NBER WORKING PAPER SERIES

TEXTUAL FACTORS: SCALABLE, INTERPRETABLE, AND DATA-DRIVEN APPROACH TO ANALYZING UNSTRUCTURED INFORMATION

Lin William Cong
Tengyuan Liang
Xiao Zhang
Wu Zhu

Working Paper 33168
http://www.nber.org/papers/w33168

NATIONAL BUREAU OF ECONOMIC RESEARCH
1050 Massachusetts Avenue
Cambridge, MA 02138
November 2024

We are especially grateful to Agostino Capponi, Gerard Hoberg, and Gustavo Schwenkler for detailed comments and direction. We thank Chunrong Ai, Kwan Chen, Tony Cookson, Tarek Hassan, Shiyang Huang, Sanya Kohli, Kai Li, Alejandro Lopez-Lira, Nadya Malenko, Alan Moreira, Deniz Okat, Lubos Pastor, Lauren Sutioso, George Tauchen, Baozhong Yang, Weiyi Zhao, and seminar and conference participants at the AEA/CERNA Joint Meeting, Ansatz Capital, Conference on Big Data, Machine Learning and AI in Economics, Baidu Du Xiaoman Financial, DataYes/KDD China AI x FinTech Workshop, Erasmus University (Rotterdam), Financial Intermediation Research Society Annual Conference (Savannah), Global Digital Economy Summit for Small and Medium Enterprises (DES2020), Guanghua International Symposium, University of Hong Kong, Hong Kong University of Science and Technology, INQUIRE Europe Autumn Seminar (Krakow), IIF International Research Conference & Award Summit (Delhi), JD.com JDD (Financial Arm), Kenan Institute Frontiers of Entrepreneurship Conference, Nanyang Technological University, New Technologies in Finance Conference (Columbia GSB), 1st NY Fed FinTech Research Conference, Singapore Management University, Tilburg University, the Second Toronto FinTech Conference, and the Zhongnan University of Economics and Law for their feedback and suggestions. Michael Fortunato, Fujie Wang, Oliver Xie, and Guanyu Zhou provided excellent research and programming assistance. Thanks also go to Shuyan Huang, Chloe Shin, Raj Shukla, Ellis Soodak, Jiashu Sun, and Connie Xu for their research assistance. We gratefully acknowledge the financial support from the Ewing Marion Kauffman Foundation, the Becker Friedman Institute of Economics, the Fama-Miller Center for Research in Finance, INQUIRE Europe, the Kenan Institute of Private Enterprise, and the Risk Institute at OSU Fisher College of Business (while Cong was a fellow at the institute). The contents of this publication are solely the responsibility of the authors. Please send correspondence to Cong at will.cong@cornell.edu. The views expressed herein are those of the authors and do not necessarily reflect the views of the National Bureau of Economic Research.

Textual Factors: A Scalable, Interpretable, and Data-driven Approach to Analyzing Unstructured Information

Lin William Cong, Tengyuan Liang, Xiao Zhang, and Wu Zhu

NBER Working Paper No. 33168
November 2024
JEL No. C13

ABSTRACT

We introduce a general approach for analyzing large-scale text-based data, combining the strengths of neural network language processing and generative statistical modeling to create a factor structure of unstructured data for downstream regressions typically used in social sciences. We generate textual factors by (i) representing texts using vector word embedding, (ii) clustering the vectors using Locality-Sensitive Hashing to generate supports of topics, and (iii) identifying relatively interpretable spanning clusters (i.e., textual factors) through topic modeling. Our data-driven approach captures complex linguistic structures while ensuring computational scalability and economic interpretability, plausibly attaining certain advantages over and complementing other unstructured data analytics used by researchers, including emergent large language models. We conduct initial validation tests of the framework and discuss three types of its applications: (i) enhancing prediction and inference with texts, (ii) interpreting (non-text-based) models, and (iii) constructing new text-based metrics and explanatory variables. We illustrate each of these applications using examples in finance and economics such as macroeconomic forecasting from news articles, interpreting multi-factor asset pricing models from corporate filings, and measuring theme-based technology breakthroughs from patents. Finally, we provide a flexible statistical package of textual factors for online distribution to facilitate future research and applications.

Lin William Cong
SC Johnson College of Business
Cornell University
Sage Hall
Ithaca, NY 14853
and NBER
will.cong@cornell.edu

Tengyuan Liang
University of Chicago
5807 S Woodlawn Ave
Chicago, IL 60637
Tengyuan.Liang@chicagobooth.edu

Xiao Zhang
Compass Lexecon LLC
xzhang@compasslexecon.com

Wu Zhu
Tsinghua University
zhuwu@sem.tsinghua.edu.cn

1. Introduction

Unstructured data are prevalent and contain rich information. For example, because social communications are primarily mediated in languages rather than numbers, texts have not only enabled econometricians to complement traditional data or surveys but also allowed practitioners to capture more granular and up-to-date information (e.g., from news). However, extracting textual information that preserves computational efficiency and economic interpretability has been challenging. Standard techniques such as word counts rely on domain expertise, and can be highly reductive or application specific. A data-driven systematic framework that links large corpuses of unstructured documents to structured and interpretable regression analysis is lacking in social science studies.

We tackle the challenge by drawing insights from both neural network models for natural language processing (NLP) and topic models in statistical machine learning. We develop *textual factors* (TFs), structured representations of texts preserving their syntactic and semantic information so that an entity of interest's covariations with different topics and themes can be systematically studied, much like factors in asset pricing models, but with the interpretability that natural languages offer. Our conceptual innovation therefore lies in highlighting for the first time how combining the strengths of word embedding and topic modeling can advance text analytics on multiple fronts. Our key technical innovation entails using practical clustering methods developed in theoretical computer science to achieve this, while attaining performance comparable to other cutting-edge methods that are computationally expensive or lacking in interpretability.[1] We demonstrate the efficacy of TFs through validation tests before applying the framework in various economic settings.

We generate TFs in three steps. We start with a continuous vector embedding of words in textual data using neural networks (Word2Vec) to preserve the semantic and syntactic meaning of words in a given context, which captures rich information and language nuances. We then cluster words based on their vector embedding using Locality-Sensitive Hashing (LSH), a large-scale approximate nearest-neighbor search algorithm to further reduce the dimensionality of our textual space, create "separability" of word clusters for efficient computation, and orient proximate vector representations for easy interpretation. Finally, we overlay a topic model to enhance interpretability by describing the frequency distributions of TFs and of supporting words within each TF. Though other clustering algorithms exist, LSH is particularly suitable for TFs because it reduces the dimensionality of embedded word vectors based on the textual context and enables a fast search

[1] One cannot overestimate the difficulty in parsing millions of documents and synthesizing the information content in texts into quantitative features. Moreover, social scientists rarely have the computational resources as tech firms such as OpenAI or are willing to supply proprietary data to black box algorithms.

to generate word supports for the topics. Our framework allows combining domain knowledge and big data to generate TFs and handle the typical low signal-to-noise data in economics and finance.

To highlight our conceptual contributions, we summarize in Figure 1 the trade-offs and limitations of existing methods. Admittedly, pre-defining dictionaries or manually adding labels in count-based textual analyses achieve computational efficiency and interpretability. Therefore, in economics and finance, Antweiler and Frank (2004) and Tetlock (2007) initially used the Naive Bayes algorithm and supervised word counting approaches. Related studies (e.g., Gentzkow and Shapiro 2010, Hanley and Hoberg 2010, Hoberg and Phillips 2010, Engelberg and Parsons 2011, Loughran and McDonald 2013, Jegadeesh and Wu 2013, Baker et al. 2016, Hassan et al. 2019, Schwenkler and Zheng 2020b) employ task-specific text representations, and often rely on researchers to pre-specify the "right" word lists.[2]

We recognize that researchers have also applied data-driven NLP tools involving structured statistical models under well-behaved distributional assumptions (e.g., Manela and Moreira 2017, Chen et al. 2019, Schwenkler and Zheng 2020a, in economics and finance). In particular, Bellstam et al. (2021) and Jegadeesh and Wu (2017) use LDA to measure corporate innovation or quantify the policy content of Federal Reserve communications. These direct topic modeling applications may miss complex linguistic structure and face limited scalability when handling large textual datasets (Wallach et al. 2009, Mikolov et al. 2013b), not to mention that LDA analysis often involves iteratively removing words incoherent with topics at researchers' discretion. In contrast, TFs provide a complementary data-driven unsupervised/semi-supervised approach that captures text dynamics and generates new insights at the same time.

Recent developments in neural network language models (e.g., Word2Vec, BERT) present an alternative approach that is computationally comparable but better preserves the syntactic and semantic structure than topic models. However, naive applications of them fall short of providing the interpretability and transparency demanded in social sciences (Cong et al. 2021). One exception attaining interpretability is Li et al. (2021) which uses Word2Vec to extend a pre-defined dictionary on corporate culture. TFs achieve interpretability from data-driven word clusters and offer a general approach to analyzing texts in a simple regressional framework.

In summary, we strive to balance and improve all three dimensions of textual analysis (and, more broadly, unstructured data analytics): computational scalability, linguistic complexity, and economic interpretability. The closest study that also combines semantic vector analysis and topic modeling is Hanley and Hoberg (2019), which overlays Word2Vec on LDA topics to detect emerging

[2] A novel and data-driven development includes extending the static "bag-of-words" approach to dynamic and customized dictionaries (Hoberg and Phillips 2016), or using revisions of documents (Brown and Tucker 2011, Hanley and Hoberg 2012, Cohen et al. 2020).

risks. Cleverly and effectively crowd-sourcing information from both banks and investors, the paper complements ours with different methodologies and focuses. The authors begin with LDA clustering, then enhance its interpretability using the weighted average of the word embedding within each cluster, and take the cosine similarity between each topic and the document as topic exposure. Instead, our methodology explores scalability and topic separability. Applying word embedding first in TF, as opposed to applying LDA first, also allows us to extract richer information from the primary texts.

After introducing the TF foundations and implementations, we illustrate their interpretability, versatility, and computational efficiency through multiple "validation" exercises and applications. We first compare TFs with the topic outputs from plain-vanilla LDA models, then test the robustness and sensibility of loadings on our constructed TFs. For example, we inspect the word clusters derived from the 1889-2018 *Wall Street Journal* (WSJ) data for representative TFs such as "Recession," "War," and "Computer." We find TFs are more interpretable, intuitive, and deployable in nowcasting and industry applications with large amounts of near real-time texts.

More importantly, we can integrate TFs with simple regressions to conduct three types of analyses allowing a mixture of human-defined and data-automated topics. First, exposure to various TFs can help predict or explain outcomes in cross-section, time series, and panel data analysis, complementing conventional data with texts. We show that TFs can better forecast various macroeconomic variables than both one-hot count-based methods and plain-vanilla LDA models by significant margins. They also provide predictive power when added to simple models with structured data, such as the commonly used AR(1) models. Moreover, the performance is comparable to cutting-edge LLMs using a similar data set (e.g., Chen et al. 2023) or existing text-based methodology using richer data sets (e.g., Kelly et al. 2021a).

Second, TFs offer interpretations of explanatory variables, coefficients, or behaviors of non-text-based models, such as factor models in asset pricing. We use TFs to explain the cross section of firms' loadings on the Fama-French factors, revealing that topics such as financial conditions, industry specifics, and share issuances and repurchases are essential in understanding a firm's exposure to asset-pricing factors. We also find that TFs explain and predict factor risk premia better than one-hot benchmark models out of the sample. We further compute TFs and loadings from firm filing data to interpret deep reinforcement learning models for investment introduced in Cong et al. (2019).

Third, TFs allow a data-driven way of constructing explanatory variables with economic interpretations, which can capture often elusive concepts such as corporate governance and innovation. Using patent abstracts and claim data for all patents issued from 1976 to 2023 by the United States

Patent Office (USPTO), we are able to learn the dynamics of theme-based innovations and construct metrics to capture technology breakthroughs and dynamics in patent quality in aggregate and by innovation themes. This last dimension also points to the possibility that TFs create new knowledge and open new frontiers of analysis.

After our initial draft in 2018, embedding algorithms beyond Glove/Word2Vec/Doc2Vec have since emerged and LLMs (e.g.,BERT and ChatGPT) became popular, especially with economists (e.g., Kim et al. 2024, Chen et al. 2022, Lopez-Lira and Tang 2023, Acikalin et al. 2022, Schwenkler and Zheng 2024). However, LLMs are complements rather than substitutes for TFs because LLMs are primarily word-embedding models; TFs can incorporate them in the first step to enhance their interpretability. More generally, transparent TFs with public source codes allow researchers to adapt, train and fine-tune their own model to specific domains and use cases, whereas interpretations generated by black-box LLMs themselves are subject to misrepresentation (e.g., Zhang et al. 2023, Xu et al. 2024).[3] Moreover, TFs are easily scalable, considering that parsing millions of long documents remains computationally intensive and time-consuming even with parallel computing.[4] Finally, even with the simplest data available, TF performance is comparable to more complex models using richer data (e.g., in macro forecasting as done in Chen et al. 2023).

Our goal is not to exhaustively discuss TF applications or to run a horse race with other methods. We introduce the concept of combining two important NLP models through clustering algorithms such as LSH and demonstrate how the methodology is applicable and sound in economic studies. Our data choices in the illustrations are also deliberately minimalistic. Our emphasis on interpretability adds to recent advances and debates concerning AI and machine learning in business and social sciences.[5] We provide detailed code for generating TFs in our statistical package (https://github.com/textualfactor/Text_Analysis), which is flexible and already allows extensions by other researchers to study technology innovation and corporate fraud (e.g. Kogan et al. 2023, Cherepanov et al. 2024). In general, we contribute to text-based analytics in social sciences (e.g., Grimmer and Stewart 2013, Evans and Aceves 2016, Gentzkow et al. 2019) as a first attempt at factor-based, data-driven analytics to capture rich textual information, balancing computational efficiency and economic interpretability.[6]

[3] Relying on LLM outputs for interpretation may also lead to the issue of over-interpretability without understanding the mechanisms behind the embeddings.

[4] Some researchers may resort to smaller LLMs like BERT, but this approach is often insufficient or too costly, as pointed out by Chen et al. (2023). ChatGPT API processes only tens of news abstracts per second using a GPU. In contrast, TFs can be trained and fine-tuned within a reasonable amount of time on personal computers.

[5] Only with interpretability can one aptly adjust complex models in the ever-changing business and financial environments, not to mention that computational efficiency is needed for low latency text analytics in practice.

[6] We are also the first to apply cutting-edge LSH, widely adopted in areas such as information retrieval and fingerprint matching (Datar et al. 2004, Andoni et al. 2015), to NLP in computer science.

2. The TFs Framework

TFs integrate two important NLP developments in statistical generative modeling (e.g., topic models, Blei et al. 2003), and black box machine learning epitomized in neural network language models (e.g., Bengio et al. 2003). In stage I, we embed texts in an abstract vector space and identify a small number of interpretable basis factors that "span" the "natural language space" (i.e., capture main variations in the texts). In stage II, we decompose each unit sample of texts using their loadings on these TFs for downstream predictions or inferences.

Stage I further comprises three steps: (i) learn, in a distributed fashion, a continuous vector embedding of the texts in a large vocabulary using neural networks (e.g., Word2Vec) to capture the semantic and syntactic meanings of words; (ii) cluster words based on their vector embedding using a large-scale approximate nearest-neighbor search LSH; and step (iii) employ topic modeling to extract interpretable factors, based on the data-driven word-clusters hitherto obtained.

Stage I aims to generate TFs to adequately represent the information in the texts, allow fast computation, and ensure economic interpretability. To this end, step (i) helps capture complex language structure and rich information in the data and, at the same time, reduces the dimensionality of the analysis; step (ii) ensures both computational efficiency via a further reduction in dimensionality and economic interpretability via generating relative independent vocabulary supports of TFs ("separability"); step (iii) enhances the interpretability by describing frequency distribution over TFs and the supporting words within each factor.

2.1. Stage I(i): Word Embedding Through Semantic Vector Representation

Social scientists traditionally use 'one-hot' representation to understand words or phrases (N-grams), representing each word by an orthogonal vector with a "1" in one position and "0"s elsewhere. This method fails to capture the complex relationships and similarities between words, a limitation "word embeddings" addresses by representing words as related vectors in a multi-dimensional space. Each word is mapped into a vector that retains its semantic and syntactic characteristics, ensuring that similar words are closely positioned in the space (see Appendix A).

An inherent trade-off between variance and bias in word embeddings poses a challenge, especially with limited data. Retraining word vectors under such conditions can increase variance and degrade model performance. Therefore, we recommend using or fine-tuning open-source pre-trained word embeddings like Google's Word2Vec and GloVe, which leverage extensive datasets like Google News, Wikipedia, Gigaword, Common Crawl, and Twitter.[7] These pre-trained models offer a foundation of general linguistic relationships that can significantly reduce variance when training data is sparse. We caution that in the absence of suitable pre-trained embeddings and

[7] Other open-source embeddings include FastText by Facebook, BERT by Google, and GPT by OpenAI.

faced with limited data, clustering solely based on word embeddings may inadvertently amplify noise. Finally, researchers with ample corpora can create domain-specific embeddings using the open-source library Gensim (https://radimrehurek.com/gensim/), which we integrate into our TF package. Hoberg et al. (2024) also found that domain-specific embeddings using corporate web text can significantly improve model performance. This approach improves the robustness and relevance of the analytical framework for specialized datasets.

2.2. Stage I(ii): Scalable Clustering through Locality-Sensitive Hashing

The semantic vector representation reduces dimensionality compared to "one-hot" encoding but remains high-dimensional (p typically being a few hundred) and lacks thematic interpretation. To address this, we cluster words based on their embeddings, which capture semantic and syntactic essence. However, clustering in high dimensions is computationally challenging, with a complexity of $O(V^2p)$, where V can reach tens of thousands or millions in real applications. This challenge arises from the need to find each vector's nearest neighbors through pairwise distance calculations. We use LSH and its variants to balance interpretability and computational complexity (Leskovec et al. 2020). LSH efficiently finds a vector's nearest neighbors with high probability, though not always perfectly, significantly reducing computation time.

We describe the intuition behind LSH next, leaving more technical details to Appendix B. LSH uses hash functions to sort items (e.g. vectors) into "buckets" based on their similarity to simplify the search for similar items in large datasets (searching the buckets rather than the entire dataset). For example, consider H as a set of all hyperplanes in $\mathbb{R}^p$. Each hyperplane divides the space into two subspaces (buckets). Thus, a sequence of N random hyperplanes (a hash table) would divide the space into 2^N buckets. Vectors within the same bucket are likely to be close, reducing computation time. The probability that vector v and its nearest neighbor do not fall in the same bucket is $1-(1-p_1)^N$.

To reduce this misgrouping, multiple hash tables (say L hash tables) can be used. For each hash table, the bucket containing v's neighbors is identified, and the search is conducted across all L buckets (one for each table), involving $L \times V/2^N$ similarity computations. The probability that v and its nearest neighbor fall into at least one of the L buckets is $1-[1-(1-p_1)^N]^L$. Fine-tuning the hyperparameters N and L improves the probability of finding the actual nearest neighbor while enhancing computation efficiency significantly. In our empirical work, we fine-tune the hyperparameters such that the accuracy of finding the actual nearest neighbor is at least 90%.

Leveraging LSH, we introduce two scalable clustering Algorithms 1 and 2 in Appendix C. Plausible choices of subroutines of the algorithm can be found in Leskovec et al. (2020) (See Section 3.4.3 for *lsh-cand-pairs*, *lsh-near-neigh* and Section 7.2.1 for *cluster-roid*). The algorithms take as

input the embedding vectors for all words and an efficient LSH algorithm as a subroutine, then output word clusters. The words in the same output cluster inherit the neighboring or similarity structure from the LSH algorithm. To demonstrate the quality of the clusters and the scalability of the methodology, we test it on the 200K-word vocabulary provided by Google in Section 2.4.

2.3. Stage I(iii): Guided Topic Modeling: A "Clustering" View

We next use the clusters obtained to enhance a topic model. Based on the semantic similarity among words captured by vector embeddings, similar words are more likely to belong to the same topic. This prior knowledge significantly reduces the search space and complexity of topic-word distributions, easing optimization and enhancing interpretability.

Topic models essentially conduct matrix factorization of text documents (Figure 2). The frequentist approach, Latent Semantic Analysis (LSA), uses singular-value decomposition of the document-word matrix (Appendix D, Eq. 2). In contrast, the Bayesian approach, represented by Latent Dirichlet Allocation (LDA), takes a probabilistic view of word distributions within topics and topic distributions within documents (Appendix D, Eq. 3). LDA typically requires MCMC/variational methods, which can be slow (and computationally expensive) or non-convergent. Moreover, common words tend to dominate all topics, resulting in limited "separability." We therefore advocate a "clustering" perspective of topic modeling and the LSA approach.[8]

Specifically, using the word clusters from Algorithms 1 and 2, we introduce computationally efficient and conceptually simple methods to learn TFs, alleviating the drawbacks of the popular LDA approach. The key insight is that, given the separability of clusters, a large document-term matrix can be reduced into smaller submatrices to be factorized in parallel or sequentially. Because the vocabulary supports of topics are relatively disjoint (words with specific meanings do not appear in multiple clusters), the topics are distinct from each other. This contrasts with plain-vanilla LDA, where common words (or stop words) dominate multiple topics (Wallach et al. 2009). For example, given the i-th cluster with word support $S_i \subset [V]$ where $[V]$ is the set of all vocabularies, we focus on the document-term submatrix N_{S_i} where the columns consist of words only in the i-th cluster.

For implementation, we can apply matrix factorization techniques to extract meaningful insights. We use the singular value decomposition (SVD) for the document-term submatrix. Denote the factor by $F_i \in \mathbb{R}^{|S_i|}$, which satisfies $\|F_i\|_{\ell_2} = 1$, and the topic importance by $d_i \in \mathbb{R}$. We define F_i as the top right singular vector of matrix N_{S_i}, and the topic importance is given by the top singular value $d_i = \|N_{S_i} F_i\|$. Appendix E describes an alternative LDA based on topic-word distribution.

