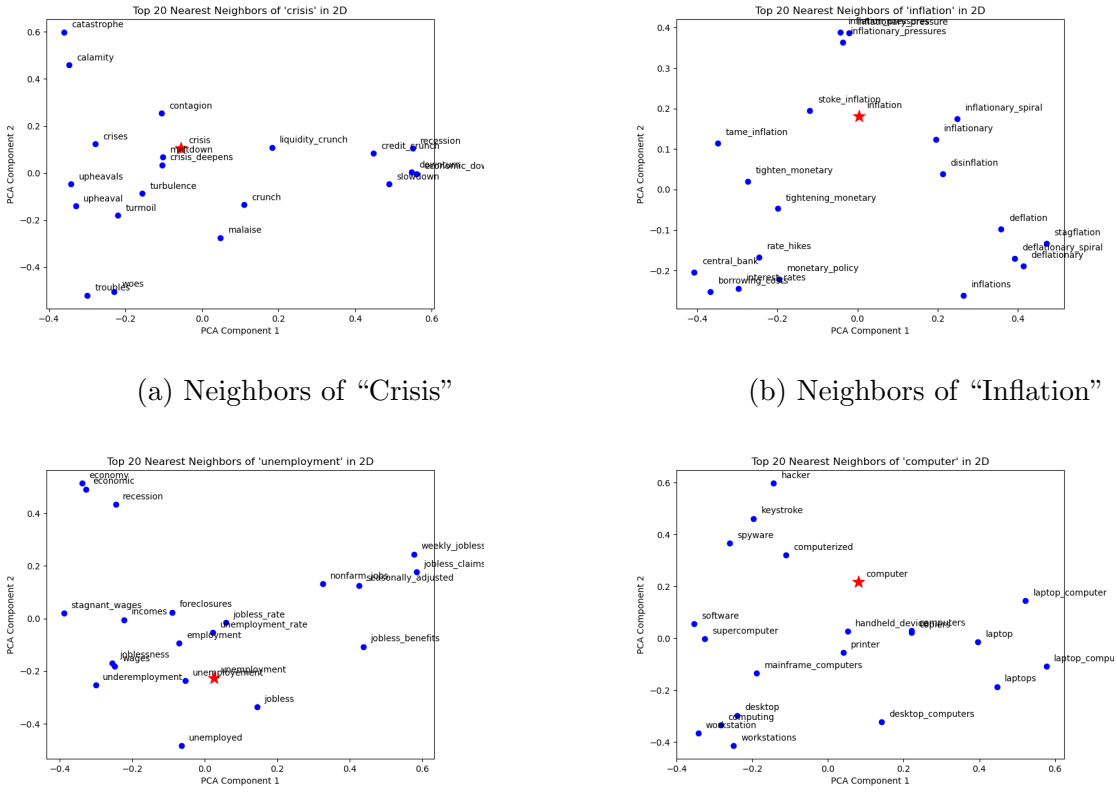
Appendix

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 associated with "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 the social sciences often adopt the "one-hot" representation, which treats words (or N-grams) as very high dimensional vectors/indices over a vocabulary with only one "1" and lots of "0"s. This representation leaves out interdependence and semantic relations, such as words similarity. It 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 word's semantic and syntactic essence. 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:[20]

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

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 the 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. Adopting word embeddings provides computational benefits and 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.[21]

B. Locality-Sensitive Hashing (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

[20] "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

[21] 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). Conversely, 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)).

We use cosine similarity as a key metric to give a concrete example of clustering similar points into a single cluster, particularly in the context of semantic vector-space models. 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 the 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 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 particular 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 rarely occurs 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 use the word frequency distribution within the clusters.

Frequency Count Approach - Topic Factor via Frequency Count. Consider the document-term sub-matrix $N_{S_i} \in \mathbb{R}^{D\times|S_i|}$. 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. The first approach directly employs Singular Value Decomposition (SVD), which is more effective but requires additional computation time. With the exception of Figure 5, we use SVD to construct the TFs throughout the paper. These methods determine the relative importance of each word for a given topic and evaluate its significance of the specific topic. SVD places more weight on words frequently used within the topic 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

