Submitted to *INFORMS Journal on Computing*
manuscript (Please, provide the manuscript number!)

Analyzing Firm Reports for Volatility Prediction: A Knowledge-Driven Text Embedding Approach

Yi Yang
Hong Kong University of Science and Technology, imyiyang@ust.hk,

Kunpeng Zhang
University of Maryland, College Park, kpzhang@umd.edu,

Yangyang Fan
Hong Kong Polytechnic University, yangyang.fan@polyu.edu.hk,

Predicting stock return volatility is the key to investment and risk management. Traditional volatility forecasting methods primarily rely on stochastic models. More recently, many machine learning approaches, particularly text mining techniques, have been implemented to predict stock return volatility, thus taking advantage of the availability of large amounts of unstructured data such as firm financial reports. Most existing studies develop simple but effective models to analyze text, such as dictionary-based matching algorithms that use a set of manually constructed keywords. However, the latent and deep semantics encoded in text are usually neglected. In this study, we build on recent progress in representation learning and propose a novel word embedding method that incorporates external knowledge from a well-known finance-domain lexicon (L&M Dictionary), which helps us learn semantic relationships among words in firm reports for better volatility prediction. Using over ten years of annual reports from Russell 3000 firms, we empirically show that compared to cutting-edge benchmarks, our proposed method achieves significant improvement in terms of prediction error, e.g., a 28.4% reduction on average. We also discuss the practical and methodological implications of our findings. Our financial-specific word embedding program is available as open-source information so that researchers can use them to analyze financial reports and assess financial risks.

Key words: Volatility prediction, machine learning, word embedding, knowledge, L&M dictionary

1. Introduction

Stock return volatility, formally defined as the standard deviation of stock return over a period of time, has been frequently used to measure the market risk associated with many downstream financial market-related tasks, including investment portfolio management and risk management. Volatility is also considered a key component for price evaluation in many well-established option-pricing models (Black and Scholes 1973). Thus, understanding and particularly forecasting volatility is of great interest to capital market participants.

Financial economic theory has shown that it is feasible to predict stock volatility using publicly available data (Bernard et al. 2007). In fact, researchers and practitioners have been exerting tremendous effort to predict volatility using various types of data. For example, Poon and Granger (2003) quantitatively model volatility in a time-varying manner by incorporating historical volatilities and propose ARCH-type models in the family of stochastic volatility. In addition, researchers in recent years have sought to improve prediction by analyzing unstructured text from firm financial reports, such as companies' annual reports (i.e., 10-K filings). In the United States, Securities and Exchange Commission (SEC) requires every publicly traded firm to file a comprehensive 10-K report each year. This report provides a comprehensive overview of a firm's business, financial condition and risk factors. The question of how to extract useful and informative information from this goldmine of data, especially in order to predict volatility, has led to a research trend in both the financial economics and the computational linguistics communities.

To analyze annual reports, the simplest and most effective text mining approach adopted by economists and finance researchers is dictionary-based. The Loughran and McDonald word list (hereafter L&M Dictionary) is the most commonly used finance-domain lexicon (Loughran and McDonald 2011). Built by manually examining word usage in firms' 10-K filings, the lexicon only includes words actually used by managers in financial reports, and it groups financial terms into different categories, such as positive, negative, uncertainty, and litigation. Prior research using the L&M Dictionary has shown that the words used in annual reports are informative in predicting a firm's future stock return volatility. In addition, advanced machine learning methods such as Naive Bayes have been developed to analyze firms' annual reports for volatility prediction (Li 2010). Along these lines, annual reports are often represented as having bag-of-words features.

However, existing volatility prediction approaches using financial reports fail to capture the complex structures within texts. Hand-crafted lexicons only contain a small set of words belonging to several pre-defined categories, and bag-of-words models are limited by their representation power; for example, the order of words is usually ignored. Recent advances in representation learning, such as word embedding (Mikolov et al. 2013) allow us to learn deep and latent structures encoded in financial reports, which also motivates our present study. Word embedding (a.k.a. neural word representation) has increasingly become an important building block of many text analysis tasks. Unlike one-hot encoding[1], each word token is a multi-dimensional vector in the word embedding representation. The multi-dimensional vectors for all words in the corpus are automatically learned from a deep neural network, where the objective is to predict a word using contextual words while word semantic and syntactic information are as well preserved as possible.