[8] Without the separability of topics, it is very hard to disentangle various topics using only the document-term matrix. In fact, it is probably NP-hard (Sontag and Roy 2011) without additional structure when the number of topics is large. A word with different meanings in different contexts can still appear in multiple topics or clusters unless we remove it from the vocabulary list once it is included in a cluster.

2.4. Stage II: TF Loadings in a Linear Framework

Suppose Stage I yields K TFs, where K is chosen based on data and specific applications. Denote the set of TFs by the triplet $(S_i, F_i \in \mathbb{R}^{|S_i|}, d_i \in \mathbb{R}_{\geq 0})$, where S_i represents the support of word-cluster $i = 1, \cdots, K$, F_i is a real-valued vector representing the TF associated with cluster S_i, and d_i is the factor importance. F_i captures the relative importance of words in cluster S_i. A new data-point (document d), represented by a document-term vector $N^{(d)} \in \mathbb{R}^V$ (L_2-normalized) where V is the number of words, has a loading on TF i given by the projection:

$$x_i^{(d)} := \frac{\langle N_{S_i}^{(d)}, F_i \rangle}{\langle F_i, F_i \rangle}, \tag{1}$$

and document d can then be represented by $(x_1^{(d)}, \ldots, x_K^{(d)}) \in \mathbb{R}^K$.

To understand the meaning of these loadings, consider a publicly listed firm. In structured data, the company discloses numbers on revenues, profits, liabilities, etc., each providing insight into a specific performance dimension. In the realm of unstructured information, texts about the company may center on topics like profitability, social responsibility, and innovation. The loading $x_i^{(d)}$ measures how closely the company is associated with a particular topic factor, serving as a metric that can be used in the conventional regression framework.[9]

2.5. Task-Specific Factor Selection and Seeded TFs

Despite our framework being data-driven, we advocate using relevant features selected by domain experts whenever available. TF is flexible to incorporate such pre-existing domain knowledge or pre-specified topics. For example, if we believe that specific topics or keywords are essential, we can seed them when we group word vectors.[10] In Algorithm 1, one can first merge clusters containing seeded words or use the seeded words as cluster-roid. In Algorithm 2, one orders the sequence of words to be considered so that all the seeded words appear first. The final output of the clustering stage is a mixture of seeded topics (TFs) and other data-driven topics discovered.

Data from financial markets and economic interactions tend to have low signal-to-noise and may generate spurious TFs that reduce the interpretability of the outputs. To deal with this, we can

[9] In an LSV implementation of the guided topic modeling, $x_i^{(d)}$ comes from the SVD outputs (the product of the matrix of left singular vectors and the one with singular values). The matrix of right singular vectors alone is often sufficient because skipping the scaling with singular values is equivalent to normalizing the input variables in downstream regressions.

[10] Suppose we want to use social media texts to predict credit outcomes for individuals. Using the plain-vanilla TF framework, we would let the entire corpus of textual information on social networks generate the TFs. However, experience and economics may tell us that issues related to borrowing and loans should be relevant, including topics on income, family support, etc. To incorporate domain knowledge, we can use "loan," "income," etc. as seeded words and require some factors to be clustered around them before we go through the Stage II of topic modeling. Note that our seeded TFs contain both the word support and relative frequency of words and, therefore, are more comprehensive compared to simply defining a bag of words or searching for synonyms. Moreover, the approach is flexible, and alternative subroutines can be developed.

(i) make TFs more precise by removing infrequent words or stopping words — the data-cleaning routine for textual analysis; (ii) use thresholds of correlation of TFs with the dependent variables or Lasso to select features for simplicity and efficiency; or (iii) use domain expertise and human judgment to manually select clusters or remove irrelevant ones.[11]

3. Preliminary Validation of the TF Framework

3.1. Data

We use titles and abstracts of all front-page articles in WSJ from July 1889 to April 2018 for simple TF inspections. Appendix F displays samples of the data, which are widely used in academic studies (e.g., Tetlock 2007, Manela and Moreira 2017, Kelly et al. 2021a).[12] For cross-sectional analysis, we use company filings from SEC EDGAR. Cong et al. (2021) provides more detailed descriptions of other sources of data that are available to researchers including patent filings (e.g., USPTO), analyst reports (e.g., LSEG), and transcripts from conference calls and meetings (e.g., Seeking Alpha and FOMC transcripts).[13]

3.2. Qualitative Inspections of TFs

We first show that TFs capture themes and topics more precisely than plain-vanilla LDA in Figures 3 and 4. Each word cloud corresponds to a TF generated from the WSJ data, and the font size of a word reflects its relative frequency within the cluster. Each word cloud in Figure 3 comprises words that intuitively go together to form a coherent topic, even when we allow the cluster size to be large. For comparison, in Figure 4, we display the top three topics by applying plain-vanilla LDA to the same data and fine-tuning them so that the topic size is the most conducive to interpretation. Nevertheless, we can see that common words tend to dominate each topic, which reduces the distinction and interpretation of different topics. In Appendix A, we further illustrate that our vector representation effectively captures the semantic closeness between words. Each subplot in Figure 6 shows the 20 words most closely associated with specific terms: "Crisis," "Inflation," "Unemployment," and "Computer." Our vector representation successfully maps semantically related words together in the 300-dimensional space. For example, the words most closely related to "Crisis" include "crisis deepens," "meltdown," "contagion," and "liquidity crunch."

[11] While a natural routine is to inspect and select topics/clusters from the ones with higher importance (singular value in the case of SVD), we remind the readers that we care about documents' loadings on certain topics as well as the overall importance of the topic. Therefore, it is better to prioritize clusters that have the highest average loadings across documents when it comes to factor/cluster/topic selection. Appendix G elaborates on this point.

[12] We focus on front-page articles only because these are manually edited and corrected for typos. This is particularly useful for newspaper articles from earlier years as they are digitized using OCR (optical character recognition), which inevitably generates typos.

[13] To be clear, there are many sources of news text data including the New York Times, the Financial Times, the Economists, Factiva, and firms' proxy statements or IPO prospectuses. Our goal is to take the most straightforward and available data sets to show that TFs are intuitive and functional without heavy tuning or data mining. What would be the best data source and econometric specification depends on the specific topic under examination.

Next, we examine TFs by plotting the time series of topic importance that are related to "computer," "war," and "recession" in Figure 5 as a validation test. These topics were selected for illustration purposes because they correspond to major themes of the last century - economic and financial crises and technological breakthroughs. Upon visual inspection, the variations in these times series are reassuring because the intensity of the TFs accurately captures the historical prominence of these topics.

Finally, TFs can effectively capture semantic similarities among words within each cluster, leading to a low-rank structure of topics within each cluster. Figure I.1 illustrates the low-rank structure of topic importance associated with the first and second eigenvalues within each cluster. Figure I.2 shows that topic importance is highly left-skewed, suggesting that a limited number of topics are prominent in the WSJ data.

3.3. Computational Efficiency

In practice, computational speed is important for real-time news analysis and index updating with textual information. We report in Table 1 the amount of time required to run Plain-vanilla Latent Dirichlet Allocation (PV-LDA) and TF Model using Singular Value Decomposition (TF-SVD), given different numbers of topics/clusters and different numbers of words in the cluster. We use company filing data from SEC EDGAR for illustration. Note that topic size is a parameter that specifies a maximum number of words in a topic while generating topics from vectorized words. Clustering in high dimensions with a large number of observations takes days to run even on distributed computing grids. In comparison, TFs are much faster (see Table 1).

4. Applications in Economics and Finance

We describe three broad types of TF applications in economics and finance: (1) forecasting and inference, (2) interpreting non-text-based models, and (3) constructing variables that are either unavailable or hard to capture using structured data. Our goal is to convey the strength of TFs relative to other approaches and open various possibilities for future research.[14]

4.1. Forecasting Macroeconomic Outcomes from News

Many macroeconomic variables are measured at low frequency, released with lags, and hard to capture even with surveys, calling for forecasting or "nowcasting" using texts (Gentzkow et al. 2019, Tetlock 2007, Baker et al. 2016, Kelly et al. 2021a). Forecasting macroeconomic variables, including CPI, GDP, housing price index, S&P 500 returns, unemployment, and private nonresidential fixed investment with WSJ data therefore constitutes a good application of TF.

[14] To construct the TFs, we extend beyond single words to include N-grams to capture more complex semantic units and relationships within the text. We first perform lemmatization and stemming using the Gensim package to standardize irregular expressions, before we generate embeddings for unigrams and bigrams.

We apply TF to the titles and abstracts of front-page articles and treat TF loadings as forecasters. Table 2 reports the ratio between the out-of-sample root mean square errors (RMSEs) for TF and for a benchmark model using one-hot representation.[15] A ratio below 1 indicates TF outperformance over benchmark models, with smaller ratios reflecting greater outperformance.

When estimating the models, we aggregate text data at a quarterly frequency and use the first 80% of the time series as the training sample. We then evaluate the goodness of fit using the remaining 20% as a test sample. Varying the number of TFs, our methodology reduces the out-of-sample RMSE by an average of 14% compared to the one-hot representation.[16] The magnitude is similar to that in Kelly et al. (2021a), despite that our data only contain titles and abstracts, not full texts. In Appendix I.2, we present out-of-sample performance against alternative models.

Note that TF construction is unsupervised and independent of downstream tasks. This creates a bias-variance trade-off when including more factors in downstream tasks. Including more TFs increases the likelihood of adding variables with predictive or explanatory power, but it also increases the risk of incorporating irrelevant variables, potentially worsening explanatory capability. We present various cases with different numbers of factor topics for illustration.

One main advantage of using TFs is their greater transparency and interpretability. In Tables 3 and Appendix I.3, we examine the predictive power of TFs on various macroeconomic variables. To demonstrate interpretability, we report the TFs with the largest coefficient magnitudes.[17] These tables show that the selected factors are intuitive and interpretable. For example, factors with the most predictive power for inflation include "inflation," "bank," and "natural disaster." Factors with the most predictive power for GDP growth are related to the financial system and policy, natural disasters, and future opportunities.

4.2. Interpreting Multi-Factor Asset Pricing and Black Box AI Models

Another application type involves interpreting non-text-based models and variables. We use TFs to interpret both time-varying risk premia and cross-sectional betas (βs) in the Fama-French-Carhart four-factor model, and then illustrate how TFs help interpret black box deep-learning models.

Time series of risk factor premia. We construct monthly TFs based on the aforementioned WSJ data and use Lasso to select TFs that can explain variations in risk premia, in an analogous manner to those in Section 4.1(Appendix J contains the details). Table J.1 shows that the most important factors explaining the market premium include topics related to "crisis," "disasters,"

[15] See Appendix H for the specification of one-hot representation.

[16] The TFs outperform one-hot representations in all but the housing price, likely because the titles and abstract of WSJ front-page articles contain little information about housing prices. This finding aligns with the results in Table I.5 as well.

[17] Because all factors are standardized, the magnitude also measures their importance in prediction.

"failures," and "collapse," and information related to commodities and pandemics. As another example, Table J.2 shows the TFs that explain the momentum premium. The most important factors are well captured by topics related to "rally, beat," "date, postpone," and "today, future, technologies," indicating that momentum could be driven by market sentiment, timing, or future expectations. These results provide insights into the sources of the momentum premium, which have been challenging to explain using numeric information.

Cross-sectional βs in FF. We next examine what TFs can reveal about firms' beta-loadings on the risk premia. For this exercise, WSJ data are insufficient as they do not contain texts relevant for cross-sectional analysis. Therefore, we use the risk factor disclosure sections of both the quarterly report (10-Q) and the annual report (10-K) to construct TFs. For the period 2005-2022, we aggregate the text data at the year and firm levels to examine whether TFs can help understand asset risk exposures. We report the Lasso-selected TFs, sorted by the magnitude of their coefficients in Tables J.4 - J.7. Table J.4 indicates that the most important TF explaining the market beta is related to forward-looking expectations, described by words such as "Future, Continue, Ongoing, Cease, Expand, Accelerate," which seem to be connected to the state evolution of the economy. See Appendix J for detailed TF construction and analyses.

Interpreting black box models. Cong et al. (2019) introduce Transformer-based deep reinforcement learning to investment and achieve great flexibility and performance of portfolio management. However, their AlphaPortfolio model involves millions of parameters and is hard to interpret. Applying TFs to firm 10K and 10Q filings to generate each firm's loadings on each TF, we can regress the "winner score," a metric determining the asset weights in an AlphaPortfolio construction, on the TF loadings in the cross-section and over time. The resulting coefficients inform us how exposures to each TF matter in determining the portfolio weights, i.e., what topics a firm is concerned with when it is traded by AlphaPortfolio.

The procedure and results are reported in Appendix K. We find that long positions in AlphaPortfolio typically mention issues such as loss-cutting, sales, and actions, as well as profitability, cash, and investment; the short positions are those heavily discussing real estate, corporate events, mistakes, uncertainty, and inventory. The patterns show certain consistency with the important features identified using the numerical polynomial sensitivity analysis in the paper.

4.3. Measuring Technology Breakthroughs by Theme and in Aggregate

The third type of application entails constructing variables and indices either unavailable or hard to capture using structured data. These variables can then be used as explanatory and dependent variables for downstream tasks. Examples include competition, tone, similarity, financial constraints, corporate governance, and innovation (e.g., Bodnaruk et al. 2015, Buehlmaier and Whited 2018,

Hoberg and Maksimovic 2015, Hoberg and Phillips 2010, Li et al. 2021). Existing constructions are often ad hoc, TFs provide a data-driven metric of technology evolution and breakthroughs.

Applying TFs to patent abstracts and claims for all patents issued from 1976 to 2023 by the USPTO, we derive (see Figure L.1) the dynamic loadings for six of the 20 most important topics: Machine Learning, IP Address and Internet, Carbon Nanotubes, Social Networks, Nucleic Acids, and Quantum Dots. The rising loadings coincide with the emergence of new technologies. For instance, the loadings on Machine Learning increased sharply around 2016, while that on Carbon Nanotube peaked around 2000, consistent with the timing of the technological breakthroughs.

To measure the significance of patents/innovations, we follow Kelly et al. (2021b), but replace the document-word matrix with the document-topic loading matrix, where a topic represents a TF. This approach allows us to bundle semantically similar words and construct a time series in Figure L.2 to demonstrate the dynamics of innovation significance at the monthly frequency for specific innovation themes or in aggregate. The constructed index reveals insights into the evolution of significant technological breakthroughs at the level of endogenously emergent themes corresponding to the TFs. For example, the sharp increase around 1984 corresponds to the emergence of Monoclonal Antibody technology, while the spike around 1998 aligns with the rise of internet technology.

As noted in Kelly et al. (2021b), constructing such an index is computationally intensive because document similarity among all patent pairs needs to be calculated. Our TF framework can significantly speed up the computation. For example, when the cluster size is 50, our framework can reduce runtime by approximately 1000 times.

5. Conclusion

Conceptually, we propose combining the strengths of representation learning and statistical topic modeling for analyzing large-scale text-based unstructured data. Methodologically, we introduce a Textual-Factor framework that integrates with structured regression analyses in social sciences. We do not target victory in a general horse race of text analytics—the specific data inputs and needs for interpretability/computational feasibility all play important roles in the choice of tools. TFs offer a novel and promising approach and complement existing approaches through jointly improved predictive and computation power, general applicability, and economic interpretability, as demonstrated through their applications in various economic settings. Finally, we share the source codes for other researchers to explore further applications.

References

Acikalin U, Caskurlu T, Hoberg G, Phillips GM (2022) Intellectual property protection lost and competition: An examination using large language models. *Tuck School of Business Working Paper* (4023622).

Andoni A, Indyk P, Laarhoven T, Razenshteyn I, Schmidt L (2015) Practical and optimal LSH for angular distance. *Advances in Neural Information Processing Systems*, 1225–1233.

Antweiler W, Frank MZ (2004) Is all that talk just noise? the information content of internet stock message boards. *Journal of finance* 59(3):1259–1294.

Baker SR, Bloom N, Davis SJ (2016) Measuring economic policy uncertainty. *Quarterly Journal of Economics* 131(4):1593–1636.

Bellstam G, Bhagat S, Cookson JA (2021) A text-based analysis of corporate innovation. *Management Science* 67(7):4004–4031.

Bengio Y, Ducharme R, Vincent P, Jauvin C (2003) A neural probabilistic language model. *Journal of machine learning research* 3(Feb):1137–1155.

Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. *Journal of machine Learning research* 3(Jan):993–1022.

Bloom N, Hassan TA, Kalyani A, Lerner J, Tahoun A (2020) The geography of new technologies. *Institute for New Economic Thinking Working Paper Series* (126).

Bloom N, Schankerman M, Van Reenen J (2013) Identifying technology spillovers and product market rivalry. *Econometrica* 81(4):1347–1393.

Blum A, Hopcroft J, Kannan R (2020) *Foundations of data science* (Cambridge University Press).

Bodnaruk A, Loughran T, McDonald B (2015) Using 10-k text to gauge financial constraints. *Journal of Financial and Quantitative Analysis* 50(4):623–646.

Brown SV, Tucker JW (2011) Large-sample evidence on firms year-over-year md&a modifications. *Journal of Accounting Research* 49(2):309–346.

Buehlmaier MM, Whited TM (2018) Are financial constraints priced? evidence from textual analysis. *Review of Financial Studies* 31(7):2693–2728.

Chen J, Tang G, Zhou G, Zhu W (2023) ChatGPT, stock market predictability and links to the macroeconomy. *Available at SSRN 4660148* .

Chen MA, Wu Q, Yang B (2019) How valuable is fintech innovation? *Review of Financial Studies* 32(5):2062–2106.

Chen Y, Kelly BT, Xiu D (2022) Expected returns and large language models. *Available at SSRN 4416687* .

Cherepanov V, Shi F, Zakolyukina A (2024) Fraud culture. *Working Paper* .

Cohen L, Malloy C, Nguyen Q (2020) Lazy prices. *Journal of Finance* 75(3):1371–1415.

Cong LW, Liang T, Yang B, Zhang X (2021) Analyzing textual information at scale. *Information for Efficient Decision Making: Big Data, Blockchain and Relevance*, 239–271 (World Scientific Publishing).

Cong LW, Tang K, Wang J, Zhang Y (2019) Alphaportfolio: Direct construction through reinforcement learning and interpretable ai. *Working Paper* .

Datar M, Immorlica N, Indyk P, Mirrokni VS (2004) Locality-sensitive hashing scheme based on p-stable distributions. *Proceedings of the twentieth annual symposium on Computational geometry*, 253–262 (ACM).

Engelberg JE, Parsons CA (2011) The causal impact of media in financial markets. *Journal of Finance* 66(1):67–97.

Evans JA, Aceves P (2016) Machine translation: mining text for social theory. *Annual Review of Sociology* 42.

Gentzkow M, Kelly B, Taddy M (2019) Text as data. *Journal of Economic Literature* 57(3):535–574.

Gentzkow M, Shapiro JM (2010) What drives media slant? evidence from us daily newspapers. *Econometrica* 78(1):35–71.

Grimmer J, Stewart BM (2013) Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political analysis* 21(3):267–297.

Hanley KW, Hoberg G (2010) The information content of ipo prospectuses. *Review of Financial Studies* 23(7):2821–2864.

Hanley KW, Hoberg G (2012) Litigation risk, strategic disclosure and the underpricing of initial public offerings. *Journal of Financial Economics* 103(2):235–254.

Hanley KW, Hoberg G (2019) Dynamic interpretation of emerging risks in the financial sector. *Review of Financial Studies* 32(12):4543–4603.

Hassan TA, Hollander S, Van Lent L, Tahoun A (2019) Firm-level political risk: Measurement and effects. *Quarterly Journal of Economics* 134(4):2135–2202.

Hoberg G, Knoblock C, Phillips G, Pujara J, Qiu Z, Raschid L (2024) Using representation learning and web text to identify competitor networks .

Hoberg G, Maksimovic V (2015) Redefining financial constraints: A text-based analysis. *Review of Financial Studies* 28(5):1312–1352.

Hoberg G, Phillips G (2010) Product market synergies and competition in mergers and acquisitions: A text-based analysis. *Review of Financial Studies* 23(10):3773–3811.

Hoberg G, Phillips G (2016) Text-based network industries and endogenous product differentiation. *Journal of Political Economy* 124(5):1423–1465.

Jaffe AB (1986) Technological opportunity and spillovers of r&d: evidence from firms' patents, profits and market value. *Working Paper* .

Jegadeesh N, Wu D (2013) Word power: A new approach for content analysis. *Journal of Financial Economics* 110(3):712–729.

Jegadeesh N, Wu DA (2017) Deciphering fedspeak: The information content of fomc meetings. *Working Paper* .

Kelly B, Manela A, Moreira A (2021a) Text selection. *Journal of Business & Economic Statistics* 39(4):859–879.

Kelly B, Papanikolaou D, Seru A, Taddy M (2021b) Measuring technological innovation over the long run. *American Economic Review: Insights* 3(3):303–320.

Kim A, Muhn M, Nikolaev VV (2024) Financial statement analysis with large language models. *Chicago Booth Research Paper Forthcoming, Fama-Miller Working Paper* .

Kogan L, Papanikolaou D, Schmidt LD, Seegmiller B (2023) Technology and labor displacement: Evidence from linking patents with worker-level data. Technical report, National Bureau of Economic Research.

Kogan L, Papanikolaou D, Seru A, Stoffman N (2017) Technological innovation, resource allocation, and growth. *Quarterly Journal of Economics* 132(2):665–712.

Lee CM, Sun ST, Wang R, Zhang R (2019) Technological links and predictable returns. *Journal of Financial Economics* 132(3):76–96.

Leskovec J, Rajaraman A, Ullman JD (2020) *Mining of massive data sets* (Cambridge university press).

Li K, Mai F, Shen R, Yan X (2021) Measuring corporate culture using machine learning. *The Review of Financial Studies* 34(7):3265–3315.

Lopez-Lira A, Tang Y (2023) Can ChatGPT forecast stock price movements? return predictability and large language models. *arXiv preprint arXiv:2304.07619* .