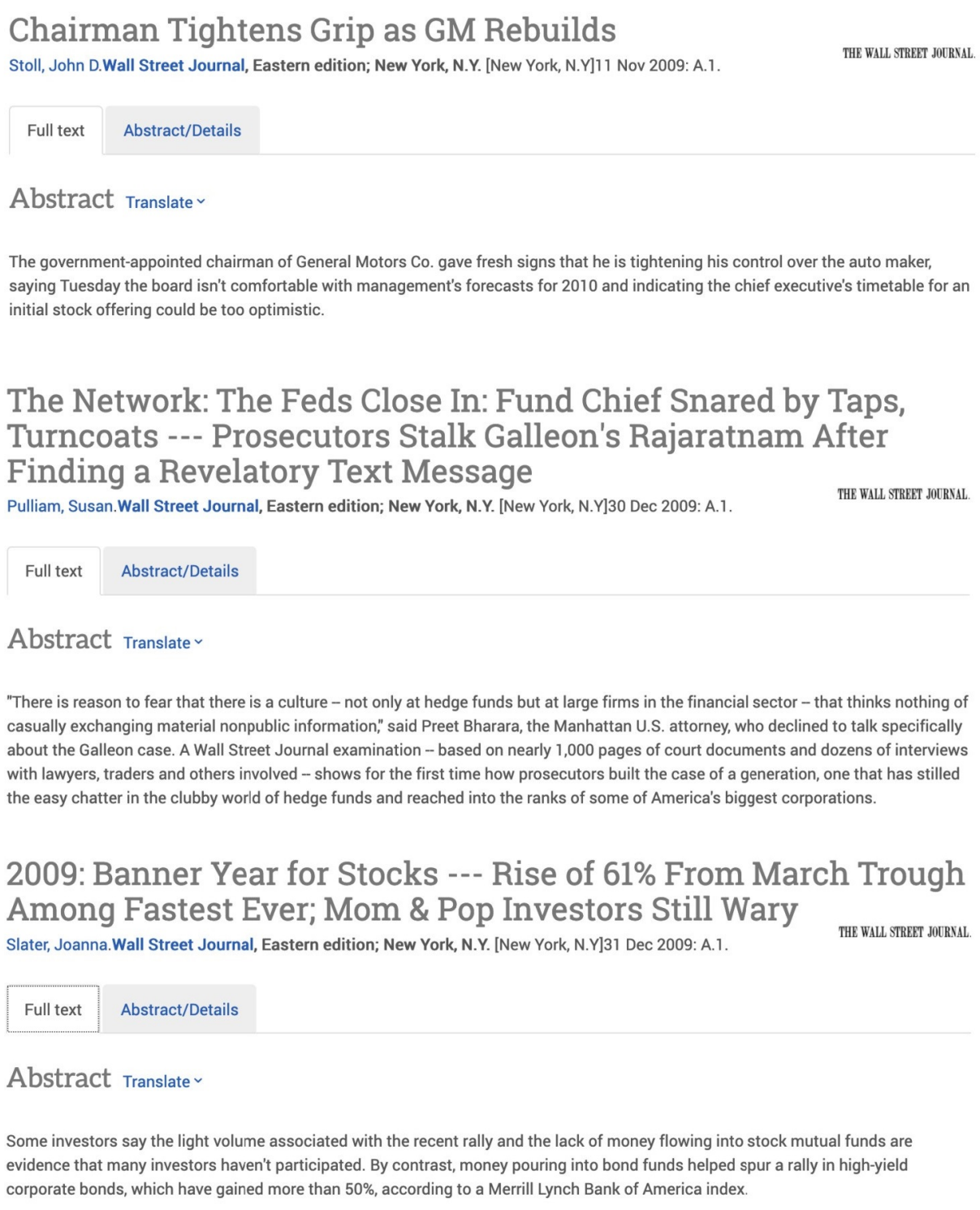
THE WALL STREET JOURNAL.

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 a $D \times |S|$ 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 corresponding to cluster S, so that these probabilities *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 characterizes 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 and its overall importance (captured by σ_i) in the universe of texts. The document loadings can be read 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 topic i but have non-zero entries, while others load fully on 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 both approaches should be explored when deciding which topics or clusters to examine first. Depending on the data set or specific application, the latter approach might help select more meaningful word clusters, even though the $\mathcal{L}^2$ 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 five words: {"predict," "predictions," "forecast," "report," "projection"}, and a corpus containing six documents. Then, we define A as 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.[22]

It should also be noted that the above SVD procedure decomposes 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.[23]

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 a 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, we apply SVM and KNN to $\mathbf{X}_{ij}^{oh}$ for fair comparisons, 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:

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

[22] 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.

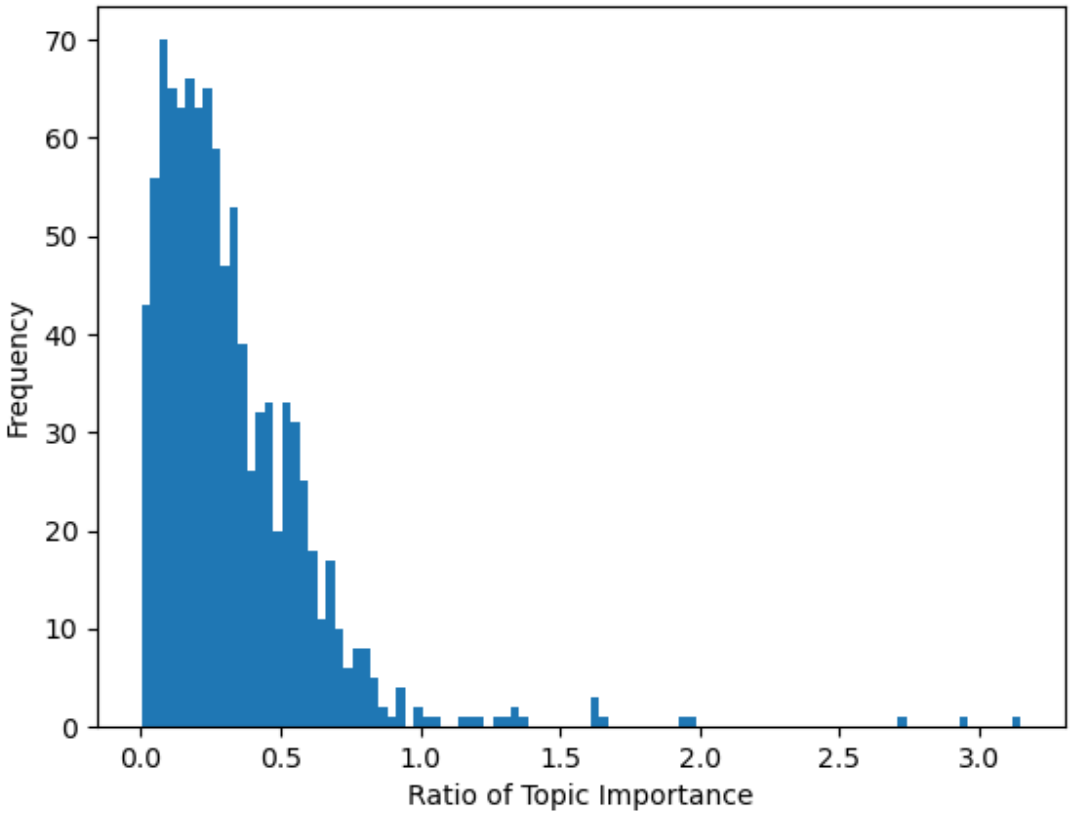
[23] Here, corpus refers to the collection of texts or documents we will analyze.