Word embedding is often pre-trained without any human supervision on very large generic text data. It has achieved superior performance in many natural language processing (NLP) tasks. Directly applying these word embeddings to financial reports may not be appropriate because words and expressions used by managers in financial reports are very different from those in general text, e.g., news articles and web discussion forums (Loughran and McDonald 2011). Even when we train our own word embeddings using annual reports, due to the relatively small scale it is still challenging to effectively achieve better representation while also capturing the linguistic characteristics and written styles of reports well. Therefore, a finance-specific domain lexicon, which contains expert knowledge about how words are categorized in business communication, becomes a natural complement to unsupervised representation learning in achieving high-quality finance-specific word embeddings. For example, the word "exonerate" is not frequently used in business communication documents, so due to this infrequency, generic vanilla word embedding training may not generate good representation for "exonerate." However, the L&M Dictionary clearly puts this word in the "Negative" category. Thus, the domain-specific lexicon provides enough knowledge about this word (as well as other words in the Negative category) to enhance the overall word representation. More importantly, once we obtain high-quality finance-specific word embeddings, we can better understand the information content of reports, which can subsequently improve volatility prediction accuracy for firms.

Motivated by this, we intend to enhance representation learning by incorporating external expert knowledge, namely, the L&M Dictionary, a human-crafted finance lexicon, for financial report analysis. To do this, we transform domain knowledge into "must-link" and "cannot-link" word associations. Specifically, we create a must-link for two words in the same category defined by the L&M Dictionary and a cannot-link for two words in

[1] Traditional one-hot encoding and bag-of-words represent each document as a $|V|$-dimensional vector where $|V|$ is the number of distinct words in the vocabulary. Each element in the vector corresponds to a unique word (token) in the vocabulary. If the token at a particular index exists in the document, that element is marked as 1, 0 otherwise.

different categories according to the L&M Dictionary. The must-links and cannot-links serve as extra constraints in the word embedding learning objective, for which we solve the optimization problem using stochastic gradient descent.

To evaluate the performance of our proposed method, we conduct experiments on a dataset collected from EDGAR SEC that contains $36,768$ 10-K filing reports of Russell 3000 companies between 2004 and 2017. In terms of stock volatility prediction error reduction, our proposed approach of incorporating a finance-specific lexicon into representation learning yields significant improvement over various state-of-the-art baselines, including market-data model, bag-of-words models, topic models and generic word embedding models. Specifically, we reduce the volatility forecast error by 22.2% over a bag-of-words model, 31.3% over a topic model, and 6.5% over vanilla word embeddings without knowledge incorporation. Our experimental results also suggest that incorporating both “must-link” and “cannot-link” constraints can facilitate better learning of word embeddings than does incorporating only one constraint type. Further, we confirm that the advantage of incorporating domain-specific lexicons for word embedding still holds even when we use deep learning models for volatility prediction.

Our knowledge discovery-driven text embedding approach for stock volatility prediction makes the following twofold contributions.

- First, we are among the first to incorporate a finance-specific lexicon into representation learning for stock volatility prediction. We propose a knowledge-driven text embedding model trained on a large amount of unstructured text data to learn high-quality word embeddings. Empirical experiments on volatility prediction show that the domain lexicon-enhanced representation learning can indeed significantly improve the performance (e.g., reduce the prediction error) over bag-of-words models and generic vanilla word embeddings. Our approach outperforms the stock volatility prediction benchmarks and serves as a new tool for volatility forecasting in addition to traditional stochastic models.
- Second, we add to relevant accounting and finance text analysis research by demonstrating a new pipeline for analyzing corporate reports. As mentioned, much of the existing literature in accounting and finance uses bag-of-words approaches to represent documents. However, as bag-of-words approaches are limited in understanding semantic similarities between words, there is a call for developing advanced machine learning approaches to capture deeper meanings and context in business text (Loughran and McDonald 2016). In this

work, we shift from a traditional bag-of-words approach to a data-driven representation learning approach and demonstrate that incorporating a finance-domain lexicon with word representation learning can better encode document semantics. Further, the new learned word embeddings can be applied to relevant finance tasks and can be of broad interest to finance practitioners and researchers.

2. Related Work

Our work is closely related to three lines of research: stock volatility forecasting using publicly available data, text analysis for financial documents and knowledge-driven machine learning.