Loughran T, McDonald B (2013) Ipo first-day returns, offer price revisions, volatility, and form s-1 language. *Journal of Financial Economics* 109(2):307–326.

Manela A, Moreira A (2017) News implied volatility and disaster concerns. *Journal of Financial Economics* 123(1):137–162.

Mikolov T, Chen K, Corrado G, Dean J (2013a) Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* .

Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013b) Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 3111–3119.

Pennington J, Socher R, Manning C (2014) Glove: Global vectors for word representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* 20:53–65.

Schubert E (2023) Stop using the elbow criterion for k-means and how to choose the number of clusters instead. *ACM SIGKDD Explorations Newsletter* 25(1):36–42.

Schwenkler G, Zheng H (2020a) Competition or contagion? evidence from cryptocurrency peersv. *Working Paper* .

Schwenkler G, Zheng H (2020b) The network of firms implied by the news. *Working Paper* .

Schwenkler G, Zheng H (2024) Why does news coverage predict returns? evidence from the underlying editor preferences for risky stocks. *Available at SSRN 3977285* .

Sontag D, Roy D (2011) Complexity of inference in latent dirichlet allocation. *Advances in neural information processing systems* 24.

Tetlock PC (2007) Giving content to investor sentiment: The role of media in the stock market. *Journal of finance* 62(3):1139–1168.

Vergani AA, Binaghi E (2018) A soft davies-bouldin separation measure. *2018 IEEE international conference on fuzzy systems (FUZZ-IEEE)*, 1–8 (IEEE).

Wallach H, Mimno D, McCallum A (2009) Rethinking lda: Why priors matter. *Advances in neural information processing systems* 22.

Xu W, Kotecha MC, McAdams DA (2024) How good is ChatGPT? an exploratory study on ChatGPT's performance in engineering design tasks and subjective decision-making. *Proceedings of the Design Society* 4:2307–2316.

Zhang Y, Li Y, Cui L, Cai D, Liu L, Fu T, Huang X, Zhao E, Zhang Y, Chen Y, et al. (2023) Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219* .

Figures

Figure 1 Trade-offs of Different Approaches

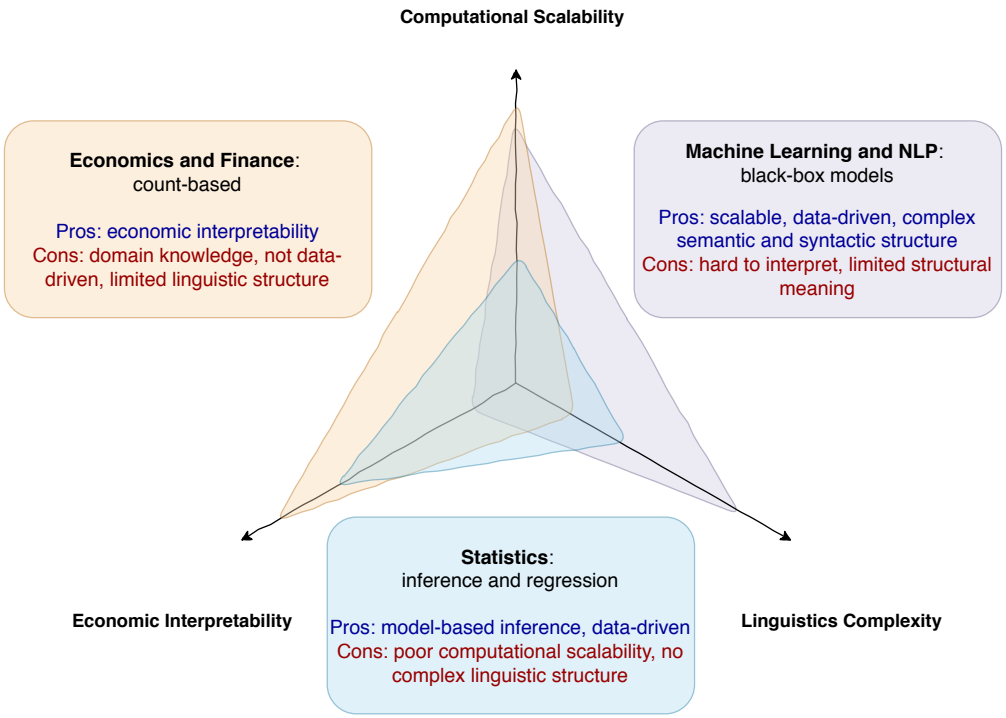


Figure 2 Word Embedding

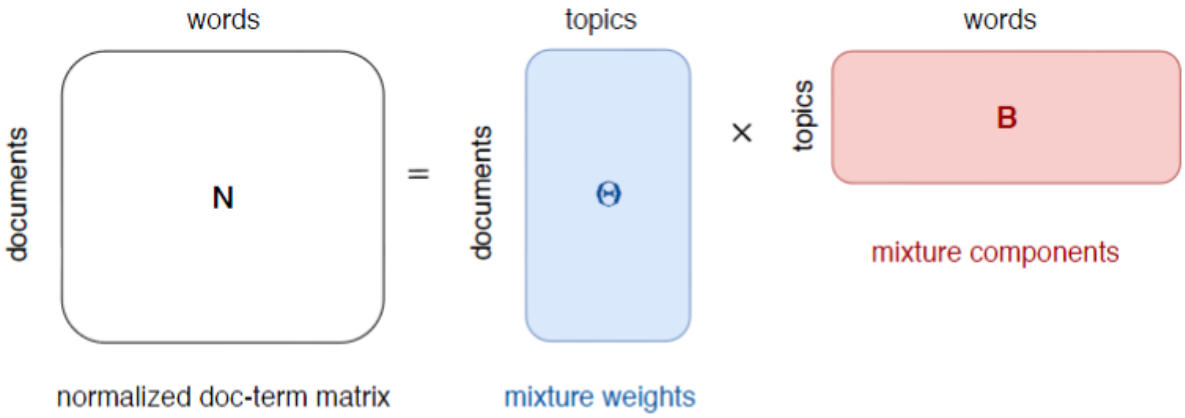
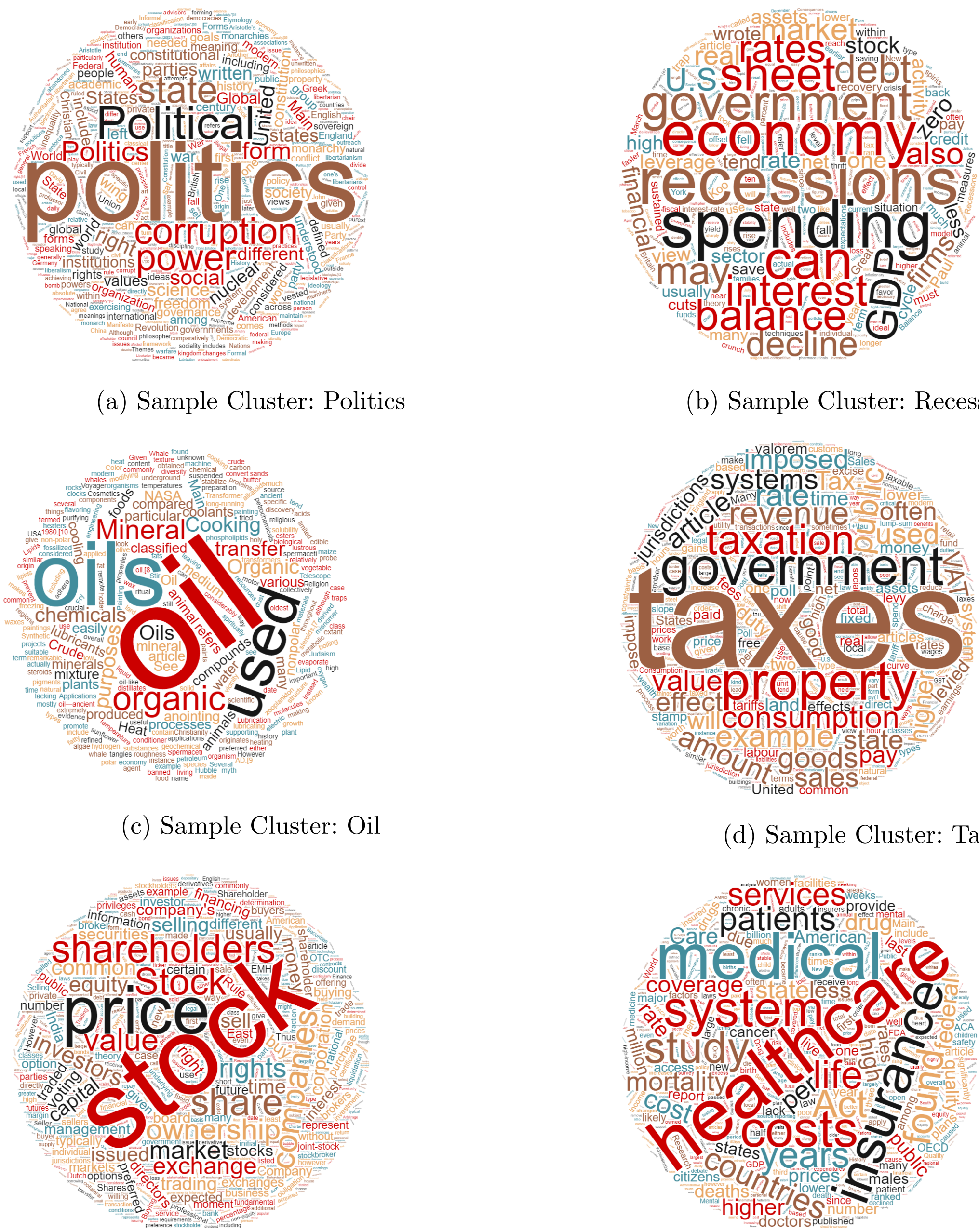


Figure 3 Sample Clusters Based on TFs



(a) Sample Cluster: Politics

(b) Sample Cluster: Recessions

(c) Sample Cluster: Oil

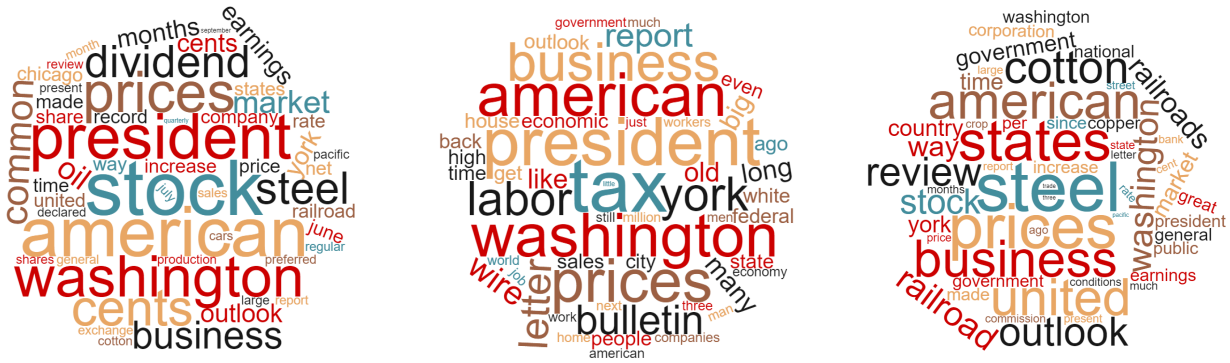
(d) Sample Cluster: Tax

(e) Sample Cluster: Stock

(f) Sample Cluster: Healthcare

Notes: This figure displays six clusters identified using vector embeddings of words extracted from the titles and abstracts of WSJ front-page articles. To construct these clusters, each word was transformed into a vector embedding using google Word2Vec, and subsequently grouped using LSH. The clusters selected for visualization focus on topics of particular relevance to academic researchers and policymakers, namely Politics, Recessions, Oil, Taxes, Stocks, and Healthcare. For each cluster, a document-word matrix was constructed using words from the respective cluster. The word loadings within each cluster were visualized using Latent Semantic Analysis (LSA) applied to the sub-matrices. This approach highlights the dominant themes and vocabulary specific to each topical cluster.

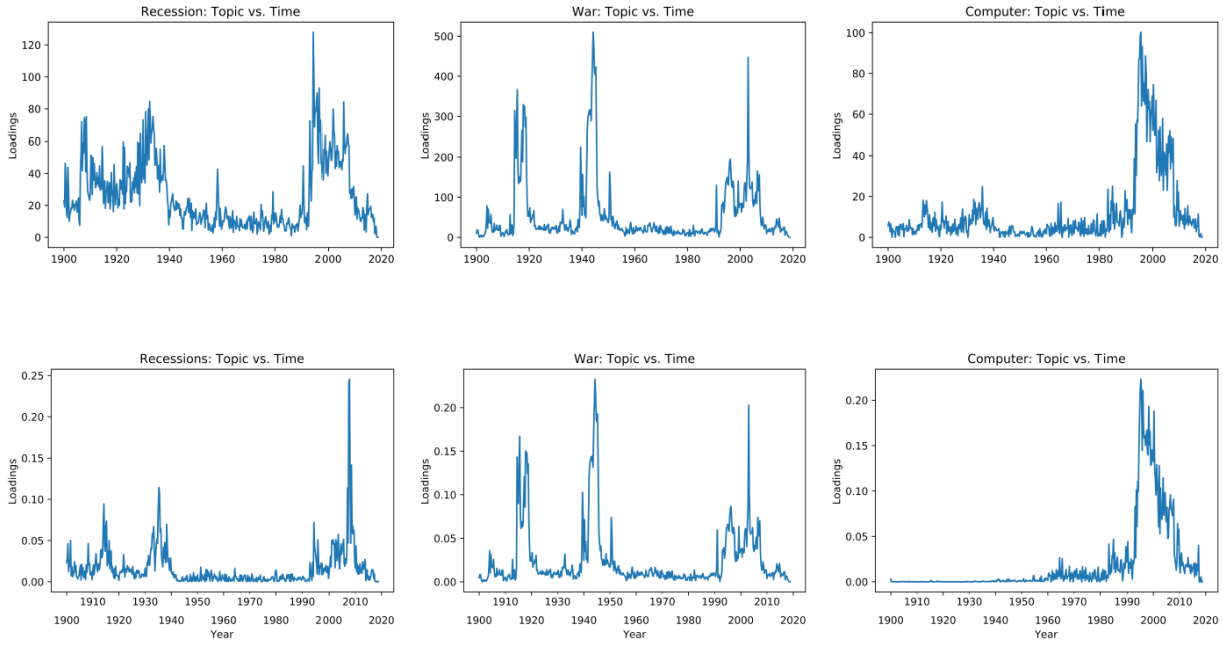
Figure 4 Sample Clusters Based on Plain-vanilla LDA



(a) Sample Cluster: Stock (b) Sample Cluster: Tax (c) Sample Cluster: Steel

Notes: This figure is constructed using clusters derived from a Latent Dirichlet Allocation (LDA) model, showcasing three topics that closely parallel those presented in Figure 3. Each topic is labeled according to its most frequent word to provide intuitive insights into its thematic focus. For instance, the first topic is labeled "Stock," reflecting that "stock" is the predominant term within this topic.

Figure 5 Importance of TFs Over Time



Notes: This figure illustrates the dynamic importance of specific TFs derived from the titles and abstracts of WSJ front-page articles, capturing the evolution of narrative emphasis over time on economic recessions, warfare, and technological advancements. The importance of each factor is calculated using the SVD approach as described in Section 2.3 (the first row) and frequency count approach as described in Section E (the second row). The three columns distinctly showcase TFs related to "Recession," "War," and "Computer." These topics were selected as they represent major themes of the last century - economic crises, wars, and technological breakthroughs - and exhibit significant temporal patterns.

Tables

Table 1 Computation Time for Textual Analysis Models

The table presents the computational efficiency of Plain-vanilla Latent Dirichlet Allocation (PV-LDA) which is the conventional LDA model and the TF Model using Singular Value Decomposition (TF-SVD) when applied to company filing data from SEC EDGAR, evaluating their performance across various topic configurations. It specifically examines execution time, system RAM, and CPU core utilization, comparing the models at comparable topic/cluster counts. For TF-SVD, 'cluster size' indicates the maximum number of words per cluster, whereas for PV-LDA, it represents the number of words in each topic. Note that cluster size is a parameter that specifies a maximum number of words in a cluster during the process of generating clusters from vectorized words for TF-SVD analysis. To construct the clusters based on TF-SVD, we first use "Sequential Word Clustering" to cluster semantic similar words by setting the cluster size, which is the maximum number of words in each cluster. Given the cluster size, the number of TF-SVD clusters will be determined. For comparisons of computation time and resources, we consider the LDA with a similar amount of topics.

Model	Cluster Size	Number of Topics	Time	System RAM	Core
PV-LDA	N.A.	1000	328.8 min	16 G	8
PV-LDA	N.A.	2293	16.31 hr	32 G	28
PV-LDA	N.A.	29523	$\geq$ 168 hr	32 G	28
TF-SVD	$\geq$ 2000	2293	12.03 min	16 G	8
TF-SVD	200	29370	312.58 min	16 G	8
TF-SVD	100	29653	147.75 min	16 G	8
TF-SVD	50	29693	74.23 min	16 G	8
TF-SVD	20	29523	29.81 min	16 G	8

Table 2 Forecast Improvements on Macroeconomic Outcomes: One-hot Benchmark

This table presents the out-of-sample Root Mean Squared Error (RMSE) for Support Vector Machine (SVM) and K-nearest Neighbors (KNN) models, which utilize TFs as opposed to a one-hot encoding of the titles and abstracts of WSJ front-page articles, covering the period from 1889 to 2018. TFs were derived by clustering word embeddings, with each cluster constrained to a maximum of 50 words. Each cell within the table displays the ratio

$$\frac{RMSE_{\text{TFs}}}{RMSE_{\text{one-hot}}}$$

This comparative study includes predictions based on the 20, 50, 100, 200, 500, and 1000 most important factors, with factor importance defined according to the SVD approach. Thus, the topic's importance is independent of downstream prediction performance. Historical economic data series such as the S&P 500 returns (starting in 1889), GDP growth, private nonresidential fixed investment growth (Fixed Investment), and the Consumer Price Index growth (CPI) all commenced in 1947, while the unemployment rate series began in 1948, and the housing price growth started in 1953. To address over-fitting, the initial 80% of observations were used to estimate the model, employing five-fold cross-validation to optimize hyperparameters for both models using one-hot representation and TFs, and the subsequent 20% of the time series was used to assess out-of-sample performance. Instances where the table displays 'NA' indicates non-convergence of the algorithm, due to an excessive number of predictors or issues with collinearity.

	# of TFs = 20		# of TFs = 50	
	SVM	KNN	SVM	KNN
CPI	0.944	0.761	0.961	0.719
GDP	0.799	1.013	0.842	0.889
Housing Price	0.919	1.290	0.955	1.268
S&P 500	0.216	0.758	0.226	0.783
Unemployment	0.821	0.905	0.807	1.006
Fixed Investment	0.824	0.855	0.827	0.978
	# of TFs = 100		# of TFs = 200	
	SVM	KNN	SVM	KNN
CPI	0.975	0.855	0.989	0.735
GDP	0.859	0.931	0.865	0.890
Housing Price	1.000	1.239	0.997	1.230
S&P 500	0.229	0.882	0.227	0.770
Unemployment	0.861	0.973	0.761	0.934
Fixed Investment	0.832	0.814	0.851	0.838
	# of TFs = 500		# of TFs = 1000	
	SVM	KNN	SVM	KNN
CPI	NA	0.745	NA	0.766
GDP	NA	0.968	NA	0.930
Housing Price	NA	1.175	NA	1.237
S&P 500	NA	0.809	0.492	0.848
Unemployment	NA	0.925	NA	0.963
Fixed Investment	0.993	0.809	0.839	0.792

Table 3 TFs and the CPI Growth

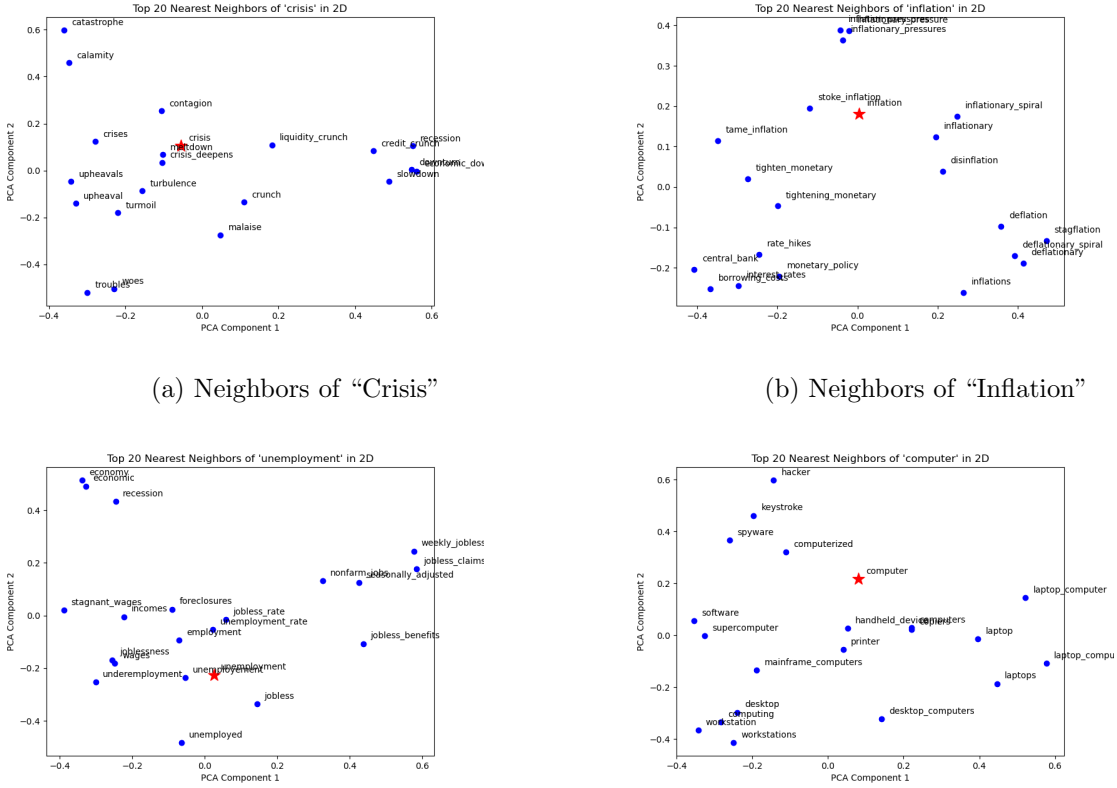
This table presents TFs that can explain next quarter's CPI growth. Hyperparameters were determined via five-fold cross-validation. Initially, the Lasso method was employed to select TFs with non-zero coefficients from the most important 500 TFs, followed by OLS regression to evaluate the predictability of these selected factors. Only TFs with the largest coefficient magnitude are presented in the table (at most 10 factors), along with their coefficients and standard errors. The adjusted R^2 from the OLS regression is 0.300. The 'Topic Distribution' column lists up to ten most relevant words and their loadings within each topic. Significance levels of the coefficients are indicated as: $***p < 0.01, **p < 0.05, *p < 0.10$.