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 can 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

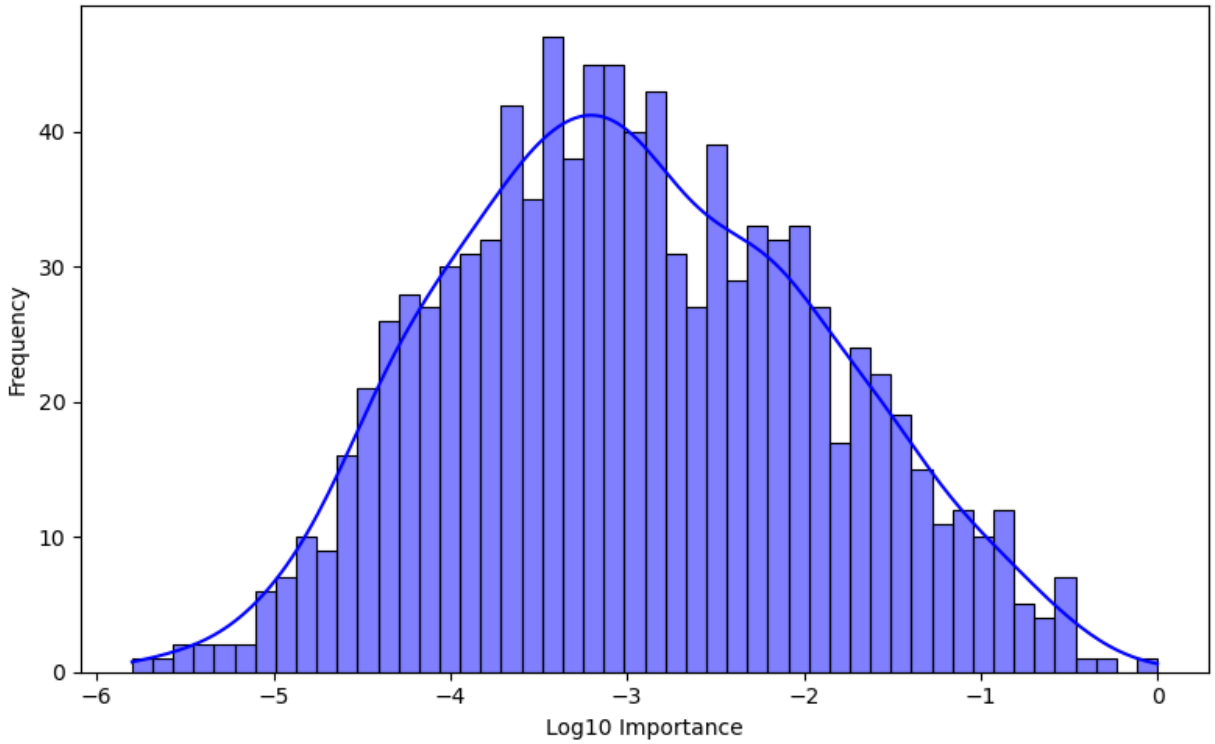
I.1. Topic Importance

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.[24]

[24] 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	0.941	0.822	0.754
GDP	0.909	0.871	0.912	0.879
Housing Price	0.968	0.829	1.006	1.000
S&P 500	1.006	0.937	1.004	0.984
Unemployment	0.539	0.766	0.808	0.831
Fixed Investment	1.000	0.754	1.000	1.134
	# of TFs = 100		# of TFs = 200	
	SVM	KNN	SVM	KNN
CPI	0.814	0.742	0.957	0.819
GDP	0.888	1.014	1.123	0.932
Housing Price	0.935	0.849	0.982	0.938
S&P 500	0.998	1.024	1.070	0.923
Unemployment	0.848	1.092	1.142	1.047
Fixed Investment	1.000	1.031	1.000	1.135
	# of TFs = 500		# of TFs = 1000	
	SVM	KNN	SVM	KNN
CPI	1.007	0.847	1.113	0.759
GDP	1.115	0.873	1.091	0.847
Housing Price	0.978	0.919	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	0.795	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	0.995	0.764
200	NA	0.939	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. [25]

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.