2.1. Stock Volatility Forecasting Using Public Data

Received wisdom in economics and finance holds that one can predict stock volatility levels using publicly available data (Bernard et al. 2007). For example, Sridharan (2015) uses financial statements to analytically predict future stock volatility. Other research has also examined stock volatility predictors from different sources of public information. Among these sources, firms' annual reports, also known as 10-K filings, are inarguably the most important. Kothari et al. (2009), Loughran and McDonald (2011) have documented that when text analysis indicates favorable disclosures in 10-K filings, a firm's stock return volatility tends to decline significantly, while unfavorable disclosures are associated with a significant increase in volatility.

In addition to drawing from 10-K filings, recent research has reported the relationship between stock volatility and other sources of public data, including firms' quarterly earning conference calls and firm-specific news. Frankel et al. (1999) shows that stock volatility is elevated during conference calls and earnings press releases, suggesting that earnings announcements provide information to the market beyond that which is found in the press release alone. They find that a positive tone is negatively related to volatility. Tetlock (2007) and Boudoukh et al. (2018) find that text information in firm-specific news greatly accounts for stock volatility. Further, researchers use text analysis on IPO prospectuses and provide empirical evidence that IPOs with high levels of uncertain text have higher subsequent volatility (Loughran and McDonald 2013).

Overall, the above studies suggest that the linguistic content of public data has predictive power in explaining stock volatility. Our work is different in that we focus on forecasting future stock volatility by leveraging the linguistic content of public data.

2.2. Text Analysis for Financial Documents

A large body of literature in finance and accounting leverages text analysis tools (see review by Loughran and McDonald (2016)). Most, if not all, of these studies rely on NLP algorithms that assume a bag-of-words structure in texts, where words are independent by assumption. That is, these models disregard grammar and word order and thus the context of a word. The word features used in the bag-of-words models are either all words from the text corpus (Li 2010) or from a lexicon in a specific domain, such as the L&M Dictionary (Loughran and McDonald 2011, Bodnaruk et al. 2015). Following earlier studies in NLP, research in finance and accounting usually states that bag-of-words algorithms yield effective results for financial document representations.

In the computational linguistics community, several effective machine learning approaches have been developed for analyzing financial documents. One pioneering work by Kogan et al. (2009) shows that simple bag-of-words features in 10-K filings combined with historical volatility can simply outperform statistical models that are built upon historical volatility only. Other work, such as Rekabsaz et al. (2017), proposes constructing weighted bag-of-words features to represent 10-K filings. Jegadeesh and Wu (2013) designs an appropriate strategy to select word weights in text analysis. Qin and Yang (2019) combine audio and text information from earnings conference calls for volatility prediction. Our work advances prior computational linguistic research by designing a novel machine learning method for domain-specific word embedding learning. We empirically show that our proposed method significantly outperforms these traditional text analysis methods (see details in later sections).

2.3. Knowledge-Driven Machine Learning Models

It is common wisdom that performance can be improved in various machine learning tasks when incorporating domain-specific knowledge (Bengio et al. 2013), particularly for clustering (Wagstaff et al. 2001), classification (Schölkopf et al. 1998), and topic modeling (Andrzejewski et al. 2009). Moreover, knowledge can be extracted and represented in different formats ranging from common-sense (Li et al. 2016) to domain specific (Pardoe et al. 2010) to ontology-based (Sharman et al. 2004), which is subsequently incorporated into intelligent information systems. Prior work in finance and accounting that uses words from the L&M Dictionary as features in machine learning models can be considered as one way of incorporating domain knowledge (Loughran and McDonald 2011, Bodnaruk et al. 2015). Other researches also propose their own lexicons. For example, Das and Chen (2007) use a human-crafted lexicon to identify sentiment from stock message boards and construct a sentiment index for stocks. Our work adds on to the existing literature in designing intelligent information systems by proposing a novel way to incorporate domain knowledge.

3. Methodology

In this section, we describe our model that incorporates financial domain lexicon into the representation learning framework to obtain domain-specific word embeddings for better stock volatility prediction. Before diving into the details, we first briefly review representation learning in NLP.