Coefficient	Standard Error	Topic Distribution
0.049***	0.007	Inflation: 0.956, Economists: 0.280, Deflation: 0.054, Monetary_policy: 0.047, Inflationary: 0.047, Joblessness: 0.012, Inflation_pressures: 0.010, Inflationary_pressures: 0.008, Inflationary_pressure: 0.003, Inflationary_spiral: 0.002
-0.010**	0.005	Banks: 0.848, Banking: 0.347, Bankers: 0.259, Investors: 0.242, Brokers: 0.116, Managers: 0.101, Regulators: 0.051, Banker: 0.051, Politicians: 0.043, Lending: 0.042
-0.007***	0.002	Weather: 0.999, Winds: 0.026, Snow: 0.020, Storm: 0.018, Storms: 0.006, Heavy_rain: 0.002, Snowfall: 0.001, Northeasterly_winds: 0.001, Unseasonable: 0.001, Stormy_weather: 0.001
-0.005**	0.002	Business: 0.572, Company: 0.498, Market: 0.491, Sales: 0.275, Industry: 0.202, Financial: 0.141, Operations: 0.118, Economic: 0.105, Corporation: 0.084, Corporate: 0.061
-0.004	0.003	Government: 0.577, Companies: 0.484, Federal: 0.468, Officials: 0.384, Firms: 0.138, Services: 0.104, Groups: 0.098, Agency: 0.089, Programs: 0.067, Businesses: 0.066
0.004*	0.002	President: 0.868, Chief: 0.225, Secretary: 0.193, Minister: 0.161, Policy: 0.154, Executive: 0.152, Commerce: 0.144, Finance: 0.137, Director: 0.115, Interests: 0.092
-0.004**	0.002	US: 1.0, Opportunity: 0.011, Opportunities: 0.005, Incentive: 0.002, Privilege: 0.001, Untapped: 0.001, Dimension: 0.0, Excuse: 0.0, Opportune: 0.0, Springboard: 0.0
0.005*	0.003	Washington: 0.904, American: 0.359, British: 0.100, Texas: 0.086, German: 0.068, America: 0.066, Germany: 0.051, Russian: 0.051, Canadian: 0.051, Illinois: 0.050

Appendix

A. Power of Word Embedding

Figure 6 Embedding Based Word Neighbors



(a) Neighbors of "Crisis"

(b) Neighbors of "Inflation"

(c) Neighbors of "Unemployment"

(d) Neighbors of "Computer"

Notes: This figure demonstrates how word embeddings effectively capture the semantic relationships between words. To construct the figure, we first create word embeddings for unigrams and bigrams from the titles and abstracts of Wall Street Journal front-page articles. We then select four words: "Crisis," "Inflation," "Unemployment," and "Computer" and identify the 20 closest words to each in the embedding space. These high-dimensional vectors are projected into a two-dimensional space using the first and second principal components. The figure clearly shows that semantically related words cluster together in the vector space, illustrating the capability of word embeddings to preserve semantic similarity. For instance, words closely related to "Crisis" include "crisis deepens," "meltdown," "contagion," and "liquidity crunch," all of which intuitively belong to the same semantic category.

Count-based and statistical models for textual analysis in social sciences often adopt the "one-hot" representation to treat words (or N-grams) as very high dimensional vectors/indices over a vocabulary with only one "1" and lots of "0"s, leaving out interdependence and semantic relations such as similarity among words. The representation is inherently sparse, high dimensional, and noisy.

To overcome this limitation, we take a one-hidden-layer neural network to learn representation (Word2Vec) proposed by Bengio et al. (2003) and Mikolov et al. (2013a). The concept of vector representation (embedding) plays a pivotal role in capturing the semantic and syntactic essence of words. Each word (or paragraph) is represented as a vector $w, \tilde{w} \in \mathbb{R}^{p \times V}$, where V represents the total vocabulary size of the document, and each column vector encapsulates the embedding for a specific word. These learned embeddings act as a form of data-derived domain knowledge, or priors, offering guidance to traditional statistical models that are more interpretable. This synergy not only reduces computational complexity but also empirically enhances model performance. The optimization goal, focused on maximizing the utility of local context windows, is formulated as:[18]

$$\min_{w,\tilde{w}} \; - \sum_{\substack{i \in \text{corpus} \\ j \in \text{context}(i)}} \left\{ \langle w_i, \tilde{w}_j \rangle - \log \left(\sum_{k \in V} \exp \left(\langle w_i, \tilde{w}_k \rangle \right) \right) \right\}.$$

Here, $w_i (\tilde{w}_i)$ denotes the vector representation of word i as the center (context) word, respectively. These representations can be combined into a single vector that represents word i. An alternative approach, inspired by the representation learning from global word co-occurrence matrices as suggested by Pennington et al. (2014), leverages a pre-defined weighting function $f(\cdot)$, for instance, $f(x) = (x/x_{\max})^{3/4} \wedge 1$. This method optimizes the weighted least squares to learn $w, \tilde{w} \in \mathbb{R}^{p \times V}$:

$$\min_{w,\tilde{w}} \sum_{i,j \in V} f\left(X_{ij}\right) \left(\langle w_i, \tilde{w}_j \rangle - \log X_{ij} \right)^2.$$

Here, X_{ij} measures the concurrence of i and j across documents. The adoption of word embeddings not only provides computational benefits but also syntactic and semantic advantages over traditional index-based and count-based methods, such as those discussed by Mikolov et al. (2013a). Unlike sparse "one-hot" representations, vector-based embeddings are dense, occupying a significantly lower dimensionality (p) compared to the vocabulary size (V). This compact representation captures word similarities whereby similar words or those appearing in similar contexts are positioned closer in the vector space, and antonyms are distanced apart while preserving linguistic structure with reduced noise.[19]

B. LSH Algorithm

This section describes the details of the LSH algorithm. LSH uses hash functions to sort items into "buckets" based on their similarity to tackle the computational challenge. We choose a family of hash functions with a

[18] "context" pertains to the surrounding words or phrases near a target word within a sentence, which helps in understanding the target word's meaning and usage

[19] Word embedding is widely applied in fields such as automatic summarization, machine translation, named entity resolution, sentiment analysis, information retrieval, speech recognition, and question answering.

special property of separability. Specifically, for any chosen hash function $h(\cdot)$ from a set H, the properties are:

1. *When Items Are Close.* If the distance $d(x, y)$ between any two items x and y is less than or equal to a threshold d_1, then $h(x) = h(y)$ with a probability of at least $1 - p_1$. This means there is a high chance (more than $1 - p_1$) they will end up in the same bucket.

2. *When Items Are Far Apart.* If $d(x, y)$ is greater than or equal to a larger threshold d_2, then $h(x) = h(y)$ with a probability of at most p_2. This means it is unlikely (less than p_2) they will be placed in the same bucket.

The probability concerns selecting hash functions from H. The computational complexity is "near-linear time," represented as $O(VN)$, where V is the data volume and N is the number of hash functions used to generate the hash table. p_1 and p_2 depend on the properties of the hash function and its sampling process.

Essentially, LSH requires balancing two key factors: the number of hash functions and the number of hash tables used. The more hash functions are used within a table, the finer the sorting but also the higher the risk of mistakenly separating items that are actually similar (think of being too specific in sorting books by genre, author, and publication date, you might end up isolating books that are broadly in the same category). On the flip side, using more hash tables increases the chances of correctly matching similar items, even if they were mistakenly sorted into different buckets initially since you will search for all buckets where the query vector is.

This balance is crucial because it affects the accuracy of identifying the "nearest neighbor". To minimize errors (both missing a similar item or wrongly identifying an item as similar), LSH uses several hash tables. For instance, employing 32 hash tables means we check 32 different buckets to find the best match for a query vector, significantly reducing the chance of mistakes. The trade-off is a balancing act: more hash tables and carefully chosen hash functions improve the chance of finding the true nearest neighbor but require more computing resources. The setup can be fine-tuned to achieve desired accuracy levels, often determined through pre-training processes that adjust the number and configurations of hash functions and tables to ensure efficient and accurate sorting of items.

By understanding these trade-offs, we can appreciate how LSH offers a practical solution for quickly finding similar items in vast datasets, crucial for applications like recommendation systems, image recognition, and beyond (see, Leskovec et al. (2020)).

To give a concrete example of clustering similar points into a single cluster, particularly in the context of semantic vector-space models, we utilize cosine similarity as a key metric. This metric is defined as

$$d(w_i, w_j) = \arccos\left(\frac{\langle w_i, w_j \rangle}{\|w_i\| \cdot \|w_j\|}\right),$$

effectively measuring the cosine of the angle between two vectors, w_i and w_j, to assess their similarity. To efficiently identify clusters of similar vectors, we employ LSH with a strategy based on random hyperplanes, using the hash function:

$$h_v(w) = \operatorname{sgn}\left(\langle v, w \rangle\right),$$

where v is a vector randomly sampled from the unit sphere S^{p-1}.

The essence of our approach lies in the generation and composition of multiple hash functions h_{v_k}, where each v_k represents an independent random direction selected from $[N] = \{1, 2, ..., N\}$. For each direction, v_k, the corresponding hash function projects a word vector w onto v_k and assigns it to one of two groups based on the sign of the projection. This binary classification is determined by whether the dot product $\langle v_k, w \rangle$ is positive or negative.

By composing these hash functions (N random hash functions), we create a multi-dimensional hashing space, where each dimension corresponds to the outcome from a distinct h_{v_k}. All points fall into one of 2^N buckets, and points within the same bucket are close enough with high probability. This multidimensional approach allows us to finely discriminate between word vectors based on their orientation and position relative to each v_k, effectively clustering them based on nuanced similarities. The composition of hash functions thus encodes the high-dimensional semantic space into a more manageable form, enabling the efficient identification of "near neighbors" vectors that are similar to a given query vector w_i. This method's strength lies in its ability to conduct similarity searches within a high-dimensional space in a computationally efficient manner, leveraging the geometric properties of vectors and the hashing space. Through this process, we achieve computational time that scales linearly with the size of the vocabulary V, making it highly effective for processing large datasets and clustering similar points into cohesive groups based on semantic similarity. To enhance accuracy in identifying the nearest neighbor and minimize the risk of missing it during the search, employing multiple hash tables, such as 32, proves effective. By searching across the collective buckets of these 32 hash tables, we significantly reduce the likelihood of overlooking the nearest neighbor.

C. LSH Clustering Algorithms

Algorithm 1: Hierarchical Word Clustering based on LSH

Output: K word clusters
Input : number of clusters K; a subroutine LSH algorithm that returns word pair candidates that are sufficiently similar, denoted by *lsh-cand-pairs*; a subroutine center-finding algorithm that returns a representative point of the cluster, denoted by *cluster-roid*.
Initialization: $numClusters = V \gg K$, and each cluster to be a word embedding vector;
while $numClusters > K$ **do**
  1. Run *lsh-cand-pairs* on all current clusters;
  (Optionally) calculate the cosine similarity over all candidate pairs to pick the most similar candidate pair;
  2. Pick one (best) candidate pair to merge, and combine the corresponding two clusters into one;
  3. Run *cluster-roid* to find the center of the new cluster then set $numClusters = numClusters - 1$;
end

Algorithm 2: Sequential Word Clustering based on LSH

Output: Word clusters
Input : a subroutine LSH algorithm that returns approximate near neighbors of a query point, denoted by *lsh-near-neigh*.
Initialization: each point to be a word embedding vector, and a sequence of points (ordered) to be considered *pointsNotVisited*;
while *pointsNotVisited* $\neq \emptyset$ **do**
1. Take *queryPoint* to be the head of the *pointsNotVisited*;
2. Run *lsh-near-neigh* based on *queryPoint*, save the new word cluster to be *lsh-near-neigh* $\cap$ *pointsNotVisited*;
3. Take out *lsh-near-neigh* from *pointsNotVisited* ;

end

When selecting the optimal number of clusters (K) for practical applications of the LSH algorithm, it is crucial to adopt a systematic approach. There are several strategies researchers can consider. First, researchers can approach the selection of K as an unsupervised learning task, utilizing established clustering criteria to identify the most appropriate number of clusters. Methods such as silhouette scores (Rousseeuw 1987), the elbow method, or the Davies-Bouldin index Vergani and Binaghi (2018), Schubert (2023) offer quantitative measures to guide the choice. These techniques evaluate clustering performance and provide insights into the optimal K by balancing cluster cohesion and separation. Second, selecting K can also be viewed as a hyperparameter optimization problem within the broader context of the research project. Researchers can view K as a hyperparameter and experiment with different values of K to find the one that optimizes out-of-sample performance and improves the interpretability of the results. This iterative approach allows for the refinement of K based on empirical evidence and practical considerations, ensuring that the chosen value contributes positively to the research outcomes. Finally, employing Lasso regression can be an effective strategy in scenarios where the chosen K leads to many clusters, which might complicate interpretability. Lasso regression applies shrinkage to less relevant topics or clusters, helping to focus on the most significant clusters. This technique, which we utilized in our empirical work, enhances overall interpretability by prioritizing the most meaningful clusters and reducing the noise from less important ones. In our empirical analysis, we set the maximum number of words in the cluster to less than 50, which implicitly determines the number of clusters. By setting a maximum number of words per cluster and using Lasso to shrink irrelevant clusters, we can ensure a more focused and interpretable set of clusters, facilitating better insights and practical applications.

D. LSA and LDA Topic Models

For the LSA, the mixture weight matrix is the product of the matrix containing all left-singular vectors and the matrix of singular values. Conceptually, the expected frequency of a certain word w in document d, denoted as $\frac{\mathbf{N}_{dw}}{\sum_{w' \in [V]} \mathbf{N}_{dw'}}$, takes the factorization form of $\sum_{t \in \text{Topics}} \Theta_{dt} B_{tw}$.

$$\mathbb{E}\left[\frac{\mathbf{N}_{dw}}{\sum_{w' \in [V]} \mathbf{N}_{dw'}} | \Theta, B\right] = [\Theta B]_{dw} \tag{2}$$

LSA is guaranteed to converge when the topics are relatively separable with little overlapping vocabulary, which occurs rarely in prior applications, making the approach computationally infeasible. Θ represents the document-topic matrix, where each entry Θ_{dt} signifies the contribution or weight of topic t to document d. On the other hand, B denotes the topic-word matrix, with each entry B_{tw} indicating the weight or contribution of word w to topic t.

Rather than modeling the expected frequency, the LDA considers the full posterior of (Θ, B) given the doc-term matrix N, where $P(N_{d\cdot}|\Theta, B)$ is modeled as a multinomial distribution. Usually, the priors $P(\Theta)$ and $P(B)$ are chosen as the conjugate Dirichlet prior. The computation typically entails MCMC/variational methods and has slow or non-convergence. Moreover, common words tend to dominate all topics and there is very limited "separability."

$$P(\Theta, B|\mathbf{N}) \propto P(\mathbf{N}|\Theta, B)P(\Theta)P(B) = \prod_{d\in[D]} \left(\mathbf{N}_d! \prod_{w\in[V]} \frac{([\Theta B]_{dw})^{\mathbf{N}_{dw}}}{\mathbf{N}_{dw}!} \right) P(\Theta)P(B) \tag{3}$$

E. Topic Factors via Word Frequency Count

An alternative method of constructing TFs is to simply use the word frequency distribution within the clusters.

Frequency Count Approach - Topic Factor via Frequency Count. Consider the document-term submatrix $N_{S_i} \in \mathbb{R}^{D\times|S_i|}$. We denote the factor by $F_i \in \mathbb{R}^{|S_i|}$, which satisfies $\|F_i\|_{\ell_1} = 1$, and the topic importance by $d_i \in \mathbb{R}$. The factor is defined as the associated word distribution $F_i = \frac{1}{\mathbf{1}^T N_{S_i} \mathbf{1}} N_{S_i}^T \mathbf{1}$, and the topic importance is given by $d_i = \frac{\langle \mathbf{1}^T N_{S_i}, F_i \rangle}{\langle F_i, F_i \rangle}$.

The two approaches use different matrix factorizations, with the first approach directly employing Singular Value Decomposition (SVD), which is more effective but requires additional computation time. With the exception of Figure 5, throughout the paper, we use SVD to construct the TFs. These methods determine the relative importance of each word for a given topic and evaluate the significance of the specific topic. SVD places more weight on words frequently used within the topic rather than the frequency count approach.

F. Sample WSJ Titles and Abstracts of Front-Page Articles

Nations Ready Big Changes To Global Economic Policy --- Rush to Set a Plan for Growth Ahead of G-20 Summit, but Enforcement Issues Loom

THE WALL STREET JOURNAL.

Davis, Bob; Fidler, Stephen.
Wall Street Journal, Eastern edition; New York, N.Y. [New York, N.Y]21 Sep 2009: A.1.

Full text | Abstract/Details

Abstract Translate

If implemented, the framework would involve measures such as the U.S. saving more and cutting its budget deficit, China relying less on exports, and Europe making structural changes to boost business investment. The U.S. helped bring along the Chinese by endorsing Beijing's view that developing countries deserve a bigger stake in international institutions such as the International Monetary Fund.

A Florida Award for Moral Courage Spurs an Ignominious Retreat --- No Prize This Year After Namesake Accused Of Tax Fraud; Sinkhole Hero Returns Obelisk

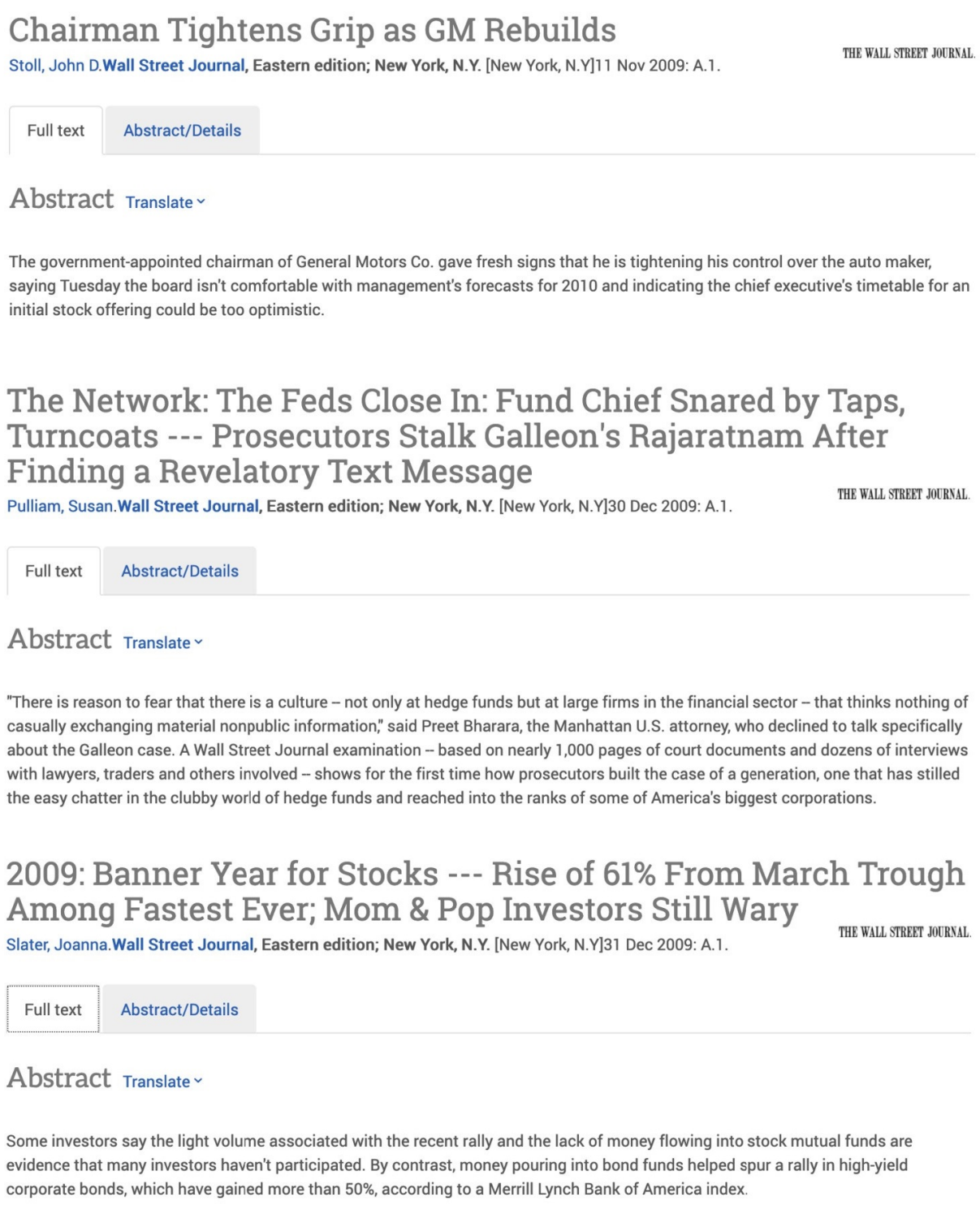
THE WALL STREET JOURNAL.