[25] 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, Government: 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 (Top 10 Keywords)
0.312***	0.052	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
-0.256***	0.038	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
-0.229***	0.035	Energy: 0.992, Electric: 0.085, Electricity: 0.074, Natural gas: 0.039, Utility: 0.038, Utilities: 0.012, Solar: 0.003, Grid: 0.002, Smart grid: 0.002, Renewable energy: 0.002
0.214***	0.051	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
-0.209***	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
-0.190***	0.035	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
-0.185***	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
0.175***	0.042	Requirements: 0.795, Demand: 0.565, Orders: 0.126, Consumption: 0.102, Consumer: 0.100, Shortages: 0.069, Supply disruptions: 0.055, Volumes: 0.048, Demands: 0.030, Inventories: 0.022
0.174**	0.065	Emission: 0.998, Emitting: 0.053, Methane: 0.013, Nitrogen oxides: 0.004, Diesel particulate: 0.003, Combustion byproducts: 0.001, Nitrogen dioxide: 0.001, Nitrogen oxide: 0.000, Particulate emissions: 0.000, Emit: 0.000
-0.158**	0.051	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
-0.152***	0.038	Useful: 1.000, Desirable: 0.005, Productive: 0.003, Nonproductive: 0.000, Conducive: 0.000, Rewarding: 0.000, Gainful: 0.000, Stressful: 0.000, Enjoyable: 0.000, Pleasant: 0.000
-0.148**	0.065	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
0.142***	0.042	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
0.131***	0.034	Experience: 0.735, Qualified: 0.442, Caused: 0.292, Severe: 0.244, Experienced: 0.222, Suffer: 0.204, Resulted: 0.136, Professional: 0.055, Skilled: 0.052, Occurred: 0.040
-0.129**	0.040	Service: 0.794, Access: 0.567, Providers: 0.137, Delivery: 0.119, Network: 0.114, Telecommunications: 0.035, Carrier: 0.020, Serving: 0.013, Prepaid: 0.006, Convenience: 0.005
-0.122***	0.029	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
0.122***	0.029	Securities: 0.978, Investors: 0.193, Issuer: 0.042, Securities laws: 0.042, Counterparties: 0.030, Bonds: 0.020, Issuers: 0.018, Debentures: 0.017, Broker dealer: 0.013, Brokerage firms: 0.012
-0.121***	0.037	Due: 0.933, Anticipated: 0.185, Projected: 0.160, Planned: 0.158, Expects: 0.132, Scheduled: 0.118, Uncertain: 0.082, Unexpected: 0.054, Delayed: 0.048, Unforeseen: 0.028
-0.120***	0.021	Volume: 0.877, Throughput: 0.355, NGL: 0.304, Frequency: 0.108, Thinly traded: 0.032, Throughputs: 0.015, Store sales: 0.010, Transacted: 0.003, Tonnage: 0.003, Outbound: 0.001
-0.119***	0.036	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

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 (Top 10 Keywords)
0.229***	0.051	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
-0.215***	0.043	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
0.212***	0.045	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
-0.188***	0.030	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
0.152**	0.056	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
-0.151***	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
-0.146***	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
-0.133***	0.033	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
-0.128***	0.033	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
-0.126***	0.035	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
0.117***	0.026	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
-0.116***	0.035	Prices: 0.976, Interest rates: 0.122, Commodities: 0.114, Libor: 0.077, Currency: 0.072, Commodity prices: 0.067, Borrowing costs: 0.043, Inflation: 0.033, Inflationary: 0.004, Macroeconomic: 0.002
-0.113***	0.020	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
0.108***	0.029	Replacement: 0.950, Replace: 0.260, Replacements: 0.169, Replaced: 0.022, Substitute: 0.006, Ingredient: 0.003, Depleted: 0.003, Substitutes: 0.001, Substituted: 0.001, Substitution: 0.000
-0.107**	0.040	Andor: 1.000, Mastercard: 0.003, PSD: 0.001, Transferee: 0.001, PCI: 0.000, Hereunder: 0.000, IDX: 0.000, BSD: 0.000, Orion: 0.000, Requir: 0.000
0.107**	0.042	Dependent: 0.765, Depend: 0.363, Depends: 0.322, Depending: 0.272, Rely: 0.262, Relies: 0.132, Concentrated: 0.083, Dependence: 0.074, Constrained: 0.063, Reliance: 0.036
0.102***	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
0.100**	0.033	Development: 0.910, Acquisition: 0.233, Acquisitions: 0.222, Increases: 0.178, Project: 0.138, Improvements: 0.083, Expansion: 0.079, Additions: 0.040, Expanding: 0.026, Expanded: 0.025
0.097***	0.028	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
-0.094***	0.025	Designed: 0.652, Developed: 0.647, Controlled: 0.350, Built: 0.114, Installed: 0.091, Patented: 0.071, Packaged: 0.064, Constructed: 0.035, Positioned: 0.029, Equipped: 0.024