3.1. Representation Learning: word2vec

The performance of machine learning algorithms largely depends on data representation (Bengio et al. 2013). In most previous NLP research, text is mainly represented by a bag of words via a one-hot encoding strategy, in which many latent and deep semantics among words and particularly the order or the contextual information of words are usually ignored. Recent breakthrough advances in neural word representation learning, also known as word embeddings, has enabled this approach to become an increasingly important building block in text analysis. Word embedding represents each word as a multi-dimensional vector, where not only individual semantic characteristics of words but also relationships among words are well captured. The word embeddings are trained via a neural language model on a large text dataset where it attempts to predict words given their context (Mikolov et al. 2013). The vector representations are such that words that often co-occur in the same context have similar vectors. In addition, various recently proposed transformer-based (Vaswani et al. 2017) language models such as BERT (Devlin et al. 2019) and GPT 2.0 (Radford et al. 2019) have shown outstanding performance in NLP tasks. They differ from previous word embedding models in that the representation of a word depends on the contextual word representations. However, they suffer from a maximum document length requirement, which may make them unsuitable for long financial documents such as 10-K filings.

In this study, we build our own domain-specific word representation based on the `word2vec` framework due to its wide adoption (Mikolov et al. 2013). Two schemes, Skip-gram (SG) and continuous bag-of-words (CBOW), are introduced in `word2vec`. Since both have very a similar objective function and can achieve nearly the same results, for the sake of simplicity we consider incorporating the domain-specific lexicon into the SG model.

Thus, we briefly review the SG model. Its objective is to maximize the probability of the context (surrounding) words conditioning on a focal (central) word:

$$P(w_{O_1},...,w_{O_T}|w_I)=\prod_{i=1}^{T}P(w_{O_i}|w_I) \tag{1}$$

where w_I is the target/focal word and $w_{O_1},...,w_{O_C}$ are context words in a certain window, with T as the window size. Thus, the objective function can be written as:

$$\mathcal{L}_{W2V}=\frac{1}{|W|}\sum_{w_I\in W}\log P(w_{O_1},...,w_{O_T}|w_I) \tag{2}$$

where W is the set of all word tokens in the corpus.

Each word plays two roles: the word itself and a specific context of other words. We have two vectors $\mathbf{w}^0$ and $\mathbf{w}$, where $\mathbf{w}^0$ is the word embedding of word w when it is treated as a word itself while $\mathbf{w}$ is the representation of word w when treated as a specific context. To approximate the conditional probability $P(w_O|w_I)$ in Eq. 1, a softmax function is used:

$$P(w_O|w_I)=\frac{\exp\left(\mathbf{w}_O\cdot\mathbf{w}_I^0\right)}{\sum_{\mathbf{w}'\in V}\exp\left(\mathbf{w}'\cdot\mathbf{w}_I^0\right)} \tag{3}$$

However, calculating this probability is intractable since the denominator sums over all unique words in the vocabulary. Therefore, a negative sampling strategy is usually used, in which only a small set of words ("negative" words) are sampled as contextual words according to their frequency in the training set[2]. With the negative sampling strategy, the probability of word w_O conditioning on word w_I is now approximated by the product of k sigmoid functions $\sigma(\cdot)$, where k is the size of negative samples:

$$P(w_O|w_I)=\sigma(\mathbf{w}_O\cdot\mathbf{w}_I^0)\prod_{j=1}^{k}\sigma(-\mathbf{w}_j\cdot\mathbf{w}_I^0) \tag{4}$$

Thus, the word2vec objective function can now be re-written as:

$$\begin{aligned}\mathcal{L}_{W2V}&=\frac{1}{|W|}\sum_{w_I\in W}\log P(w_{O_1},...,w_{O_T}|w_I)=\frac{1}{|W|}\sum_{w_I\in W}\sum_{i=1}^{T}\log P(w_{O_i}|w_I)\\&=\frac{1}{|W|}\sum_{w_I\in W}\sum_{i=1}^{T}[\log\sigma(\mathbf{w}_{O_i}\cdot\mathbf{w}_I^0)+\sum_{j=1}^{k}\log\sigma(-\mathbf{w}_j\cdot\mathbf{w}_I^0)]\end{aligned} \tag{5}$$

[2] For details of word2vec training and negative sampling, we refer readers to Mikolov et al. (2013).

3.2. Incorporating a Lexicon with Word Embeddings

The Skip-gram model does not require any human labeling or external domain knowledge, as it is an unsupervised data-driven language model that simply predicts context words given a focal word. However, in many applications, external domain knowledge sources such as hand-crafted domain lexicons are largely available. For example, in the biomedical area, researchers build bio-domain lexicons and ontologies, such as Unified Medical Language System (UMLS), to provide a necessary framework for semantic representation (Spasic et al. 2005). In the financial community, researchers rely on the L&M Dictionary to gauge the tone of annual reports.