Newman, Barry.**Wall Street Journal, Eastern edition; New York, N.Y.** [New York, N.Y]01 Oct 2009: A.1.

Full text | Abstract/Details

Abstract Translate

A Florida Award for Moral Courage Spurs an Ignominious Retreat --- No Prize This Year After Namesake Accused Of Tax Fraud; Sinkhole Hero Returns Obelisk Hillsborough's homage to moral courage dates to a dark day 26 years ago when three commissioners were marched out of their offices in handcuffs, and ultimately were jailed for extorting money from people seeking zoning favors.



Chairman Tightens Grip as GM Rebuilds

THE WALL STREET JOURNAL.

Stoll, John D.**Wall Street Journal, Eastern edition; New York, N.Y.** [New York, N.Y]11 Nov 2009: A.1.

Full text | Abstract/Details

Abstract Translate

The government-appointed chairman of General Motors Co. gave fresh signs that he is tightening his control over the auto maker, saying Tuesday the board isn't comfortable with management's forecasts for 2010 and indicating the chief executive's timetable for an initial stock offering could be too optimistic.

The Network: The Feds Close In: Fund Chief Snared by Taps, Turncoats --- Prosecutors Stalk Galleon's Rajaratnam After Finding a Revelatory Text Message

THE WALL STREET JOURNAL.

Pulliam, Susan.**Wall Street Journal, Eastern edition; New York, N.Y.** [New York, N.Y]30 Dec 2009: A.1.

Full text | Abstract/Details

Abstract Translate

"There is reason to fear that there is a culture -- not only at hedge funds but at large firms in the financial sector -- that thinks nothing of casually exchanging material nonpublic information," said Preet Bharara, the Manhattan U.S. attorney, who declined to talk specifically about the Galleon case. A Wall Street Journal examination -- based on nearly 1,000 pages of court documents and dozens of interviews with lawyers, traders and others involved -- shows for the first time how prosecutors built the case of a generation, one that has stilled the easy chatter in the clubby world of hedge funds and reached into the ranks of some of America's biggest corporations.

2009: Banner Year for Stocks --- Rise of 61% From March Trough Among Fastest Ever; Mom & Pop Investors Still Wary

THE WALL STREET JOURNAL.

Slater, Joanna.**Wall Street Journal, Eastern edition; New York, N.Y.** [New York, N.Y]31 Dec 2009: A.1.

Full text | Abstract/Details

Abstract Translate

Some investors say the light volume associated with the recent rally and the lack of money flowing into stock mutual funds are evidence that many investors haven't participated. By contrast, money pouring into bond funds helped spur a rally in high-yield corporate bonds, which have gained more than 50%, according to a Merrill Lynch Bank of America index.

G. SVD in Topic Models

In linear algebra, the singular value decomposition of an $m \times n$ matrix A is the factorization of the form UDV^T, where U and V have orthonormal columns and D is a diagonal matrix with positive real entries.

In the context of the textual analysis model, given a cluster S with support in the vocabulary dictionary, i.e., $S \subset V$, we consider the singular value decomposition of A^S, the document-term matrix associated with cluster S. $A^S \in \mathbb{R}^{D \times |S|}$, where D is the number of documents in the corpus, $|S|$ is the cardinality of the cluster, and the entry A^S_{ji} is the frequency of word i of cluster S in document j.

The goal of SVD can be intuitively understood as to find a best-fit k-dimensional subspace concerning the matrix's row vectors, or more explicitly, to maximize the sum of the squares of the lengths of these row vectors' projections onto the subspace. Thus, the singular value decomposition of matrix A can be derived from the following greedy algorithm:

Definition 1 *The first right singular vector v_1 of A^S is vector in $\mathbb{R}^{|S|}$ such that*

$$|A^S v_1| = \max_{|v|=1} |A^S v|,$$

and $\sigma_1(A^S) = |A^S v_1|$ is called the first singular value of A^S.

Definition 2 *Given v_1 of A^S, the i^{th} right singular vector v_i is then defined by*

$$|A^S v_i| = \max_{|v|=1, v \perp v_1, \ldots, v_{i-1}} |Av|,$$

and the process stops at v_{i-1} when $v_i = 0$. The i^{th} singular value is also defined similarly, with $\sigma_i(A^S) = |A^S v_i|$.

It can be shown that the subspace V_k spanned by $v_1, \ldots, v_k$ maximizes $\sum_{i=1}^k \sigma_i^2(A^S)$, which formalizes the claim that V_k is the best-fit k-dimensional subspace that maximizes the sum of the square of A's projection. Doing so is equivalent to minimizing the sum of squares of A's rows to the subspace, akin to least-square optimization.

Definition 3 *The i^{th} left singular vector u_i is defined as: $u_i = \frac{1}{\sigma_i(A^S)}(A^S v_i)$.*

Let A^S be an $D \times |S|$ matrix with right singular vectors $v_1, \ldots, v_k$, left singular vectors $u_1, \ldots, u_k$, and corresponding singular values $\sigma_1, \ldots, \sigma_k$. Then

$$A = \sum_{i=1}^k \sigma_i u_i v_i^T.$$

By the definition of left singular vectors, $Av_j = \sum_{i=1}^k \sigma_i u_i v_i^T v_j = Av_j \times 1 = Av_j$. Since any vector v can be expressed as a linear combination of the right singular vectors plus vector perpendicular to v_i, we have $Av = \sum_{i=1}^k \sigma_i u_i v_i^T v$ and $A = \sum_{i=1}^k \sigma_i u_i v_i^T$. Thus, the matrix U can be well found by $A^S(A^S)^T = UDU^T$ with U being the matrix consisting of the eigen-vectors of matrix $A^S(A^S)^T$.

Therefore, if we define U and V as matrices consisting of left and right singular vectors as column vectors respectively, and D as the diagonal matrix whose diagonal entries are the singular values, we have $A^S = UDV^T$.

Now in our textual analysis model, we consider the best-fit 1-dimensional subspace of $R^{|S|}$ with respect to rows of the document-term matrix, which means we focus on the first singular value and vector. In particular, let $v_1 \in \mathbb{R}^S$ be the first right singular vector and σ_1 be the first singular value associated with A^S.

We can view the document-term matrix A^S as D points in an $|S|$-dimensional space, where each point is a document's "actual loading" of cluster S (represented by the number of occurrences of every word in cluster S in that document). The first right singular vector v_1 that we derive from SVD expresses the probabilities of the words that correspond to cluster S, in a way that these probabilities would *best* account for the "actual loadings" of cluster S on each document that we observe through A^S. The first singular value σ_1 then plays the role of characterizing how "*best*" those probabilities reflect the observed textual data.

To put the above intuition in more concrete terms, note that $A^S v_1$ is really a list of lengths of the projections of each document's "actual loadings" of cluster S on v_1. $\sigma_1 = |A^S v_1|$ thus captures the magnitude of all documents' projection on topic S, (we use the word "topic" to refer to both the words in S and the distribution of those words), or the "component" of A^S that is in v_1. While v_1 is chosen to maximize such magnitude, different levels of σ_1 still reflect the difference between how much the corpus can be projected on different topics. In other words, the first singular value σ_1 indicates how "important" a topic S is to the observed textual data by evaluating the "component" of A^S on vector v_1.

In fact, for topic/cluster importance, we care about the document loadings on the topic/cluster in question as well as its overall importance (captured by σ_i) in the universe of texts. The document loadings can be read off as entries from $A^S v_i = \sigma_i(A^S) u_i$. Because the matrix containing u_i in SVD is unitary, taking the $\mathcal{L}^2$ norm of the ith column is equivalent to looking at σ_i. That said, in practice texts come with a lot of noise. Many documents do not load on a topic i but yet have non-zero entries, while others load fully on a topic i and are under-represented in the document-term matrix. In such situations, taking an $\mathcal{L}^1$ norm of $\sigma_i(A^S)u_i$ more accurately reflects how important the cluster/topic is. Therefore, we advocate that when deciding which topics or clusters to examine first, both approaches should be explored. Depending on the data set or specific application, the latter approach might help select clusters of words that are more meaningful, even though $\mathcal{L}^2$ norm is used for the greedy algorithm in the SVD itself (Blum et al. 2020).

We provide a numerical example. Suppose that we have a cluster S containing 5 words: {"predict," "predictions," "forecast," "report," "projection"}, and a corpus containing six documents. Then we define A to be the 6×5 document-term matrix for cluster S, where A_{ij} is the frequency of word j of the cluster in document i. Specifically, let

$$A = \begin{bmatrix} 2&1&0&0&3 \\ 0&2&0&2&1 \\ 4&3&3&2&1 \\ 0&2&5&1&0 \\ 0&0&1&3&0 \\ 4&0&2&1&1 \end{bmatrix}.$$

To perform SVD of A, we factorize matrix A into the form of UDV^T, such that U and V have orthonormal columns and D is a rectangular diagonal matrix with positive diagonal entries. Using the above document term matrix A, we have $A = UDV^T$, where

$$U = \begin{bmatrix} -0.24 & 0.54 & 0.15 & -0.51 & -0.56 & -0.25 \\ -0.191 & -0.04 & 0.71 & -0.22 & 0.16 & 0.62 \\ -0.68 & 0.13 & 0.03 & -0.01 & 0.60 & -0.39 \\ -0.46 & -0.70 & -0.29 & -0.33 & -0.30 & 0.10 \\ -0.18 & -0.25 & 0.52 & 0.59 & -0.40 & -0.35 \\ -0.44 & 0.37 & -0.35 & 0.48 & -0.21 & 0.52 \end{bmatrix}, D = \begin{bmatrix} 9.02 & 0 & 0 & 0 & 0 \\ 0 & 4.67 & 0 & 0 & 0 \\ 0 & 0 & 3.30 & 0 & 0 \\ 0 & 0 & 0 & 2.69 & 0 \\ 0 & 0 & 0 & 0 & 1.65 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, V^T = \begin{bmatrix} -0.55 & -0.40 & -0.60 & -0.35 & -0.23 \\ 0.65 & -0.12 & -0.57 & -0.19 & 0.44 \\ -0.30 & 0.33 & -0.47 & 0.72 & 0.25 \\ 0.33 & -0.61 & -0.04 & 0.54 & -0.47 \\ 0.27 & 0.59 & -0.310 & -0.12 & -0.68 \end{bmatrix}.$$

In our textual analysis model, the topic factor associated with the cluster is then represented by the top right singular vector v_1, which is the first column of matrix V. v_1 is a unit vector whose i^{th} entry (in absolute value) can be understood as the topic's loading on word i of cluster S. Thus, in the context of the example

above, for the topic associated with cluster S, the ratio of its loading on the word "predictions" to that on "forecast" is 0.4/0.6.

Moreover, since Av_1 is a vector whose i^{th} entry's absolute value is the length of the projection of document i's actual loadings (described by frequency of words in document-term matrix A) of cluster S on the topic factor v_1, we use vector Av_1 (with entries in absolute value) to describe documents' loadings on the topic, and its $l1$ norm $|Av_1|_{l1}$ to represent the topic's overall importance in the corpus.[20]

It should also be noted that the above SVD procedure is a procedure of decomposing a matrix into a weighted, ordered sum of separable matrices. By separable, we usually mean a matrix can be represented by an outer product to two vectors. Even though Theorem 1 says $A = \sum_k \sigma_k u_k v_k^T = \sum_k \sigma_k u_k \otimes v_k$, where σ_k is the k^{th} diagonal entry in D, and u_k is the k^{th} column of matrix U, the decomposition can be significantly sped up if we can separately perform SVD for smaller matrices and add them together. That is what "separability" buys us after we use LSH to cluster words.

H. Count-Based One-Hot Benchmark Model

Unlike our textual analysis model, one hot representation for textual data does not rely on the topic concept. Instead of a topic (a cluster of words), every word in the corpus vocabulary support becomes an explanatory variable. One hot representation then uses the frequency of these words in the corpus to model the relationship between the text and the dependent variable.[21]

In the application to forecast macroeconomic outcomes discussed in Section 4.1, the benchmark model uses a matrix whose entry $\mathbf{X}_{ij}^{oh}$ is simply the frequency of the j^{th} word in the i^{th} quarter's WSJ front-page title and abstract. $\mathbf{X}_{ij}^{oh}$ is very large, with the dimension equal to the number of quarters times the number of words in the entire corpus. $\mathbf{X}_{ij}^{oh}$ is sparse, with most entries in $\mathbf{X}_{ij}^{oh}$ being zero because most words appear very infrequently in the texts. This is why the representation is called "one hot" in the natural language process literature. In the application, for fair comparisons, we apply SVM and KNN to $\mathbf{X}_{ij}^{oh}$ and use them as our benchmark.

In the application to better understand factor pricing models in Section 4.2, the y variable for both our textual analysis model and one-hot representation is the monthly risk premium of a Fama-French factor. While in our textual analysis model, the $\mathbf{X}_{ij}$ entry in the regressor matrix is the i^{th} month's textual data's loadings on the j^{th} cluster derived from SVD, in the one hot representation, $\mathbf{X}_{ij}^{oh}$ is simply the frequency of the j^{th} word in the i^{th} month's textual data. As in our textual analysis model, we use Lasso regression to select variables to enhance the prediction accuracy. Specifically, we perform the following minimization:

$$\min_{\boldsymbol{\beta}} \frac{1}{2n} \|\mathbf{X}_{ij}^{oh}\boldsymbol{\beta} - \mathbf{y}\|_2^2 + \alpha\|\boldsymbol{\beta}\|_1,$$

[20] However, when we examine the importance of a particular time window, we advocate the use of the $l1$ norm divided by the number of documents. That would give us the average loading of the documents in that year on a particular TF. The number of documents each year varies over time, and this averaged value better captures the importance of the TF among a fixed set of TFs.

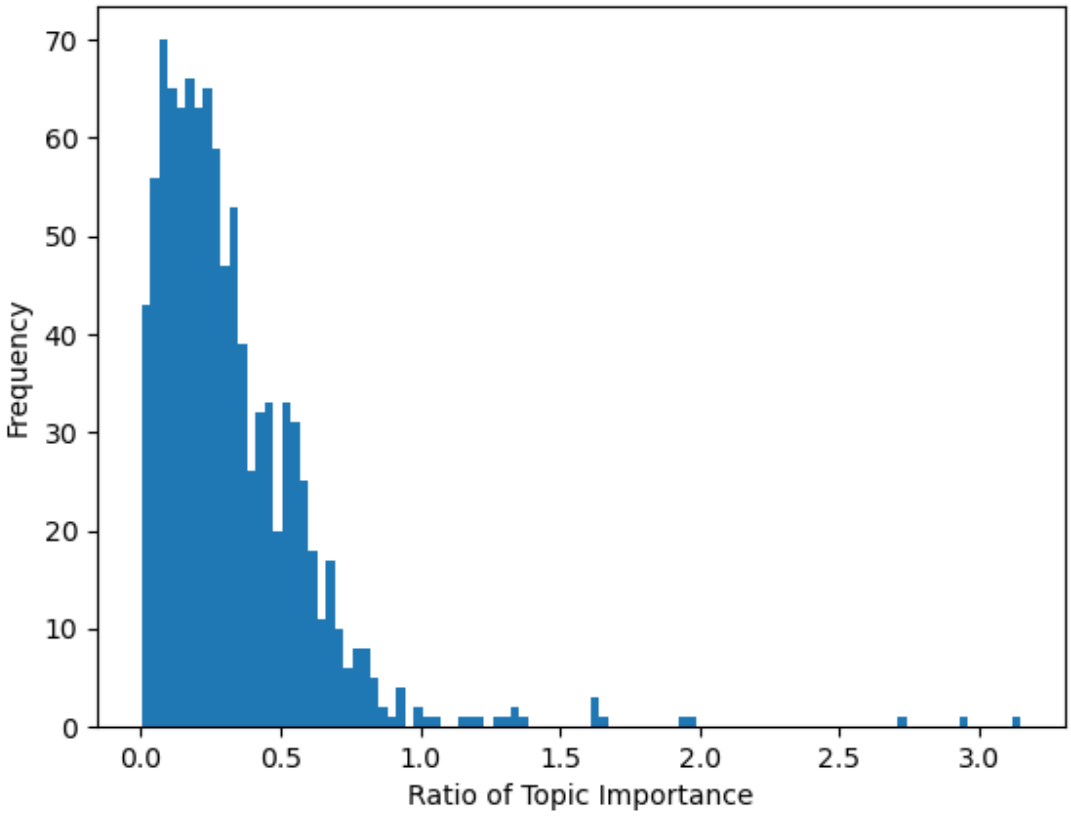
[21] Here, corpus refers to the collection of texts or documents we will analyze.

where n is the number of observations, $\mathbf{X}_{ij}^{oh} \in \mathbb{R}^{n \times |V|}$ is the one-hot regressor matrix with the same number of columns as vocabulary size $|V|$, and α is the same Lasso penalty coefficient as in our model. After the Lasso regression, we are then able to use the remaining one-hot x variables (single words) and their nonzero coefficients to predict out-of-sample data, and use the out-of-sample MSE to compare one hot representation with our textual analysis model.

I. Textual Factors and Macroeconomic Outcomes

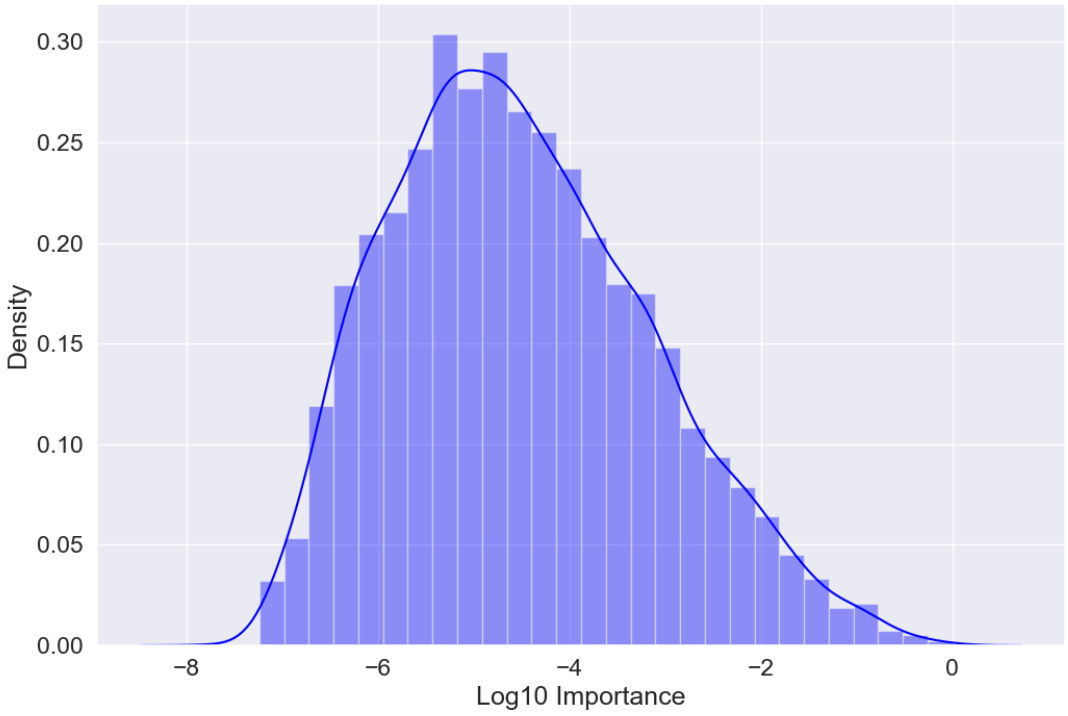
I.1. Topic Importance

Figure I.1 Histogram on the Ratio of the Second to the First Topic



This figure illustrates the low-rank structure of our topics within each cluster. Specifically, within each cluster, we employ Latent Semantic Analysis (LSA) to perform singular value decomposition (SVD), thereby identifying topics associated with the leading and second largest eigenvalue. We then calculate the importance of these two topics within each cluster and the ratio of their importance. The figure demonstrates that for most clusters, the topic importance associated with the leading eigenvalue is significantly greater than that associated with the second largest eigenvalue.

Figure I.2 Distribution of Topic Importance (Log10)



Notes: This figure presents the topic distribution, scaled using log10. We construct the monthly TFs using the titles and abstracts of WSJ front-page articles, covering the period from 1889 to 2018. The topic importance was calculated based on SVD as described in Section 2.3 and subsequently scaled by the maximum topic importance, resulting in values ranging from 0 to 1. This distribution is highly skewed, and the figure displays the log-scaled distribution.

I.2. Out-of-sample Performance Against Benchmark Models

In addition to using the one-hot representation as a benchmark model (Table 2), we examine the out-of-sample performance against alternative benchmark models in this section.

In Table I.1, we use plain-vanilla LDA with the number of topics as our TFs as an alternative benchmark model and report the ratio $\frac{RMSE_{textualfactors}}{RMSE_{LDA}}$, which demonstrates TFs' improvement over traditional topic modeling. In most cases, the ratio increased compared to those in Table 2 while most of them still remain below 1, indicating that much of the improvement from using TFs comes from incorporating word embedding into our methodology. Plain-vanilla LDA performs better than simple word counts but still misses out on text information on average.

We next show that our methodology improves an autoregressive benchmark using numerical data with one lag (AR(1) model), commonly used in macroeconomic forecasts and hard to beat empirically. We examine CPI growth for illustration. In Table I.2, we report the ratio $\frac{RMSE_{textualfactors+AR(1)}}{RMSE_{AR(1)model}}$ for CPI time series. We examine the performance of additional models, such as Ridge, Lasso, and Random Forest, and various numbers of TFs, all out-of-sample. In this exercise, we find that all ratios in Table I.2 are less than 1, meaning that using our framework and textual data improves predictive ability. While the magnitude of improvement in some of the above exercises is moderate, they are still better than many other textual analytics, and the Ridge model and KNN model with more TFs tend to perform particularly well.[22]

[22] The fact that our TF model improves over an AR(1) model is not trivial. Ultimately, whether text models beat the model that uses structured data depends on how much information is contained in the texts not captured by structured data. Because textual data are inherently noisy, it should not be taken for granted that adding texts to structured data always improves predictability. One needs effective tools to extract useful information.

