volume 2*, pages 90–94. Association for Computational Linguistics, 2012.