Therefore, such domain knowledge provides additional information that may not be easily learned by advanced data-driven models. For example, the word "exonerate" in business communication often occurs in a negative context (e.g., "The measures may not be adequate to exonerate us from relevant civil and other liabilities"). It is relatively less frequent in the 10-K filings (as counted by L&M Dictionary, it appears 3,241 times, accounting for only 1.89×10^{-7} among billions of word tokens in all 10-K/Qs). Using the vanilla SG model, which conducts negative sampling for the sake of efficiency, this word is much less likely to be sampled and will then surely be under-represented in the learned word embedding. To alleviate this issue, we propose an enhanced data-driven approach for representation learning with domain knowledge taken into account. Specifically, word "exonerate" is annotated by the L&M Dictionary in the "Negative" category, and this can be leveraged to guide the SG model to learn a better word representation. In this improved representation, we would expect that "exonerate" to have a similar vector to other negative words in the "Negative" category. Motivated by this, we discuss our method of incorporating the domain lexicon into the original SG model to learn domain-specific word embeddings.

Assume that a lexicon consists of K categories $C_1, C_2, \ldots, C_K$, and each category includes a set of semantically similar words. In the case of the L&M Dictionary, it has several word categories, such as Positive, Negative, Constraining, Superfluous and several others. The words in the same category often occur in the same business context.

Constraint Objective The general idea of incorporating domain lexicons into the SG model is to impose knowledge as extra constraints $\mathcal{L}_C$ on the original `word2vec` objective function $\mathcal{L}_{W2V}$ as follows:

$$\mathcal{L}_{joint} = \mathcal{L}_{W2V} + \lambda \mathcal{L}_C \tag{6}$$

where λ is a hyperparameter controlling the strength of constraints.

The goal of word representation learning is that words that often co-occur in the same context have similar embeddings. That is, embeddings of words that have similar meanings should be close in the embedding space, while those of words with distinct meanings should be far away from each other in the embedding space. Based on this observation and inspired by previous constrained learning tasks, such as constrained clustering (Zhang et al. 2007), we introduce two sets of constraints used when representing lexicons: "must-link" and "cannot-link". Words in the same lexicon category can be regarded as having "must-link" constraints, and words in different lexicon categories have "cannot-link" constraints. For example, since "exonerate" and "acquit" are both annotated in the "Negative" category, we form a must-link between these two words, meaning that their learned word embeddings should be closer. Note that the must-link constraints are transitive. For example, if there is a must-link between word pair (a, b) and word pair (b, c), then (a, c) will implicitly have a must-link, and (a, b, c) shares a transitive closure. Note that these two constraints require an important assumption that word category should be exclusive. By incorporating both must-link and cannot-link constraints, we expect to learn better domain-specific word embeddings, so as to improve downstream prediction tasks. Specifically, the new objective function with constraints incorporated is now formalized as:

$$\mathcal{L}_C = \sum_{\substack{k \neq h \\ k,h=1}}^{K} \beta_C \sum_{\substack{w \in C_k \\ v \in C_h}} \|\mathbf{w}^0 - \mathbf{v}^0\|^2 - \sum_{k=1}^{K} \beta_M \sum_{\substack{w \in C_k \\ v \in C_k}} \|\mathbf{w}^0 - \mathbf{v}^0\|^2 \tag{7}$$

where $\| \cdot \|^2$ is the Euclidean distance; β_C and β_M are hyperparameters controlling the strength of cannot-link and must-link respectively; and $\mathbf{w}^0, \mathbf{v}^0$ are the corresponding embeddings for words w and v. Note that unlike the word2vec model that uses a focal (input) word to maximize the probability of a context (output) word, the above constraint-based objective only includes focal (input) words. This makes the optimization process more tractable. By maximizing this constraint-based objective, we hope to decrease the distance in the embedding space for words in the same category while increasing the distance for words from different categories. We can re-write the overall objective function $\mathcal{L}_{joint}$ as:

$$\begin{aligned}\mathcal{L}_{joint} &= \mathcal{L}_{W2V} + \lambda\mathcal{L}_C \\ &= \frac{1}{|W|}\sum_{w_I \in W}\sum_{i=1}^{T}[\log\sigma(\mathbf{w}_{O_i}\cdot\mathbf{w}_I^0) + \sum_{j=1}^{k}\log\sigma(-\mathbf{w}_j\cdot\mathbf{w}_I^0)] \\ &+ \lambda(\sum_{\substack{k\neq h\\ k,h=1}}^{K}\beta_C\sum_{\substack{w\in C_k\\ v\in C_h}}\|\mathbf{w}^0-\mathbf{v}^0\|^2 - \sum_{k=1}^{K}\beta_M\sum_{\substack{w\in C_k\\ v\in C_k}}\|\mathbf{w}^0-\mathbf{v}^0\|^2)\end{aligned} \tag{8}$$

where λ is a hyperparameter controlling the relative importance of $\mathcal{L}_C$ to $\mathcal{L}_{W2V}$.

Optimization Gradient ascent is often used in solving such an optimization problem, as well as in the original word2vec model. Note that in the SG model, a word w has two word vectors. $\mathbf{w}^0$ (input vector) is the representation of word w when it is treated as a word itself, and it is the word vector of interest. $\mathbf{w}$ (output vector) is the representation of word w when it is treated as a specific context word. These two word vectors are learned jointly so that $\mathbf{w}$ can enhance the representation of $\mathbf{w}^0$. Using gradient ascent, we iteratively update $\mathbf{w}$ and $\mathbf{w}^0$ until the model gradient is converged. We show the gradient ascent updating process for both $\mathbf{w}^0$ and $\mathbf{w}$ below. Specifically, the updating process of a representation vector for a word and its corresponding gradient is:

$$\mathbf{w}_I^0 \leftarrow \mathbf{w}_I^0 + \alpha\frac{\partial\mathcal{L}_{joint}}{\partial\mathbf{w}_I^0} = \mathbf{w}_I^0 + \alpha\frac{\partial\mathcal{L}_{W2V}}{\partial\mathbf{w}_I^0} + \alpha\lambda\frac{\partial\mathcal{L}_C}{\partial\mathbf{w}_I^0} \tag{9}$$

where

$$\frac{\partial\mathcal{L}_{W2V}}{\partial\mathbf{w}_I^0} = \sum_{i=1}^{T}[\sigma(-\mathbf{w}_{O_i}\cdot\mathbf{w}_I^0)\mathbf{w}_{O_i} - \sum_{j=1}^{k}\sigma(\mathbf{w}_j\cdot\mathbf{w}_I^0)\mathbf{w}_j] \tag{10}$$

$$\frac{\partial\mathcal{L}_C}{\partial\mathbf{w}_I^0} = \sum_{\substack{k\neq h\\ k,h=1}}^{K}2\beta_C\sum_{v\in C_h}(\mathbf{w}_I^0-\mathbf{v}^0) - \sum_{k=1}^{K}2\beta_M\sum_{v\in C_k}(\mathbf{w}_I^0-\mathbf{v}^0) \tag{11}$$

The constraint-based objective function $\mathcal{L}_C$ does not involve a word output vector, and only the word2vec objective function $\mathcal{L}_{W2V}$ has a word output vector. Therefore, the updating process for a word's (output) vector is :

$$\mathbf{w}_j \leftarrow \mathbf{w}_j + \alpha\frac{\partial\mathcal{L}_{joint}}{\partial\mathbf{w}_j} = \mathbf{w}_j + \alpha\frac{\partial\mathcal{L}_{W2V}}{\partial\mathbf{w}_j} + \alpha\frac{\partial\mathcal{L}_C}{\partial\mathbf{w}_j} = \mathbf{w}_j + \alpha\frac{\partial\mathcal{L}_{W2V}}{\partial\mathbf{w}_j} \tag{12}$$

Where

$$\frac{\partial \mathcal{L}_{W2V}}{\partial \mathbf{w}_j} = \begin{cases} \sigma(-\mathbf{w}_j \cdot \mathbf{w}_I^0)\mathbf{w}_I^0,\ if\ w_j\ is\ a\ contextual\ word \\ -\sigma(\mathbf{w}_j \cdot \mathbf{w}_I^0)\mathbf{w}_I^0,\ otherwise \end{cases} \tag{13}$$

In our implementation, we use the asynchronized stochastic gradient ascent. Specifically, in each iteration, we sample a word and update its relevant input vector $\mathbf{w}_I$ and output vector $\mathbf{w}_O$, according to Eq. 9 and Eq. 12. We follow the same sampling strategy as the negative sampling for the constraint-based objective because frequent words have been sampled more in the SG model and are more representative for the lexicon categories to which they belong. Optimizing $\mathcal{L}_{joint}$ (Equation 8) makes words in the same lexicon category gravitate together and words in different lexicon categories move towards separation in the embedding space, while still traded off by the objective $\mathcal{L}_{W2V}$. Without loss of generality, we set $\beta_C = \beta_M = 1.0$ so that constraint-based objective weight is determined by λ.