Table I.1 Forecast Improvements on Macroeconomic Outcomes: LDA Benchmark

This table presents the out-of-sample Root Mean Squared Error (RMSE) for Support Vector Machine (SVM) and K-nearest Neighbors (KNN) models, comparing the performance of models utilizing TFs to those using conventional Latent Dirichlet Allocation (LDA) topic loadings. The analysis is based on the titles and abstracts of WSJ front-page articles, covering the period from 1889 to 2018. TFs were constructed by clustering word embeddings, limiting cluster size to 50 words each. The RMSE ratios reported in each cell are calculated as

$$\frac{RMSE_{\text{TFs}}}{RMSE_{\text{LDA}}}.$$

This comparative study includes predictions based on the 20, 50, 100, 200, 500, and 1000 most important factors, with factor importance given by the SVD. Thus, the topic's importance is independent of downstream prediction performance. Historical economic data series such as the S&P 500 returns (starting in 1889), GDP growth, private nonresidential fixed investment growth (Fixed Investment), and the Consumer Price Index growth (CPI) all commenced in 1947, while the unemployment rate series began in 1948, and the housing price growth started in 1953. To address over-fitting, the initial 80% of observations were used to estimate the model, employing five-fold cross-validation to optimize hyperparameters for both models using one-hot representation and TFs, and the subsequent 20% of the time series was used to assess out-of-sample performance.

	# of TFs = 20		# of TFs = 50	
	SVM	KNN	SVM	KNN
CPI	0.867	0.940	0.822	0.754
GDP	0.909	0.871	0.912	0.879
Housing Price	0.968	1.006	1.000	1.000
S&P 500	1.000	0.936	1.004	0.984
Unemployment	0.539	0.765	0.808	0.830
Fixed Investment	1.000	0.756	1.000	1.133
	# of TFs = 100		# of TFs = 200	
	SVM	KNN	SVM	KNN
CPI	0.814	0.742	0.957	0.818
GDP	0.888	1.014	1.123	0.931
Housing Price	1.005	0.849	0.982	0.937
S&P 500	0.997	1.024	1.070	0.923
Unemployment	0.848	1.092	1.142	1.047
Fixed Investment	0.999	1.030	1.000	1.135
	# of TFs = 500		# of TFs = 1000	
	SVM	KNN	SVM	KNN
CPI	1.006	0.846	1.113	0.858
GDP	1.115	0.872	1.091	0.847
Housing Price	0.978	0.821	0.890	0.967
S&P 500	1.063	1.045	1.055	0.904
Unemployment	0.995	1.046	0.983	1.033
Fixed Investment	0.999	0.794	0.999	0.773

Table I.2 Performance Relative to AR(1) model, CPI

This table compares the out-of-sample Root Mean Squared Error (RMSE) for various predictive models: Ridge, Lasso, Support Vector Machine (SVM), K-nearest Neighbors (KNN), and Random Forest. Each model integrates one-quarter lagged Consumer Price Index (CPI) and TFs derived from titles and abstracts of WSJ front-page articles, covering the period from 1889 to 2018. The baseline model is an AR(1) model utilizing one-quarter lagged CPI growth. The efficiency of TFs in improving model predictions is quantified through the ratio:

$$\frac{RMSE_{\text{TFs + AR(1)}}}{RMSE_{\text{AR(1) model}}}$$

for each modeling approach. The analysis includes predictions based on the 20, 50, 100, 200, 500, and 1000 most significant factors, where the importance of TFs is determined according to SVD approach. Thus, the topic's importance is independent of downstream prediction performance. The CPI growth data started in 1947. To mitigate overfitting, the initial 80% of observations were used to estimate the model, employing five-fold cross-validation to optimize hyperparameters for both models using one-hot representation and TFs, and the subsequent 20% of the time series was used to assess out-of-sample performance. Instances where the table displays 'NA' indicate non-convergence of the algorithm, due to an excessive number of predictors or issues with collinearity.

No. of TFs	Ridge	Lasso	KNN	SVM	Random Forest
20	0.693	0.878	0.801	0.705	0.781
50	0.552	0.869	0.765	0.829	0.776
100	0.553	0.884	0.722	0.994	0.764
200	NA	0.917	0.743	0.716	0.787
500	NA	0.885	0.761	0.788	0.784
1000	NA	0.871	0.741	0.797	0.783

I.3. Interpretable TFs - Additional Results

Table I.3 TFs and the GDP Growth

This table demonstrates the predictive power of TFs on next quarter's GDP growth using Lasso regression for variable selection. Hyperparameters were determined via five-fold cross-validation. Initially, the Lasso method was employed to select TFs with non-zero coefficients from the most important 500 TFs, followed by OLS regression to evaluate the predictability of these selected factors. Only TFs with the largest coefficient magnitudes are presented in the table (at most 10 factors), along with their coefficients and standard errors. The adjusted R^2 from the OLS regression is 0.135. The 'Topic Distribution' column lists up to ten of the most relevant words and their loadings within each topic. Significance levels of the coefficients are indicated as: $***p < 0.01, **p < 0.05, *p < 0.10$.

Coefficient	Standard Error	Topic Distribution
-0.063***	0.013	Banks: 0.848, Banking: 0.347, Bankers: 0.259, Investors: 0.242, Brokers: 0.116, Managers: 0.101, Regulators: 0.051, Banker: 0.051, Politicians: 0.043, Lending: 0.042
-0.030***	0.008	Weather: 0.999, Winds: 0.026, Snow: 0.020, Storm: 0.018, Storms: 0.006, Heavy_rain: 0.002, Snowfall: 0.001, Northeasterly_winds: 0.001, Unseasonable: 0.001, Stormy_weather: 0.001
-0.012**	0.006	US: 1.000, Opportunity: 0.011, Opportunities: 0.005, Incentive: 0.002, Privilege: 0.001, Untapped: 0.001, Dimension: 0.000, Excuse: 0.000, Opportune: 0.000, Springboard: 0.000
-0.004	0.005	Bush: 1.000, Plains: 0.009, Tropical: 0.007, Jungle: 0.006, Prairie: 0.005, Equatorial: 0.001, Pasture: 0.001, Savannah: 0.001, Safari: 0.001, Highland: 0.001

Table I.4 TFs, Private Investment Growth, and S&P 500 Returns

This table presents the TFs that could explain next quarter's Private Investment growth and S&P 500 returns. Hyperparameters were determined via five-fold cross-validation. Initially, the Lasso method was employed to select TFs with non-zero coefficients from the most important 200 TFs, followed by OLS regression to evaluate the predictability of these selected factors. Only TFs with the largest coefficient magnitudes are presented in the table (at most 10 factors), along with their coefficients and standard errors. The adjusted R^2 from the OLS regression is 0.075 for private investment growth and 0.058 for S&P 500 returns. The 'Topic Distribution' column lists up to ten of the most relevant words and their loadings within each topic. Significance levels of the coefficients are indicated as: $***p < 0.01, **p < 0.05, *p < 0.10$.

Coefficient	Standard Error	Topic Distribution
Panel A: Private Investment Growth		
-0.083**	0.027	Banks: 0.848, Banking: 0.347, Bankers: 0.259, Investors: 0.242, Brokers: 0.116, Managers: 0.101, Regulators: 0.051, Banker: 0.051, Politicians: 0.043, Lending: 0.042
-0.065**	0.022	Vol: 0.999, Pg: 0.038, Brent: 0.001, Ober: 0.001, Sem: 0.000, Fra: 0.000, Debat: 0.000
0.036**	0.018	Wire: 1.000, Rods: 0.009, Rod: 0.006, Tailhook: 0.005, Bolts: 0.004, Twine: 0.002, Piping: 0.002, Coils: 0.001, Stringing: 0.001, Clamps: 0.001
Panel B: S&P500 Return		
0.55***	0.142	Farm: 0.809, Farmers: 0.41, Agriculture: 0.299, Agricultural: 0.216, Farmer: 0.108, Rural: 0.097, Farms: 0.072, Dairy: 0.063, Growers: 0.062, Farming: 0.054
-0.384**	0.125	Credit: 0.995, Collateral: 0.076, Default: 0.051, Guaranty: 0.034, Defaults: 0.021, Lien: 0.017, Rediscounting: 0.002, Securitize: 0.002, Securitized: 0.002, Bondholder: 0.002
0.177**	0.085	Political: 0.631, Party: 0.581, Democrats: 0.376, Opposition: 0.267, Parties: 0.156, Coalition: 0.126, Democrat: 0.094, Politician: 0.03, Birthday: 0.021, Independents: 0.017
-0.128	0.149	Index: 0.997, Inventory: 0.062, Durable_goods: 0.021, Seasonally_adjusted: 0.02, Unfilled_orders: 0.018, Dreamliner: 0.003, Layoff_announcements: 0.002, Wholesale_inventories: 0.002, Widebody: 0.002
-0.081**	0.037	Net: 0.934, Operating: 0.291, Fiscal: 0.139, Consolidated: 0.124, Revenues: 0.093, Aggregated: 0.016, Reorganized: 0.006, Consolidating: 0.003, Integrated: 0.003, Non_recurring: 0.003
-0.07	0.048	Reading: 0.999, Write: 0.037, Writing: 0.012, Listening: 0.002, Composing: 0.002, Math: 0.002, Poetry: 0.001, Readings: 0.001, Mathematics: 0.001, Reciting: 0.001
-0.069	0.132	Vol: 0.999, Pg: 0.038, Brent: 0.001, Ober: 0.001, Sem: 0.000, Fra: 0.000, Debat: 0.000
0.004	0.034	Tax: 1.0, Transit: 0.02, Death_toll: 0.013, Levy: 0.012, Taxing: 0.008, Motorists: 0.006, Excise_tax: 0.004, Surtax: 0.004, Commuters: 0.003, Congestion: 0.003

Table I.5 TFs, Unemployment Rate, and Housing Price Growth

This table presents the TFs that could explain next quarter's Unemployment rate and Housing Pricing growth. Hyperparameters were determined via five-fold cross-validation. Initially, the Lasso method was employed to select TFs with non-zero coefficients from the most important 200 TFs, followed by OLS regression to evaluate the predictability of these selected factors. Only TFs with the largest coefficient magnitudes are presented in the table (at most 10 factors), along with their coefficients and standard errors. The adjusted R^2 from the OLS regression is 0.581 for unemployment rate and 0.03 for housing price growth. The 'Topic Distribution' column lists up to ten of the most relevant words and their loadings within each topic. Significance levels of the coefficients are indicated as: $***p < 0.01, **p < 0.05, *p < 0.10$.

Coefficient	Standard Error	Topic Distribution
Panel A: Unemployment Rate		
0.266***	0.042	Recession: 0.743, Unemployment: 0.525, Funding: 0.255, Jobless: 0.243, Bailout: 0.154, Stimulus: 0.114, Subsidies: 0.095, Stimulus_package: 0.023, Bailouts: 0.018, Earmark: 0.015
0.192***	0.040	Months: 0.449, Since: 0.439, Ago: 0.348, Old: 0.296, Days: 0.267, Past: 0.25, Ended: 0.248, Recent: 0.239, Weeks: 0.182, Recently: 0.159
-0.191**	0.065	Orders: 0.759, Order: 0.622, Need: 0.168, Try: 0.083, Decree: 0.039, Thereby: 0.021, Urgent: 0.018, Ordering: 0.015, Desiring: 0.004, Enjoined: 0.004
-0.139**	0.055	Johnson: 0.823, Christopher: 0.342, Benjamin: 0.169, Lewis: 0.161, Williams: 0.160, Alexander: 0.152, Cohen: 0.122, Reynolds: 0.118, Thompson: 0.101, Samuel: 0.1
0.106***	0.030	Plan: 0.684, Plans: 0.511, Program: 0.386, Budget: 0.256, Proposal: 0.139, Strategy: 0.108, Project: 0.097, Planning: 0.084, Approach: 0.068, Agenda: 0.059
-0.089*	0.049	Likely: 0.753, Going: 0.595, Unless: 0.172, Headed: 0.125, Unlikely: 0.112, Laden: 0.086, Cargo: 0.056, Loaded: 0.052, Bound: 0.05, Stuck: 0.039
-0.085**	0.030	Stock: 0.871, Dividend: 0.436, Dividends: 0.145, Income: 0.132, Stockholders: 0.112, Debentures: 0.009, Rediscount_rate: 0.008, Distributions: 0.001, Interim_dividend: 0.001, Divi: 0.001
-0.084*	0.050	Mrs: 1.0, Constituents: 0.023, Sarkozy: 0.011, Brill: 0.008, Unionist: 0.005, Tories: 0.004, Bjorn: 0.004, Maes: 0.003, Hon: 0.002, Ok_ok: 0.001
-0.082**	0.027	State: 0.964, Town: 0.157, District: 0.138, County: 0.106, Schools: 0.090, Municipal: 0.050, Village: 0.043, Neighborhood: 0.041, Teachers: 0.028, Teacher: 0.021
-0.082**	0.038	Big: 0.62, Another: 0.356, Least: 0.29, Second: 0.282, Major: 0.245, Every: 0.219, Might: 0.212, Third: 0.192, Probably: 0.173, Biggest: 0.171
Panel B: Housing Price Growth		
0.012**	0.004	Also: 0.693, Still: 0.396, Including: 0.29, Several: 0.268, Though: 0.201, Already: 0.192, However: 0.186, Meanwhile: 0.177, Although: 0.153, Added: 0.091

J. Interpreting Multi-Factor Asset Pricing Using Textual Factors

This section describes the interpretation exercise of both the time series of risk factor premia and cross-sectional betas (βs) in the Fama-French-Carhart four-factor model:

$$R_{it} - R_{ft} = \alpha + \beta_i^{Market}(R_{Mt} - R_{ft}) + \beta_i^{Size} SMB_t + \beta_i^{Value} HML_t + \beta_i^{Momentum} UMD_t + \epsilon_{i,t}, \tag{4}$$

where R_{it} is the stock return of firm i in month t, R_{ft} is the risk-free return; R_{Mt} is the return on the value-weight market portfolio; SMB_t is the return on a diversified portfolio of small stocks minus the return on a diversified portfolio of big stocks; HML_t is the difference between the returns on diversified portfolios of high and low book-to-market (B/M) stocks; and UMD_t is the difference between the returns on diversified portfolios of high and low return stocks. [23]

Time series of risk factor premia. To interpret risk premia using textual data, we construct monthly TFs based on the titles and abstracts of WSJ front-page articles. We focus on TFs that include at least ten words. To mitigate overfitting, we employ a two-stage variable selection procedure. First, we use single-variable OLS regressions to select variables significant at the 10% level. From these pre-selected variables, we apply the Lasso method with five-fold cross-validation to identify TFs with non-zero coefficients in explaining the risk factor premia. Finally, we conduct an OLS regression on the Lasso-selected variables to evaluate their performance in explaining the risk factor premia.

Tables J.1, J.2, and J.3 show TFs that explain the contemporaneous market, momentum, and value premia, respectively. We do not find any TFs that explain the premium associated with the size premium and thus do not report those results here.

Cross-sectional βs. We used the texts from the risk factor disclosure section in the companies' filings to examine the firms' beta-loadings on the four risk premia. This risk factor section, in which companies discuss potential risk factors associated with business and financial operations, has been a required disclosure since 2005, and prior studies have shown that the text in this section can explain stock returns (see, e.g., Cohen et al. (2020)). Using the text of the risk factor disclosure sections of both the quarterly report (10-Q) and the annual report (10-K) to construct TFs from 2005 to 2002, we aggregate the risk factor analyses at the year and firm level and merge the TFs with stock data using the CIK-PERMNO linking table provided by WRDS. To estimate the βs, we estimate Fama-French-Carhart four-factor models for each stock and require at least 36 months of return data to ensure robust β estimates.

Tables J.4 - J.7 report the most important TFs explaining the Fama-French factor betas. The coefficients, standard errors, and word distributions for each TF, sorted by the magnitude of their coefficients, are reported in Tables J.4 - J.7. All the TFs are normalized to have a mean of zero and a standard deviation of one for ease of interpretation. We use a similar procedure of TF selection in explaining the time series of risk premia.

[23] We obtain these four risk premia directly from Kenneth French's Website. `https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/Data_Library/det_mom_factor.html`.

Table J.1 TFs and Market Premium

This table presents the TFs that could explain market premium. We construct monthly TFs using titles and abstracts of WSJ front-page articles. To mitigate overfitting, we employ a two-stage variable selection process. Initially, single-variable OLS regressions are used to select variables significant at the 10% level. From these pre-selected variables, we apply the Lasso method with five-fold cross-validation to identify TFs with non-zero coefficients in explaining the market risk premium. Finally, an OLS regression is conducted on the Lasso-selected variables to evaluate their performance in explaining the market risk premium. The table presents only the TFs with the largest coefficient magnitudes, along with their coefficients and standard errors. The adjusted R^2 from the OLS regression is 0.095. The 'Topic Distribution' column lists up to ten of the most relevant words and their loadings within each topic. Significance levels of the coefficients are indicated as: $***p < 0.01, **p < 0.05, *p < 0.10$.

Coefficient	Standard Error	Topic Distribution
-0.119**	0.047	Crisis: 0.984, Disaster: 0.127, Poverty: 0.074, Drought: 0.050, Humanitarian: 0.040, Crises: 0.033, Floods: 0.032, Holocaust: 0.029, Hunger: 0.026, Famine: 0.022
-0.101**	0.045	Failures: 0.921, Collapse: 0.369, Collapsed: 0.107, Upheaval: 0.035, Meltdown: 0.027, Breakdown: 0.024, Crumbling: 0.019, Downfall: 0.018, Unraveling: 0.015, Toppled: 0.014
-0.087**	0.040	Tube: 0.983, Tubes: 0.152, Pneumatic: 0.101, Feeding tube: 0.021, Plugs: 0.015, Tubular: 0.007, Duct: 0.003, Valves: 0.003, Membrane: 0.001, Suction tube: 0.001
0.086**	0.042	Rubber: 0.989, Synthetic: 0.103, Cement: 0.069, Synthetic rubber: 0.060, Plastic: 0.036, Plastics: 0.022, Nylon: 0.016, Asphalt: 0.015, Rubbers: 0.006, Elastic: 0.003
-0.080*	0.044	Pfizer: 0.993, Cox: 0.103, Statin: 0.028, Statins: 0.020, Antidepressants: 0.018, Cholesterol lowering: 0.016, Diuretics: 0.014, Naproxen: 0.012, Anticlotting: 0.011, Calcium channel: 0.010
-0.078**	0.040	Rfc: 1.000, Conductors: 0.004, Termini: 0.002, Plexus: 0.001, Corpora: 0.001, Pathway: 0.001, Terminus: 0.000, Erb: 0.000, Protease: 0.000, Railway stations: 0.000
-0.077*	0.039	Chronicle: 1.000, Tale: 0.009, Documentary: 0.006, Tales: 0.005, Chronicled: 0.005, Illustrate: 0.004, Odyssey: 0.003, Biography: 0.003, Diary: 0.002, Depict: 0.001
0.073*	0.044	Modified: 0.966, Eliminated: 0.233, Switched: 0.066, Discontinued: 0.062, Scrapped: 0.048, Phased: 0.031, Discontinue: 0.028, Discontinuance: 0.019, Shelved: 0.018, Reformulated: 0.015
0.071*	0.042	Sars: 0.994, Flu: 0.085, Influenza: 0.050, Epidemic: 0.025, Contagious: 0.019, Ebola: 0.017, Dysentery: 0.012, Sniffles: 0.009, Chickenpox: 0.004, Diphtheria: 0.004
-0.065*	0.039	Nixon: 1.000, Moore: 0.020, Irwin: 0.007, Jenny: 0.003, Richie: 0.002, Leah: 0.002, Kerr: 0.002, Sara: 0.002, Underwood: 0.001, Gigi: 0.000

Table J.2 TFs and the Momentum Premium

This table presents the TFs that could explain momentum factor. We construct the monthly TFs using the titles and abstracts of WSJ front-page articles. To avoid overfitting, we first use single-variable OLS regression to select variables significant at the 10% level. For the first-stage selected variables, we use Lasso and five-fold cross-validation to select TFs with non-zero coefficients in explaining the momentum premium. Finally, we use OLS regression for the Lasso-selected variables to evaluate the performance of TFs in explaining the market risk premium. Only TFs with the largest coefficient magnitudes are presented in the table, along with their coefficients and standard errors. The adjusted R^2 from the OLS regression is 0.252. The 'Topic Distribution' column lists up to ten of the most relevant words and their loadings within each topic. Significance levels of the coefficients are indicated as: $***p < 0.01, **p < 0.05, *p < 0.10$.