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 (Top 10 Keywords)
-0.239***	0.055	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
0.202***	0.052	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
0.193***	0.038	Credit: 0.694, Loan: 0.681, Collateral: 0.153, Borrower: 0.090, Borrowers: 0.089, Lenders: 0.063, Revolving credit: 0.063, Refinance: 0.054, Defaults: 0.038, Credits: 0.029
-0.173***	0.027	Glass: 0.996, Glazing: 0.076, Glazings: 0.039, Windshield: 0.028, Glazed: 0.006, Plastic cups: 0.000, Beer bottles: 0.000, Cork: 0.000, Enamels: 0.000, Plexiglass: 0.000
-0.167***	0.040	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
0.163***	0.045	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
-0.157***	0.024	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
0.157**	0.061	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
-0.152**	0.061	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
0.148***	0.044	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
0.143***	0.038	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
0.138**	0.044	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
-0.137***	0.022	Volume: 0.877, Throughput: 0.355, NGL: 0.304, Frequency: 0.108, Thinly traded: 0.032, Throughputs: 0.015, Store sales: 0.010, Transacted: 0.003, Tonnage: 0.003, Outbound: 0.001
0.133**	0.050	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
-0.131**	0.050	Candidates: 0.929, Parties: 0.330, Party: 0.131, Agencies: 0.096, Vendors: 0.021, Opposition: 0.008, Stakeholders: 0.007, Committees: 0.005, Oppositions: 0.002, Allies: 0.000
-0.130***	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
-0.128***	0.038	Sell: 0.620, Purchase: 0.510, Sale: 0.387, Owned: 0.238, Acquired: 0.190, Selling: 0.188, Sold: 0.173, Owns: 0.120, Sells: 0.087, Distributed: 0.083
0.125***	0.037	Prices: 0.976, Interest rates: 0.122, Commodities: 0.114, Libor: 0.077, Currency: 0.072, Commodity prices: 0.067, Borrowing costs: 0.043, Inflation: 0.033, Inflationary: 0.004, Macroeconomic: 0.002
0.124***	0.036	Impose: 0.765, Imposed: 0.635, Constrain: 0.086, Imposing: 0.053, Declare: 0.024, Enact: 0.003, Circumvent: 0.002, Levied: 0.002, Sanction: 0.001, Rescind: 0.001
-0.124***	0.024	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

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 (Top 10 Keywords)
-0.232***	0.042	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
-0.193***	0.018	Mineral: 0.994, Royalty: 0.109, Smelter royalty: 0.009, Unpatented claims: 0.005, Farmout: 0.005, Unitization: 0.002, Farmout agreement: 0.002, Mineral extraction: 0.001, Optionor: 0.001, Overriding royalty: 0.001
-0.192***	0.048	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
-0.176**	0.055	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
0.161***	0.033	Development: 0.910, Acquisition: 0.233, Acquisitions: 0.222, Increases: 0.178, Project: 0.138, Improvements: 0.083, Expansion: 0.079, Additions: 0.040, Expanding: 0.026, Expanded: 0.025
0.160***	0.033	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
0.155**	0.049	Rate: 0.801, Standards: 0.496, Levels: 0.236, Level: 0.165, High: 0.089, Grade: 0.072, Low: 0.061, Average: 0.055, Minimum: 0.054, Target: 0.049
-0.154***	0.032	Capacity: 0.783, Facility: 0.573, Unit: 0.230, Terminal: 0.047, Centers: 0.041, Center: 0.025, Site: 0.023, Onsite: 0.010, Hub: 0.007, Beds: 0.006
-0.152***	0.042	Enter: 0.600, Invest: 0.499, Engage: 0.379, Expose: 0.351, Engaged: 0.219, Participate: 0.201, Utilize: 0.111, Commit: 0.101, Integrate: 0.060, Initiate: 0.059
0.139***	0.035	Perform: 0.652, Conduct: 0.556, Recorded: 0.295, Completed: 0.218, Operated: 0.207, Conducted: 0.170, Performed: 0.167, Conducting: 0.113, Represented: 0.080, Administered: 0.054
0.126***	0.037	Prices: 0.976, Interest rates: 0.122, Commodities: 0.114, Libor: 0.077, Currency: 0.072, Commodity prices: 0.067, Borrowing costs: 0.043, Inflation: 0.033, Inflationary: 0.004, Macroeconomic: 0.002
0.126***	0.027	Commercially: 0.978, Commercially viable: 0.153, Commercialized: 0.105, Sufficient quantities: 0.087, Commercially exploitable: 0.044, Experimental: 0.022, Commercially feasible: 0.010, Scientifically: 0.004, Technically feasible: 0.001, Conventionally: 0.001
0.116***	0.021	FCC: 1.000, NBA: 0.015, Cablecard: 0.001, Lebron: 0.000, SGI: 0.000, Linux: 0.000, ISO: 0.000, MPEG: 0.000, ISOs: 0.000, SQL Server: 0.000
-0.115**	0.038	Sufficient: 0.792, Possible: 0.388, Acceptable: 0.270, Reasonable: 0.264, Appropriate: 0.218, Suitable: 0.123, Satisfactory: 0.079, Discretion: 0.073, Inappropriate: 0.042, Unreasonable: 0.038
0.108***	0.033	Use: 0.991, Usage: 0.082, Distribute: 0.079, Users: 0.044, Install: 0.036, Employ: 0.030, Utilization: 0.026, Prescribe: 0.012, Possess: 0.010, Misuse: 0.010
-0.107**	0.047	Adversely: 0.540, Affect: 0.523, Adverse: 0.368, Adversely affect: 0.330, Affected: 0.262, Negatively: 0.214, Materially: 0.164, Adversely affected: 0.128, Negatively impact: 0.116, Affecting: 0.102
0.104***	0.016	Gold: 0.959, Metals: 0.189, Ore: 0.171, Metal: 0.084, Zinc: 0.063, Precious metals: 0.052, Copper: 0.037, Ores: 0.024, Feldspar: 0.012, Precious metal: 0.009
0.104***	0.027	Reimbursement: 0.945, Payors: 0.271, Payers: 0.170, Payor: 0.054, Party payors: 0.015, Formulary: 0.015, Reimbursable: 0.014, Formularies: 0.014, Per diem: 0.010, Copayment: 0.007
0.103***	0.022	Internal: 0.927, Independent: 0.361, External: 0.100, Optical: 0.006, Externally: 0.004, Peripheral: 0.001, Macro: 0.000, Interparticle: 0.000, Exogenous: 0.000, Externally sourced: 0.000

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. (2022) 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 D.1 in Cong et al. (2021b) 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.