Coefficient	Standard Error	Topic Distribution
-0.189***	0.052	Rally: 0.902, Beat: 0.290, Rallied: 0.230, Rallies: 0.127, Rallying: 0.102, Sank: 0.098, Edged: 0.076, Battled: 0.063, Rebounded: 0.048, Steadied: 0.017
0.183**	0.073	Date: 0.651, Calendar: 0.562, Details: 0.298, Schedule: 0.285, Names: 0.194, Schedules: 0.147, Postponed: 0.144, Dates: 0.066, Postpone: 0.034, Earliest: 0.031
-0.182***	0.039	Sun: 0.972, Garden: 0.168, Shade: 0.100, Arbor: 0.095, Grove: 0.042, Tan: 0.040, Shadows: 0.032, Shades: 0.027, Sunlight: 0.025, Shading: 0.023
0.144**	0.052	Today: 0.998, Technologies: 0.064, Tomorrow: 0.023, Computing: 0.010, Pop culture: 0.006, Travolta: 0.002, Com boom: 0.002, Cincinnati reds: 0.001, Com bubble: 0.001, Richard blumenthal: 0.001
-0.140**	0.046	Seat: 0.999, Pew: 0.028, Seat vacated: 0.010, Governorships: 0.010, Unopposed: 0.009, Berths: 0.004, Judgeship: 0.004, Speakership: 0.003, Backseat: 0.003, Seater: 0.003
0.131**	0.050	Setback: 0.802, Upset: 0.570, Hurdle: 0.154, Turnabout: 0.057, Uphill battle: 0.056, Vindication: 0.028, Stumbling block: 0.019, Comedown: 0.009, Blemish: 0.005, Downer: 0.004
-0.119**	0.039	Taliban: 0.999, Karzai: 0.036, Syrian: 0.032, Mugabe: 0.004, Jimmy carter: 0.003, Zambia: 0.002, Niger delta: 0.001, Anc: 0.001, Sonia gandhi: 0.001, Governmen: 0.000
0.103**	0.049	Prize: 0.580, Awards: 0.505, Award: 0.467, Honor: 0.359, Coveted: 0.101, Prizes: 0.091, Oscar: 0.091, Achievement: 0.087, Prestigious: 0.083, Awarding: 0.068
-0.103**	0.036	Sars: 0.994, Flu: 0.085, Influenza: 0.050, Epidemic: 0.025, Contagious: 0.019, Ebola: 0.017, Dysentery: 0.012, Sniffles: 0.009, Chickenpox: 0.004, Diphtheria: 0.004
0.100**	0.042	Absolutely: 0.983, Truly: 0.133, Utterly: 0.088, Seemingly: 0.077, Profoundly: 0.026, Singularly: 0.022, Sadly: 0.018, Manifestly: 0.012, Incomprehensible: 0.009, Spectacularly: 0.005
0.096**	0.038	Fisher: 0.992, Species: 0.111, Antarctica: 0.031, Americana: 0.031, Specimen: 0.023, Creatures: 0.023, Specimens: 0.021, Primates: 0.009, Guinea: 0.009, Specie: 0.008
0.095**	0.039	Miss: 0.993, Skip: 0.077, Forgo: 0.049, Misses: 0.047, Sore: 0.032, Dearly: 0.030, Skipped: 0.024, Spoil: 0.022, Elbow: 0.022, Rejoin: 0.013
-0.094**	0.045	Purchasing: 0.939, Investing: 0.330, Venture capital: 0.068, Diversified: 0.063, Diversification: 0.037, Reinvest: 0.005, Innovating: 0.005, Reinvesting: 0.002, Investment: 0.001, Stock picker: 0.000
-0.087**	0.041	Apple: 0.993, Apples: 0.098, Beans: 0.044, Onion: 0.020, Stalks: 0.018, Pea: 0.016, Bits: 0.016, Crust: 0.013, Onions: 0.006, Kernel: 0.005
-0.086**	0.040	Bonus: 0.998, Consols: 0.066, Butterfly: 0.009, Relay: 0.006, Talley: 0.004, Pbs: 0.003, Vaulting: 0.002, Finals: 0.002, Consolation: 0.002, Runner: 0.002
-0.085**	0.040	Engines: 0.996, Cylinder: 0.070, Turbine: 0.029, Jet engines: 0.027, Hp: 0.016, Turbo: 0.012, Propellers: 0.012, Propeller: 0.012, Engined: 0.011, Carburetor: 0.009
0.081*	0.042	Watching: 0.987, Watches: 0.139, Gazing: 0.057, Rooting: 0.027, Glued: 0.022, Munching: 0.019, Riveted: 0.015, Glancing: 0.013, Keeping$_t abs$: 0.012, $Squirming$: 0.011
0.08**	0.039	Santa: 0.991, Claus: 0.076, Santa$_c laus$: 0.070, $Halloween$: 0.048, $Donna$: 0.039, $Santas$: 0.035, $Betty$: 0.033, $Alice$: 0.026, $Lyon$: 0.009, $Halloweencostume$: 0.005
-0.078*	0.045	Cell: 0.977, Battery: 0.185, Laptop: 0.091, Handset: 0.029, Accessory: 0.023, Bulb: 0.016, Flash$_m emory$: 0.013, $Warranty$: 0.013, $Lithiumbatteries$: 0.009, $Recharged$: 0.005

Table J.3 TFs and with Value Premium

This table presents the TFs that could explain value premium. We construct the monthly TFs using the titles and abstracts of WSJ front-page articles. To avoid overfitting, we first use single-variable OLS regression to select variables significant at the 10% level. For the first-stage selected variables, we use Lasso and five-fold cross-validation to select TFs with non-zero coefficients in explaining the HML premium. Finally, we use OLS regression for the Lasso-selected variables to evaluate the performance of TFs in explaining the market risk premium. Only TFs with the largest coefficient magnitudes are presented in the table, along with their coefficients and standard errors. The adjusted R^2 from the OLS regression is 0.207. The 'Topic Distribution' column lists up to ten of the most relevant words and their loadings within each topic. Significance levels of the coefficients are indicated as: $***p < 0.01, **p < 0.05, *p < 0.10$.

Coefficient	Standard Error	Topic Distribution
0.161**	0.069	Bush: 1.000, Plains: 0.009, Tropical: 0.007, Jungle: 0.006, Prairie: 0.005, Equatorial: 0.001, Pasture: 0.001, Savannah: 0.001, Safari: 0.001, Highland: 0.001
-0.140**	0.043	Absolutely: 0.983, Truly: 0.133, Utterly: 0.088, Seemingly: 0.077, Profoundly: 0.026, Singularly: 0.022, Sadly: 0.018, Manifestly: 0.012, Incomprehensible: 0.009, Spectacularly: 0.005
-0.136**	0.066	Yr: 0.999, Conn: 0.022, Merrill lynch: 0.020, Cos: 0.019, Ecb: 0.018, Oman: 0.012, Sri: 0.012, Ind: 0.011, Nsa: 0.010, Tex: 0.007
-0.132**	0.063	Food: 0.712, Goods: 0.646, Items: 0.147, Merchandise: 0.138, Clothing: 0.097, Furniture: 0.090, Clothes: 0.070, Autos: 0.056, Raw materials: 0.052, Toys: 0.045
0.107*	0.056	Hotel: 0.812, Restaurant: 0.409, Luxury: 0.223, Casino: 0.222, Resort: 0.201, Tourist: 0.135, Resorts: 0.092, Inn: 0.050, Hospitality: 0.032, Nightclub: 0.031
-0.106**	0.043	Strikes: 0.977, Airstrikes: 0.149, Warplanes: 0.134, Walkout: 0.055, Throws: 0.031, Shutdowns: 0.029, Bombardment: 0.020, Airstrike: 0.019, Offensives: 0.009, Missile strikes: 0.009
0.103**	0.048	Supreme: 0.995, Judges: 0.090, Absolute: 0.039, Upholding: 0.023, Virtue: 0.015, Upholds: 0.008, Undoubted: 0.005, Arbiter: 0.004, Supreme leader: 0.002, Majesty: 0.002
-0.102**	0.044	Turkish: 0.831, Greek: 0.509, Hispanic: 0.180, Greeks: 0.081, Athletic: 0.070, Asian american: 0.056, Latino: 0.035, Dana: 0.032, Athletics: 0.012, Rutgers: 0.011
0.102**	0.047	Fox: 0.964, Birds: 0.256, Frogs: 0.039, Swan: 0.030, Crow: 0.028, Monkeys: 0.023, Owl: 0.023, Otter: 0.017, Owls: 0.014, Goldfish: 0.013
-0.100**	0.043	Assembly: 1.000, Assemblies: 0.018, Stamping: 0.017, Fab: 0.012, Assemblers: 0.009, Pcb: 0.006, Stampings: 0.002, Crankshafts: 0.002, Congresses: 0.001, Disassembly: 0.001
-0.098**	0.043	Modified: 0.966, Eliminated: 0.233, Switched: 0.066, Discontinued: 0.062, Scrapped: 0.048, Phased: 0.031, Discontinue: 0.028, Discontinuance: 0.019, Shelved: 0.018, Reformulated: 0.015
-0.097**	0.047	Empire: 0.982, Heir: 0.137, Titan: 0.078, Baron: 0.070, Millionaire: 0.056, Lover: 0.030, Moguls: 0.028, Scion: 0.026, Playboy: 0.023, Billionaires: 0.016
0.096**	0.043	Mills: 0.892, Mill: 0.348, Pig iron: 0.200, Factories: 0.144, Steel mills: 0.082, Steel ingot: 0.062, Foundries: 0.055, Blast furnaces: 0.048, Smelters: 0.042, Smelter: 0.037
-0.095**	0.040	Sir: 0.981, Gentleman: 0.128, Lady: 0.127, Patriot: 0.051, Acquaintance: 0.027, Honorable: 0.027, Soft spoken: 0.021, White haired: 0.017, Courteous: 0.013, Gentlemanly: 0.009
-0.088**	0.041	Locomotive: 0.966, Traction: 0.166, Furnace: 0.124, Steam: 0.104, Engine: 0.095, Heat: 0.043, Blown: 0.029, Impetus: 0.026, Combustion: 0.023, Boiler: 0.010
0.083**	0.036	Woolen: 0.990, Cloth: 0.107, Worsted: 0.075, Woolens: 0.051, Mohair: 0.008, Tweed: 0.005, Woollens: 0.004, Woolen fabrics: 0.004, Fleece: 0.004, Outerwear: 0.003
-0.074**	0.036	Furnaces: 0.999, Weld: 0.038, Sewing: 0.023, Hydraulic: 0.013, Welding: 0.013, Plumbing: 0.008, Forgings: 0.008, Saws: 0.006, Electricians: 0.005, Welded: 0.004
-0.065*	0.038	Syndicate: 0.997, Underwriting syndicate: 0.053, Underwriters: 0.044, Syndicates: 0.024, Financier: 0.021, Cartel: 0.007, Gang: 0.007, Gangs: 0.004, Smuggling: 0.004, Syndicated: 0.003

Table J.4 Interpreting Cross-Sectional β^{Market}

This table presents the TFs that could explain market β. We use the Item 7 "Risk Factors" discussion in the annual 10-K files to construct TFs at the firm and year level. We then estimate the annual beta using Fama-French regression based on the last 120 months of observations. Our goal is to examine which TFs can explain a firm's risk exposure. To avoid overfitting, we first use single-variable OLS regression to select variables significant at the 10% level. For these first-stage selected variables, we use Lasso regression with five-fold cross-validation to select TFs with non-zero coefficients in explaining the risk exposure. Finally, we apply OLS regression to the Lasso-selected variables to evaluate the performance of TFs in explaining market risk premium. Only TFs with the largest coefficient magnitudes are presented in the table, along with their coefficients and standard errors. The adjusted R^2 from the OLS regression is 0.417. The 'Topic Distribution' column lists up to ten of the most relevant words and their loadings within each topic. Significance levels of the coefficients are indicated as: $***p<0.01, **p<0.05, *p<0.10$.

Coefficient	Standard Error	Topic Distribution
0.289***	0.049	Future: 0.968, Continue: 0.226, Ongoing: 0.063, Continued: 0.046, Remain: 0.033, Intend: 0.031, Cease: 0.029, Expand: 0.028, Accelerate: 0.024, Foreseeable future: 0.022
-0.243***	0.023	Medicare: 0.474, Health: 0.457, Healthcare: 0.447, Hospitals: 0.370, Care: 0.313, Medical: 0.265, Safety: 0.224, Health care: 0.099, Education: 0.016, Occupational: 0.010
-0.198***	0.027	Public: 0.879, Local: 0.383, Regional: 0.205, National: 0.181, Community: 0.068, City: 0.017, County: 0.016, Associations: 0.013, Representatives: 0.013, Municipal: 0.010
0.177***	0.049	Certain: 0.799, Number: 0.327, One: 0.278, Various: 0.253, Many: 0.246, Several: 0.120, Particularly: 0.100, Two: 0.089, Variety: 0.071, Three: 0.065
-0.168***	0.036	Mitigation: 0.868, Remediation: 0.480, Spill: 0.102, Reactor: 0.051, Cleanup: 0.051, Containment: 0.017, Storage tanks: 0.012, Oil spill: 0.008, Wellbore: 0.004, Leak: 0.003
-0.164***	0.049	Impact: 0.773, Changes: 0.549, Effect: 0.273, Negative: 0.090, Benefit: 0.072, Effective: 0.059, Impacts: 0.052, Affects: 0.043, Promulgated: 0.042, Imposition: 0.036
-0.162***	0.033	Losses: 0.998, Chargeoffs: 0.042, Downgrades: 0.032, Delinquencies: 0.027, Impairment charges: 0.025, Writedown: 0.009, Writeoffs: 0.004, Shortfalls: 0.004, Writeoff: 0.004, Blowouts: 0.003
-0.156***	0.038	Texas: 0.996, Oklahoma: 0.064, Tennessee: 0.053, Columbia: 0.011, Mexico: 0.010, Delaware: 0.006, Florida: 0.006, Houston: 0.002, Maryland: 0.002, Victoria: 0.002
-0.156***	0.034	Products: 0.782, Product: 0.580, Technology: 0.139, Markets: 0.129, Technologies: 0.091, Applications: 0.070, Solutions: 0.026, Devices: 0.026, Brand: 0.022, Offerings: 0.021
-0.151**	0.060	Condition: 0.642, Conditions: 0.485, Regulations: 0.344, Weather: 0.325, Terms: 0.290, Rules: 0.119, Climate: 0.108, Restrictions: 0.085, Environment: 0.075, Guidelines: 0.055
0.148***	0.030	Patent: 0.693, Patents: 0.492, Intellectual property: 0.394, Proprietary: 0.247, Commercialize: 0.224, Licensing: 0.089, Trademark: 0.041, Patent litigation: 0.020, Patent infringement: 0.017, Patentable: 0.015
-0.147***	0.030	Preclinical: 0.931, Preclinical studies: 0.352, Antibody: 0.076, Neurotrophic: 0.043, Monoclonal: 0.026, Recombinant: 0.019, Vivo: 0.010, Vitro: 0.010, Transgenic: 0.009, Cytotoxic: 0.008
0.145***	0.037	Net: 1.000, Unrealized gains: 0.006, Netting: 0.001, Rebound: 0.000, Qtr: 0.000, Daily charterhire: 0.000, Inforce premiums: 0.000, Unrealized hedging: 0.000, Unaudited consolidated: 0.000, Distributable earnings: 0.000
-0.138***	0.029	Expense: 0.996, Postretirement expense: 0.071, Expenses incurred: 0.039, Nondeductible: 0.024, Accrual: 0.014, Expensed: 0.009, Noninterest expenses: 0.003, Noncovered: 0.001, Nonoperating income: 0.000, Reimbursable expenses: 0.000
0.128**	0.040	Requirements: 0.794, Demand: 0.565, Orders: 0.126, Consumption: 0.102, Consumer: 0.100, Shortages: 0.069, Supply disruptions: 0.055, Commodity prices: 0.052, Volumes: 0.048, Demands: 0.030
0.126***	0.035	Default: 0.798, Bankruptcy: 0.523, Counterparty: 0.258, Creditors: 0.120, Downgrade: 0.064, Lien: 0.049, Insolvent: 0.030, Creditor: 0.017, Bankruptcy filing: 0.014, Noninvestment grade: 0.010
-0.125***	0.023	Historically: 0.693, Typically: 0.614, Unusually: 0.257, Historical: 0.247, Usually: 0.113, Largely: 0.045, Traditionally: 0.028, Disproportionately: 0.013, Predominantly: 0.008, Statistically: 0.004
0.124***	0.029	Well: 0.996, Best: 0.084, Hard: 0.005, Poorly: 0.002, Smoothly: 0.001, Conservatively: 0.001, Perfectly: 0.001, Fantastic: 0.000, Pretty: 0.000, Comfortably: 0.000
0.122***	0.031	Develop: 0.989, Grow: 0.115, Rise: 0.060, Build: 0.047, Evolve: 0.038, Mature: 0.021, Matures: 0.014, Grown: 0.008, Grows: 0.007, Innovate: 0.003

Table J.5 The Interpretation of Cross-Sectional β^{Size}

This table presents the TFs that could explain size β. We use the Item 7 "Risk Factors" discussion in the annual 10-K files to construct TFs at the firm and year level. We then estimate the annual beta using Fama-French regression based on the last 120 months of observations. Our goal is to determine whether TFs can explain a firm's risk exposure. To avoid overfitting, we first use single-variable OLS regression to select variables significant at the 10% level. For these first-stage selected variables, we use Lasso regression with five-fold cross-validation to select TFs with non-zero coefficients in explaining the risk exposure. Finally, we apply OLS regression to the Lasso-selected variables to evaluate the performance of TFs in explaining market risk premium. Only TFs with the largest coefficient magnitudes are presented in the table, along with their coefficients and standard errors. The adjusted R^2 from the OLS regression is 0.434. The 'Topic Distribution' column lists up to ten of the most relevant words and their loadings within each topic. Significance levels of the coefficients are indicated as: $***p < 0.01, **p < 0.05, *p < 0.10$.

Coefficient	Standard Error	Topic Distribution
-0.236***	0.041	Regulatory: 0.935, Regulation: 0.233, Penalties: 0.189, Legislation: 0.163, Rule: 0.071, Regulators: 0.058, Reform: 0.029, Rulemaking: 0.011, Regulates: 0.01, Deregulation: 0.007
0.219***	0.042	Flows: 0.994, Flow: 0.109, Traffic: 0.019, Infiltration: 0.003, Movement: 0.001, Migration: 0.001, Stream: 0.0, Valves: 0.0, Valve: 0.0, Pumping: 0.0
0.201***	0.049	Future: 0.968, Continue: 0.226, Ongoing: 0.063, Continued: 0.046, Remain: 0.033, Intend: 0.031, Cease: 0.029, Expand: 0.028, Accelerate: 0.024, Foreseeable future: 0.022
0.165**	0.055	Clinical: 0.973, Patients: 0.194, Patient: 0.087, Physicians: 0.081, Physician: 0.023, Hospital: 0.018, Outpatient: 0.008, Medication: 0.005, Pediatric: 0.005, Cardiac: 0.004
-0.162***	0.029	Failure: 1.0, Refusal: 0.011, Deficiency: 0.01, Failing: 0.008, Defect: 0.006, Defective: 0.005, Abandonment: 0.005, Malfunction: 0.003, Faulty: 0.003, Omission: 0.002
-0.155***	0.031	Extent: 0.953, Jurisdiction: 0.183, Severity: 0.145, Scale: 0.113, Scope: 0.095, Magnitude: 0.073, Interpretation: 0.069, Remit: 0.065, Timeframe: 0.028, Quantity: 0.018
-0.153***	0.032	Time: 0.937, Period: 0.293, Periods: 0.178, Duration: 0.032, Prolonged: 0.032, Cycle: 0.021, Durations: 0.019, Quarters: 0.019, Cycles: 0.015, Carryforwards: 0.009
-0.138***	0.028	Tax: 0.97, Taxes: 0.221, Borrowing: 0.071, Spending: 0.053, Purchases: 0.035, Budget: 0.022, Expenditure: 0.011, Spend: 0.01, Cuts: 0.009, Discretionary: 0.007
-0.136***	0.034	Exposed: 0.796, Exposure: 0.505, Exposes: 0.296, Covered: 0.144, Harmed: 0.055, Protected: 0.012, Discovered: 0.01, Compromised: 0.009, Exacerbated: 0.006, Tested: 0.005
-0.129***	0.025	Review: 0.782, Assessment: 0.455, Evaluation: 0.201, Discussion: 0.16, Investigation: 0.159, Investigations: 0.146, Audit: 0.13, Assessments: 0.12, Update: 0.115, Reviewing: 0.067
0.127***	0.029	Predict: 0.73, Believe: 0.479, Anticipate: 0.305, Believes: 0.25, See: 0.22, Realize: 0.097, Recognize: 0.086, Prove: 0.082, Agree: 0.081, Deem: 0.033
0.112***	0.029	Respective: 1.0, Aforementioned: 0.005, Overlapping: 0.003, Mortgage guaranty: 0.0, Thir: 0.0, Alphabetical order: 0.0, Successively: 0.0, Widely divergent: 0.0, Dueling: 0.0, Successors assigns: 0.0
-0.111***	0.021	Plans: 0.998, Intended: 0.037, Attempt: 0.026, Obligated: 0.02, Targeted: 0.019, Purpose: 0.01, Intention: 0.008, Intent: 0.006, Objective: 0.004, Instructed: 0.003
-0.109***	0.029	Insurance: 0.943, Reinsurance: 0.244, Premiums: 0.13, Underwriting: 0.096, Insurers: 0.087, Insured: 0.065, Insurer: 0.054, Reinsurers: 0.047, Annuity: 0.045, Policyholders: 0.035
0.106***	0.025	Material: 0.952, Contents: 0.259, Confidential: 0.126, Contain: 0.082, Contained: 0.048, Content: 0.035, Documents: 0.027, Containing: 0.016, Stored: 0.012, Document: 0.009
-0.104***	0.022	Governmental: 0.835, Technical: 0.405, Technological: 0.355, Technological advances: 0.072, Inventions: 0.055, Rapid technological: 0.044, Innovation: 0.031, Invention: 0.026, Technological innovations: 0.02, Technological developments: 0.02
0.103***	0.031	Common: 0.709, Shares: 0.607, Shareholders: 0.25, Stockholders: 0.194, Share: 0.136, Per: 0.085, Dividend: 0.047, Compared: 0.027, Diluted: 0.008

Table J.6 The Interpretation of Cross-Sectional β^{Value}

This table presents the TFs that could explain value β. We use the Item 7 "Risk Factors" discussion in the annual 10-K files to construct TFs at the firm and year level. We then estimate the annual beta using Fama-French regression based on the last 120 months of observations. Our goal is to determine whether TFs can explain a firm's risk exposure. To avoid overfitting, we first use single-variable OLS regression to select variables significant at the 10% level. For these first-stage selected variables, we use Lasso regression with five-fold cross-validation to select TFs with non-zero coefficients in explaining the risk exposure. Finally, we apply OLS regression to the Lasso-selected variables to evaluate the performance of TFs in explaining market risk premium. Only TFs with the largest coefficient magnitudes are presented in the table, along with their coefficients and standard errors. The adjusted R^2 from the OLS regression is 0.458. The 'Topic Distribution' column lists up to ten of the most relevant words and their loadings within each topic. Significance levels of the coefficients are indicated as: $***p<0.01, **p<0.05, *p<0.10$.