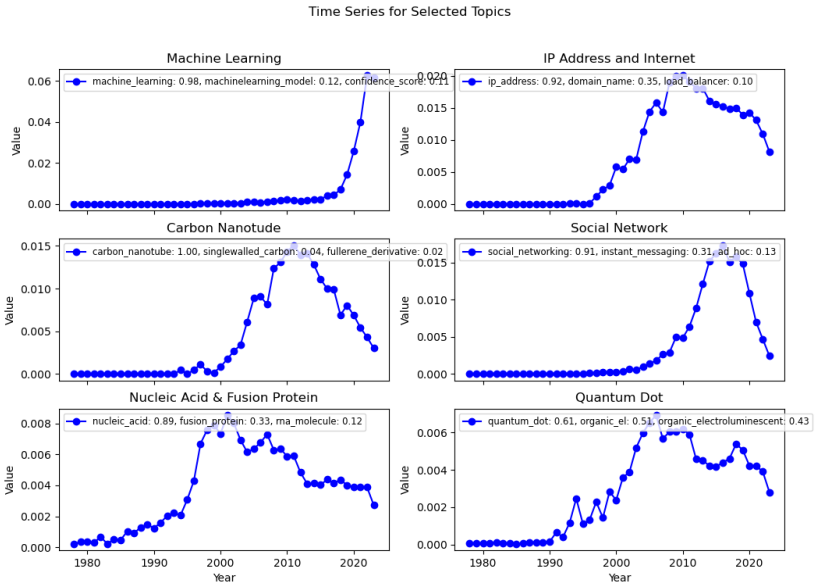
This textual factor analysis only constitutes an initial step towards developing a narrative for how black box models behave. Our simple choice of text data and the application of TFs are not meant to be optimal and definitive but serve as an illustration of interpreting AI models with texts.

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.

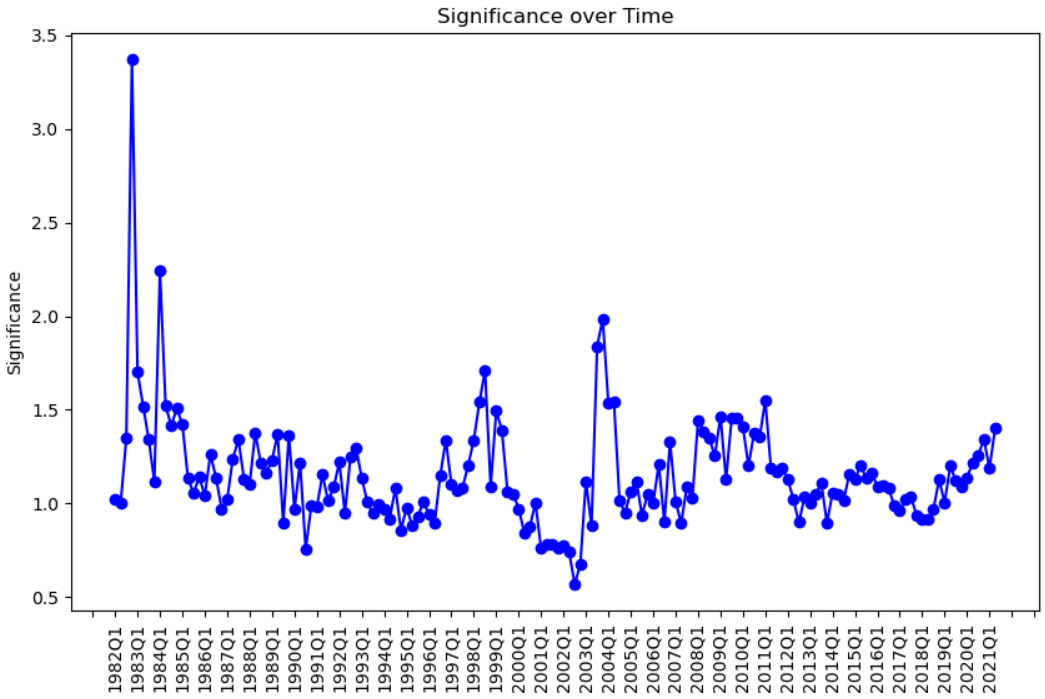
We construct a technological breakthrough index as an illustration. By analyzing patent filings, we can assess how an innovation's loading on an innovation-related TF impacts the economy's subsequent aggregate loading on the same factor. This approach complements Kogan et al. (2017)'s method of assessing 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.

M. A Comparison with LLM (e.g., ChatGPT-3.5)

This section presents a preliminary investigation of the performance of GPT-3.5-turbo on macroeconomic forecasts as an illustration of how TFs compare with and relate to LLMs. Our sample consists of Wall Street Journal (WSJ) news articles issued between January 1984 and December 2022. To reduce computational costs, 300 news articles are randomly sampled each month. For each article, GPT-3.5-turbo is instructed to analyze the news title and full context to predict whether the article indicates a change in six key macroeconomic variables: CPI, GDP growth, housing price growth, stock prices, unemployment rate, and fixed investment rate following Bybee (2023). Specifically, GPT-3.5-turbo determines whether each macroeconomic variable is likely to increase ("going up"), decrease ("going down"), or remain uncertain ("unknown"). Table M.1 summarizes the percentage distribution of these predictions for each macroeconomic variable. For example, the first column of the table reports the distribution for the CPI, showing the proportion of news articles classified as indicating an increase, a decrease, or uncertainty.

Overall, GPT-3.5-turbo delivers mixed results in terms of predictability, as measured by adjusted R^2. Specifically, our textual factors (TFs) achieve an adjusted R^2 of 0.300 for CPI growth, 0.135 for GDP growth, 0.075 for fixed investment growth, 0.058 for stock price return, 0.581 for unemployment, and 0.030 for housing price growth (see subsection I.3 for details). In comparison, GPT-3.5-turbo achieves adjusted R^2 values of 0.016 for CPI growth, 0.125 for GDP growth, 0.307 for fixed investment growth, -0.003 for stock price return, 0.344 for unemployment and 0.167 for housing price growth.

The results reveal that TFs outperform GPT-3.5-turbo in the predictability of CPI growth, stock price returns, and the unemployment rate. However, GPT-3.5-turbo surpasses TFs in predicting fixed investment growth and housing price growth. These findings highlight the nuanced strengths and weaknesses of GPT-3.5-turbo. While its ability to process complex textual data can lead to strong performance in certain areas, such as fixed investment and housing prices, its lower adjusted R^2 in CPI and stock price predictions suggests limitations in capturing specific economic signals. It is worth noting that Chen et al. (2024a) employs GPT-3.5 to extract signals from headlines of economic or finance-related news appearing on the front page, achieving predictability levels comparable to those of our TFs. These differences underscore the need for further refinement of GPT-based approaches to better extract and leverage macroeconomic insights from textual data.

In addition to macroeconomic predictability, GPT-3.5-turbo is employed to extract at most five of the most relevant topics from each news article. However, this task presents notable challenges. Traditional topic modeling techniques, such as Latent Dirichlet Allocation (LDA) and Latent Semantic Analysis (LSA), leverage global information by analyzing all articles collectively to generate consistent topics and their associated loadings. In contrast, GPT-3.5 operates on individual news articles due to its token limitations, which prevent it from considering the broader corpus simultaneously. This limitation leads to significant inconsistency in topic extraction, as GPT-3.5 lacks the ability to align topics across articles or over time.

One clear manifestation of this inconsistency is the rapid growth in the number of unique topics over time. Figure M.1 illustrates the cumulative number of unique topics extracted by GPT-3.5. The figure shows a sharp increase in unique topics, many of which are semantically similar but described differently by GPT-3.5.

For instance, terms like "unemployment rate increase" and "rising joblessness" might represent the same concept but are treated as distinct topics by GPT. This inconsistency undermines the interpretability of the extracted topics and challenges their utility in longitudinal analyses.

To address this issue, one promising approach is to create embeddings for each topic extracted by GPT-3.5 and then cluster semantically similar topics, as advocated by our paper. Such clustering would significantly reduce the number of unique topics and improve the coherence and interpretability of the results, aligning the GPT-based approach with traditional topic modeling methods that rely on global context.

The inconsistency of GPT-3.5 in topic extraction becomes particularly evident in the frequency distribution of topics. Over the entire sample, GPT-3.5 extracts more than 310,000 unique topics, yet most of these topics appear only once. Figure M.2 presents a histogram of topic frequencies, highlighting the skewed distribution. While a small number of topics appear with high frequency (e.g., more than 1,000 occurrences), the vast majority of topics occur only once. This extreme sparsity hinders meaningful interpretation and limits the reliability of the extracted topics for broader applications.

In conclusion, while GPT-3.5 demonstrates promise in predicting macroeconomic variables and identifying topics, its token limitations and lack of global context introduce significant drawbacks in topic extraction. Addressing these challenges, such as through topic clustering, is essential for enhancing the utility of GPT-3.5 in longitudinal and large-scale textual analyses.

Another consideration worth emphasizing is the cost associated with the LLMs. To give a concrete example of the costs associated with using GPT-3.5-turbo, each month, we randomly sampled 300 news articles per month from the Wall Street Journal (WSJ), including both the title and full text, as discussed in this section. This sample represents approximately 1/12 of the total number of articles in our dataset, which spans from January 1984 to December 2022. Processing these 300 articles for sentiment analysis and topic extraction using GPT-3.5-turbo required approximately three days and incurred a cost of $300 USD.

Extrapolating these figures to the full dataset, processing all WSJ articles from 1984 to 2022 would take an estimated 36 days and cost approximately $3,600 USD. In contrast, processing the same dataset using TFs required significantly less time and resources, as shown in Table 1 of the manuscript.