Coefficient	Standard Error	Topic Distribution
-0.255***	0.050	Future: 0.968, Continue: 0.226, Ongoing: 0.063, Continued: 0.046, Remain: 0.033, Intend: 0.031, Cease: 0.029, Expand: 0.028, Accelerate: 0.024, Foreseeable future: 0.022
-0.206***	0.057	Income: 0.856, Revenues: 0.347, Expenses: 0.225, Growth: 0.180, Revenue: 0.174, Capital expenditures: 0.121, Cash flow: 0.116, Billion: 0.052, Fiscal: 0.042, Administrative expenses: 0.004
0.180***	0.042	Amount: 0.673, Amounts: 0.454, Portion: 0.379, Part: 0.363, Majority: 0.138, Bulk: 0.117, Proceeds: 0.108, Section: 0.090, Segment: 0.076, Contingent: 0.050
0.171***	0.047	Reduce: 0.999, Eliminate: 0.032, Minimize: 0.010, Maximize: 0.004, Reduces: 0.004, Eliminating: 0.001, Lowering: 0.001, Cut: 0.001, Lowers: 0.000, Minimizing: 0.000
0.170***	0.041	Availability: 0.742, Reliability: 0.585, Quality: 0.316, Unavailability: 0.076, Accessibility: 0.031, Compatibility: 0.009, Marketability: 0.003, Connectivity: 0.002, Scalability: 0.001, Inventory obsolescence: 0.000
-0.163***	0.023	Process: 0.944, Processes: 0.328, Methodologies: 0.047, Frameworks: 0.006, Supply chains: 0.004, Implementations: 0.003, Lifecycle: 0.003, Functionalities: 0.001, Automates: 0.001, Lifecycles: 0.001
0.161***	0.045	Risks: 0.772, Risk: 0.434, Factors: 0.316, Potential: 0.254, Uncertainties: 0.108, Challenges: 0.085, Effects: 0.085, Fluctuations: 0.082, Risks associated: 0.069, Difficulties: 0.063
-0.155***	0.024	Glass: 0.996, Glazing: 0.076, Glazings: 0.039, Windshield: 0.028, Glazed: 0.006, Plastic cups: 0.000, Beer bottles: 0.000, Enamels: 0.000, Plexiglass: 0.000, Cork: 0.000
-0.154***	0.038	Agreements: 0.618, Allowances: 0.570, Arrangements: 0.320, Structures: 0.262, Commitments: 0.235, Procedures: 0.201, Structure: 0.118, Instruments: 0.100, Arrangement: 0.033, Contractual obligations: 0.019
-0.146***	0.019	History: 0.898, Record: 0.394, Records: 0.136, Pace: 0.106, Exceeding: 0.091, Recording: 0.032, Lows: 0.005, Eclipse: 0.004, Totals: 0.003, Straight: 0.003
-0.137***	0.023	Intellectual: 0.986, Political: 0.155, Social: 0.055, Academic: 0.032, Literature: 0.008, Cultural: 0.006, Intrinsic: 0.003, Confiscatory taxation: 0.001, Creativity: 0.001, Cognitive: 0.000
0.135***	0.041	Interest: 0.999, Notice: 0.031, Attention: 0.030, Scrutiny: 0.024, Focus: 0.019, Publicity: 0.010, Negative publicity: 0.005, Condemnation: 0.000, Undivided attention: 0.000, Intense scrutiny: 0.000
0.132***	0.028	Real: 0.999, Ultimate: 0.035, Fact: 0.021, True: 0.016, Bona fide: 0.001, Honest: 0.000, Reality: 0.000, Realistic: 0.000, Rooted: 0.000, Noble: 0.000
-0.132***	0.028	Nonperforming: 0.827, Nonaccrual: 0.305, Nonperforming assets: 0.293, Nonperforming loans: 0.292, Loan chargeoffs: 0.154, Nonaccrual loans: 0.146, Net chargeoffs: 0.069, Delinquent loans: 0.023, Uncollectible: 0.018, Accrual status: 0.017
0.129***	0.022	Advertising: 0.979, Advertisers: 0.178, Promotional: 0.053, Advertisements: 0.050, Print: 0.050, Campaigns: 0.025, Advertiser: 0.025, Promotions: 0.019, Advertisement: 0.017, Ads: 0.016
-0.127***	0.024	Drug: 0.937, Pharmaceutical: 0.282, Drugs: 0.192, Prescription: 0.057, Controlled substances: 0.025, Prescription drug: 0.024, Antibiotic: 0.017, Prescription drugs: 0.013, Opioid: 0.013, Medications: 0.007
0.121***	0.025	Authorities: 0.986, Bodies: 0.143, Investigators: 0.077, Allegedly: 0.017, Ministry: 0.012, Suspected: 0.006, Cooperating: 0.005, Antibribery laws: 0.005, Enforcement agencies

Table J.7 The Interpretation of Cross-Sectional $\beta^{Momentum}$

This table presents the TFs that could explain momentum β. We use the Item 7 "Risk Factors" discussion in the annual 10-K files to construct TFs at the firm and year level. We then estimate the annual beta using Fama-French regression based on the last 120 months of observations. Our goal is to determine whether TFs can explain a firm's risk exposure. To avoid overfitting, we first use single-variable OLS regression to select variables significant at the 10% level. For these first-stage selected variables, we use Lasso regression with five-fold cross-validation to select TFs with non-zero coefficients in explaining the risk exposure. Finally, we apply OLS regression to the Lasso-selected variables to evaluate the performance of TFs in explaining market risk premium. Only TFs with the largest coefficient magnitudes are presented in the table, along with their coefficients and standard errors. The adjusted R^2 from the OLS regression is 0.434. The 'Topic Distribution' column lists up to ten of the most relevant words and their loadings within each topic. Significance levels of the coefficients are indicated as: $***p < 0.01, **p < 0.05, *p < 0.10$.

Coefficient	Standard Error	Topic Distribution
-0.236***	0.041	Regulatory: 0.935, Regulation: 0.233, Penalties: 0.189, Legislation: 0.163, Rule: 0.071, Regulators: 0.058, Reform: 0.029, Rulemaking: 0.011, Regulates: 0.010, Deregulation: 0.007
0.219***	0.042	Flows: 0.994, Flow: 0.109, Traffic: 0.019, Infiltration: 0.003, Movement: 0.001, Migration: 0.001, Stream: 0.000, Valves: 0.000, Valve: 0.000, Pumping: 0.000
0.201***	0.049	Future: 0.968, Continue: 0.226, Ongoing: 0.063, Continued: 0.046, Remain: 0.033, Intend: 0.031, Cease: 0.029, Expand: 0.028, Accelerate: 0.024, Foreseeable future: 0.022
0.165**	0.055	Clinical: 0.973, Patients: 0.194, Patient: 0.087, Physicians: 0.081, Physician: 0.023, Hospital: 0.018, Outpatient: 0.008, Medication: 0.005, Pediatric: 0.005, Cardiac: 0.004
-0.162***	0.029	Failure: 1.000, Refusal: 0.011, Deficiency: 0.010, Failing: 0.008, Defect: 0.006, Defective: 0.005, Abandonment: 0.005, Malfunction: 0.003, Faulty: 0.003, Omission: 0.002
-0.155***	0.031	Extent: 0.953, Jurisdiction: 0.183, Severity: 0.145, Scale: 0.113, Scope: 0.095, Magnitude: 0.073, Interpretation: 0.069, Remit: 0.065, Timeframe: 0.028, Quantity: 0.018
-0.153***	0.032	Time: 0.937, Period: 0.293, Periods: 0.178, Duration: 0.032, Prolonged: 0.032, Cycle: 0.021, Durations: 0.019, Quarters: 0.019, Cycles: 0.015, Carryforwards: 0.009
-0.138***	0.028	Tax: 0.970, Taxes: 0.221, Borrowing: 0.071, Spending: 0.053, Purchases: 0.035, Budget: 0.022, Expenditure: 0.011, Spend: 0.010, Cuts: 0.009, Discretionary: 0.007
-0.136***	0.034	Exposed: 0.796, Exposure: 0.505, Exposes: 0.296, Covered: 0.144, Harmed: 0.055, Protected: 0.012, Discovered: 0.010, Compromised: 0.009, Exacerbated: 0.006, Tested: 0.005
-0.129***	0.025	Review: 0.782, Assessment: 0.455, Evaluation: 0.201, Discussion: 0.160, Investigation: 0.159, Investigations: 0.146, Audit: 0.130, Assessments: 0.120, Update: 0.115, Reviewing: 0.067
0.127***	0.029	Predict: 0.730, Believe: 0.479, Anticipate: 0.305, Believes: 0.250, See: 0.220, Realize: 0.097, Recognize: 0.086, Prove: 0.082, Agree: 0.081, Deem: 0.033
0.112***	0.029	Respective: 1.000, Aforementioned: 0.005, Overlapping: 0.003, Mortgage guaranty: 0.000, Thir: 0.000, Alphabetical order: 0.000, Successively: 0.000, Widely divergent: 0.000, Dueling: 0.000, Successors assigns: 0.000
-0.111***	0.021	Plans: 0.998, Intended: 0.037, Attempt: 0.026, Obligated: 0.020, Targeted: 0.019, Purpose: 0.010, Intention: 0.008, Intent: 0.006, Objective: 0.004, Instructed: 0.003
-0.109***	0.029	Insurance: 0.943, Reinsurance: 0.244, Premiums: 0.130, Underwriting: 0.096, Insurers: 0.087, Insured: 0.065, Insurer: 0.054, Reinsurers: 0.047, Annuity: 0.045, Policyholders: 0.035
0.106***	0.025	Material: 0.952, Contents: 0.259, Confidential: 0.126, Contain: 0.082, Contained: 0.048, Content: 0.035, Documents: 0.027, Containing: 0.016, Stored: 0.012, Document: 0.009
-0.104***	0.022	Governmental: 0.835, Technical: 0.405, Technological: 0.355, Technological advances: 0.072, Inventions: 0.055, Rapid technological: 0.044, Innovation: 0.031, Invention: 0.026, Technological innovations: 0.020, Technological developments: 0.020
0.103***	0.031	Common: 0.709, Shares: 0.607, Shareholders: 0.250, Stockholders: 0.194, Share: 0.136, Per: 0.085, Dividend: 0.047, Compared: 0.027, Diluted: 0.008, Pershare: 0.004

K. Textual Factors for Interpreting Black Box Models

TFs allow us to project a complex black box model onto interpretable natural language space. We take Cong et al. (2019) as an example. The authors use a large Transformer-based deep learning model to construct portfolios of U.S. equities to maximize the OOS Sharpe ratio in the baseline specification. We collect the textual data on the firms' Management Discussion and Analysis (MD&A) sections of both the quarterly report (10-Q) and the annual report (10-K).

After generating TFs and each firm document's loadings on them, we regress each stock's winner score in each month from the AlphaPortfolio onto the contemporaneous textual βs. We iterate the process a few times to reduce the number of textual factors based on their interpretability, importance in the MD&A data, and significance in the AlphaPortfolio construction. Specifically, after each iteration, we discard word clusters that are incoherent or are infrequently mentioned in the firms' filing or have insignificant correlations with the winner score. Table K.1 contains examples of the most loaded topics/textual factors when constructing AlphaPortfolio. A positive coefficient indicates that, when discussions on the topic dominate the firm's text data, the firm's stock more likely receives a long position; a negative coefficient indicates the opposite. The word lists are the corresponding words within each textual factor.

The stocks AlphaPortfolio buys typically mention issues such as loss-cutting, sales, and actions as well as profitability, cash, and investment that are related to C (cash and short-term investments to total assets), C^2, investment, ipm (pretax profit margin), and ipm^2 variables identified as important by the polynomial sensitivity analysis in their paper; the stocks AlphaPortfolio short-sells are the ones heavily discussing real estate, corporate events, and acknowledging mistakes as well as uncertainty and inventory, which relate to C^2, delta_so (changes in shares outstanding), ivc (inventory changes), ivc^2, Idol_vol (idiosyncratic volatility), and Idol_vol^2. Further analysis can be conducted. For example, one can relate the negative loading on corporate events textual factor to theories explaining why stock returns may be negative on average for firms going through certain corporate events.

This textual factor analysis only constitutes an initial step towards developing a narrative for how black box models behave. Our simple choice of text data and the application of TFs are not meant to be optimal and definitive but serve as an illustration of interpreting AI models with texts.

Table K.1 Winner Scores' Loadings on Textual Factors

This table contains examples of the most loaded topics/textual factors when firms' winner scores are regressed on over 30 contemporaneous textual factor loadings selected based on factor importance and domain expertise. The regression coefficients are reported under textual topics, where a positive (negative) number indicates a long (short) position on stocks with the topic prominently showing up in the firm's filings. The remaining columns list words within each textual factor. We do not stem words because we use word vectors trained directly by Google Word2Vec in the word embedding step. For parsimony, we have only included representative textual factors.

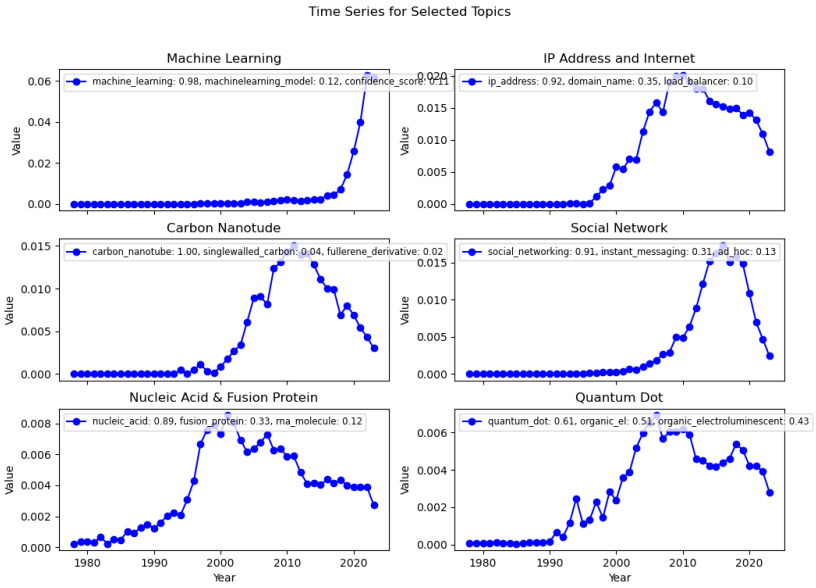
Topics	**Most Frequent Words in the Textual Factor**				
Loss-cutting (0.1500)	deferral	curtailment	divestiture	diminution	cutback
	shutdown	abolishment	retrenchment	imposition	reductions
	decrease	reclassification	amortization	resell	resale
	recycled	afford	salable		
Profitability (0.1358)	profit	ebit	quarterly	earnings	profitably
	profits	pretax	revenue	revenues	viable
	profitable	writedown	business	unprofitable	net_loss
	yoy	gross_margin	net_income	efficiency	operational
Sales (0.0428)	sales	purchases	growth	retail	earnings
	profit	income	profits	comps	resales
	shipments	sale	fiscal	revenue	
Cash/Invest (0.0381)	subsidizing	unaffordable	reimburse	reinvestment	subsidy
	invest	paying	afford	cash_trapped	tariffs
Actions (0.0339)	secures	closing	unveils	receives	exploring
	acquire	introduces	signs_agreement	deploys	stopped
	completion	donates	acquires	announces	delayed
Inventory (-0.0209)	hoarding	replenishing	stockpiling	overcharging	supplying
	accumulating	distributing	producing	restock	stockpile
	rationing	buying	storing	restocking	inventory
	transfers	seasonal	restocked	shipping	stocked
Mistakes (-0.0536)	forgiveness	confess	forgives	admit	forgiving
	contrition	forgiven	wrongs	excuses	atonement
	apologize	punish	repay	mistake	amends
	clarifications	sorry	forgave	repent	redress
Uncertainty (-0.1411)	volatility	speculator	irrationals	traders	fluctuation
	speculative	risky	turmoil	instability	uncertainty
	turbulance	changing	evolving	unpredictable	hedge
Corporate Events (-0.1444)	recapitalization	divestitures	mergers	divestiture	unbundling
	buyout	acquire	transaction	acquisitive	restructure
	acquisitions	acquired	divestiture	merger	amalgamated
	takeovers	synergies	expansions	takeover	takeover
Real Estate (-0.2362)	bungalows	dwellings	acres	houses	carport
	residences	cottages	barns	cabins	outbuilding
	homes	buildings	condos	lofts	mansion
	condominiums	townhome	real_estate	foreclosure	backyard
	apartment	farmhouse	cottage	bedroom	villa
	rented	residence	duplex	ranch	motorhome

L. Textual Factor Indices of Technological Innovations

This appendix provides some detailed explanation on the application of TFs in patent texts. Using this corpus, we first apply Word2Vec to estimate 300-dimensional embeddings for unigrams and bigrams. Following Bloom et al. (2020), to avoid the ambiguity associated with unigrams, we use bigrams to construct TFs and set the cluster size to 50.

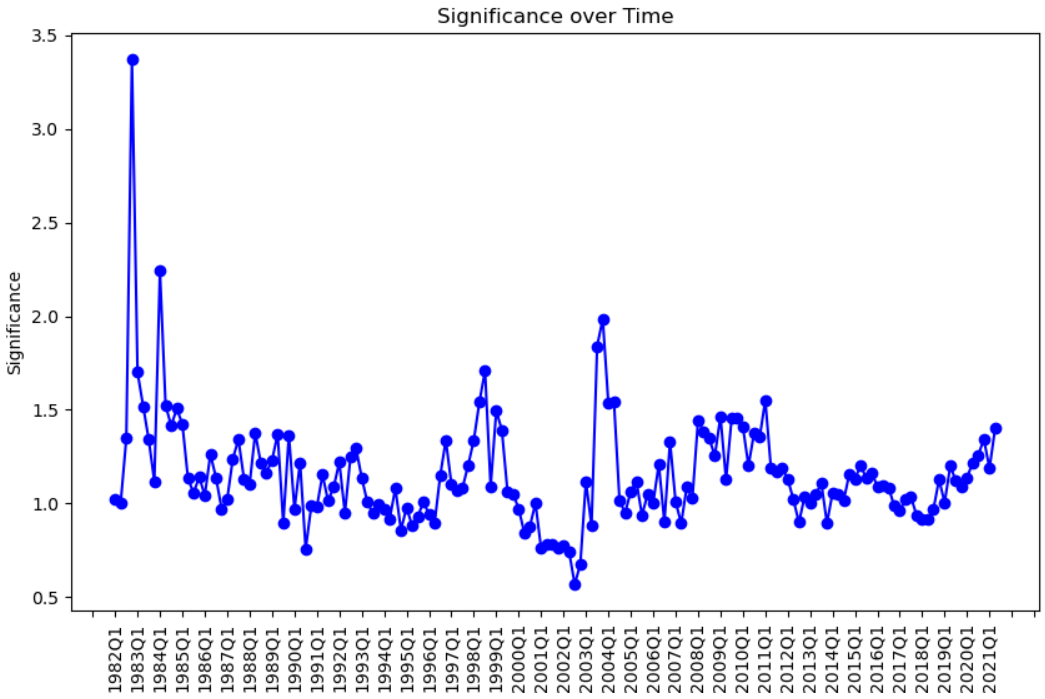
We construct a technological breakthrough index as an illustration. By analyzing patent filings, we can assess how an innovation's loading on an innovation-related TF impacts the economy's subsequent aggregate loading on the same factor. This approach complements Kogan et al. (2017)'s method of assessing the economic importance of innovations by combining patent data with the stock market response to patent news. TFs can help differentiate more refined categories of innovations, similar to Chen et al. (2019), who use Support Vector Machine (SVM) and Neural Networks to analyze patents and stock price data for various FinTech innovations. In addition to measuring technological similarity using an uncentered correlation of patent distributions (Jaffe 1986, Bloom et al. 2013, Lee et al. 2019), one can test how firms' loadings on various innovation-related TFs relate to one another.

Figure L.1 Technology Evolution over Time



Notes: This figure shows the time series of several technology breakthroughs using the US patent abstract and claims. Specifically, to measure the technology breakthrough, we first use the texts of patent abstract and claims to train a Word2Vec with 300 dimensions, and then construct the TFs. We show six of the most important topics out of the top 20 factors. Each subplot also shows the word distribution of the topic which captures a specific technology class. For example, the top-left captures the time series of machine-learning related technology, the top-right technology time series concerns ip-address and internet, while the bottom-right represents the quantum-dot technology etc.

Figure L.2 Innovation Significance over Time



Notes: This figure shows the significance of the quarterly patent using US patent abstracts and claims. To measure the significance, we first train a 300-dimensional Word2Vec model using the texts from patent abstracts and claims, construct the TFs, and then follow the methodology proposed by Kelly et al. (2021b) to assess technological significance. The figure highlights notable rises in the significance index at multiple points: around 1984, coinciding with the emergence of patents related to Monoclonal Antibodies; around 1999, associated with the surge in internet-related patents; and around 2003, linked to an increase in patents for Torque Transmitting technologies. On the construction of significance, specifically, as in Kelly et al. (2021b), each patent is represented as a vector of word frequencies, normalized using the term frequency-backward inverse document frequency (TF-IDF). The similarity between two patents is calculated as the cosine similarity between their respective word vectors, with appropriate normalization (details are provided in the paper). The significance of patent i issued in year t is defined as:

$$significance_{it} = \frac{\sum_{j\in[t,t+\tau]} similarity_{ij}}{\sum_{j\in[t-\tau,t]} similarity_{ij}}.$$

where the numerator is the sum of the similarities between patent i and any patent j issued in the following τ years, and the denominator is the sum of the similarities between patent i and any patent j issued in the preceding τ years.