Table M.1 Distribution of Sentiment Across Macroeconomic Variables

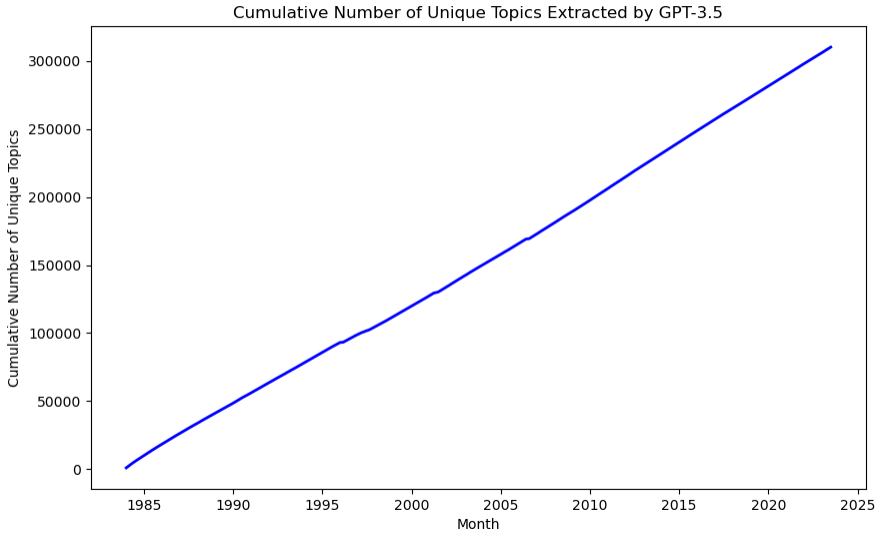
Sentiment	CPI	GDP	Housing Prices	Stock Prices	Unemployment	Fixed Investment
Going Down	0.121	3.955	2.007	7.636	0.885	4.995
Going Up	1.145	10.408	2.718	15.148	3.208	16.565
Unknown	98.734	85.637	95.274	77.216	95.906	78.440

Table M.2 GPT's Predictability on Macroeconomic Variables

This table reports the predictability of GPT-3.5 on macroeconomic variables at the quarterly level. Between 1984 and 2022, each month, 300 news articles, including the title and full texts, were randomly sampled, and GPT-3.5 was instructed to predict whether the news indicated an increase, a decrease, or no change (unknown) in the corresponding macro variable. The percentages of articles indicating "going up" and "going down" were calculated as GPT-sentiment. Lagged GPT-sentiment was then used to predict the corresponding macroeconomic variable for the next quarter. $***p < 0.01, **p < 0.05, *p < 0.10$.

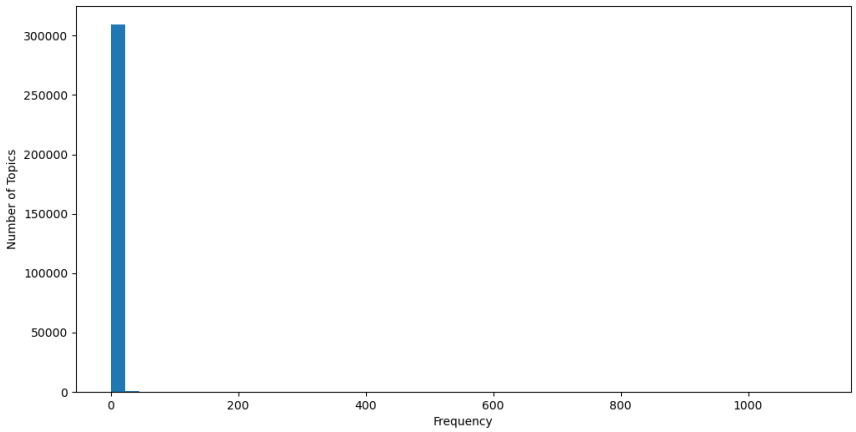
	CPI	GDP	Housing Prices	Stock Prices	Unemployment	Fixed Investment
const	0.206***	0.843***	1.160***	1.709	3.761***	0.733
	(0.026)	(0.284)	(0.430)	(2.658)	(0.308)	(0.551)
CPI_down	-0.085					
	(0.112)					
CPI_up	0.031**					
	(0.015)					
Fixed Investment_down						-0.567***
						(0.084)
Fixed Investment_up						0.195***
						(0.030)
GDP_down		-0.097***				
		(0.029)				
GDP_up		0.046**				
		(0.021)				
Housing Prices_down			-0.660***			
			(0.126)			
Housing Prices_up			0.218			
			(0.134)			
Stock Prices_down				-0.232		
				(0.230)		
Stock Prices_up				0.154		
				(0.156)		
Unemployment_down					0.577**	
					(0.275)	
Unemployment_up					0.576***	
					(0.071)	
Adjusted R^2	0.016	0.125	0.167	-0.003	0.344	0.307

Figure M.1 The Cumulative Number of Unique Topics Over Time



Notes: This figure illustrates the cumulative number of unique topics identified over time. To construct the figure, we track the first appearance of each topic extracted by GPT-3.5 and compute the cumulative count of unique topics up to period t. The results clearly demonstrate an increasing trend in the number of unique topics over time. This pattern suggests that GPT-3.5 does not consistently extract the same topics across periods, likely due to its inability to process all news articles in a single batch because of token limitations.

Figure M.2 Distribution of Topic Frequency



Notes: This figure shows the distribution of topic frequency. Specifically, for each unique topic extracted by GPT-3.5 turbo, we calculate its frequency , and then we plot the histogram of the frequency. This figure clearly shows that only a very small fraction of topics appear more than 1000 times, with nearly all topics appearing one time, indicating the GPT cannot well extract consistent topics over time.