Mathematical Property Our design of a constraint-based objective can be explained based on the mathematical property of word embeddings. Nonlinear word embedding models like `word2vec` and `GloVe` (Pennington et al. 2014) are well known for their good mathematical properties on the learned embeddings. For example, $\mathbf{w}(\text{“}\mathbf{king}\text{”}) - \mathbf{w}(\text{“}\mathbf{man}\text{”}) + \mathbf{w}(\text{“}\mathbf{woman}\text{”}) = \mathbf{w}(\text{“}\mathbf{queen}\text{”})$. Most of models can be traced back to the fact that the inner product of two word vectors is approximately the pointwise mutual information (PMI) of these two words up to some shift (Arora et al. 2016).

$$\mathbf{w} \cdot \mathbf{v} \approx PMI(w, v) = \log \frac{P(w|v)}{P(w)} \tag{14}$$

Therefore, for a context word c, we have

$$\mathbf{c} \cdot (\mathbf{w_1} - \mathbf{w_2}) \approx \log \frac{P(c|w_1)}{P(c|w_2)} \tag{15}$$

If two words w_1 and w_2 have similar semantic meanings, then for every context word c, $P(c|w_1)$ and $P(c|w_2)$ should be considerably similar too. Since the value of $P(c|w_1)$ and $P(c|w_2)$ are close, $\frac{P(c|w_1)}{P(c|w_2)}$ is close to 1, which means that

$$\mathbf{c} \cdot (\mathbf{w_1} - \mathbf{w_2}) = \|\mathbf{v}_c\| \|\mathbf{w_1} - \mathbf{w_2}\| \cos\theta_c \approx 0 \tag{16}$$

where θ_c is the angle between $\mathbf{c}$ and $\mathbf{w_1} - \mathbf{w_2}$. Since this equation holds for every contextual word c, $\|\mathbf{w_1} - \mathbf{w_2}\|$ must be considerably small. If w_1 and w_2 are very semantically different, then $\frac{P(c|w_1)}{P(c|w_2)}$ is not close to 1, meaning that $\|\mathbf{w_1} - \mathbf{w_2}\|$ is not small. Therefore, through our constraint-based objective (Equation 7), we simply enforce this mathematical property by making the embeddings of semantically similar words closer in the embedding space while making the embeddings of semantically distinct words farther apart.

3.3. Stock Volatility Prediction Using Annual Reports

Once domain-specific embeddings are learned, they can be subsequently used for representing each firm annual report. This is referred to as document embedding, where each report is represented as a vector in the same dimension as learned word embeddings.

Our volatility prediction time horizon is one year, meaning that we use corporate annual reports to predict the annual volatility for the subsequent year. Annual volatility prediction is valuable for investors whose investment horizon is from months to years. First, annual volatility is associated with annual risk and uncertainty, the key attributes in investing, option pricing and risk management (Poon and Granger 2005). For example, the risk-neutral valuation principle established by Black and Scholes means that the mean return on the stock is irrelevant and volatility is the most important factor in determining option prices. Therefore, annual volatility prediction allows investors to assess the value over the option's maturity (one year). Second, it is also possible to trade volatility directly, known as volatility arbitrage. To conduct volatility arbitrage, a trader must first forecast the underlying security's future volatility. In this context, predicting annual volatility is crucial for finance market participants.

We use subscript t to denote year t and superscript i to denote firm i. For example, firm i's annual report in year t is denoted as d_t^i. To simplify notations, the superscript is dropped without explicit mention and it is understood that all terms now refer to firm i. Therefore, d_t^i is simplified as d_t.

A common document embedding method in NLP is the tf-idf weighted average over embeddings of all words (Arora et al. 2017). This simple document embedding strategy is proven to be very effective in preserving document semantics. We use this method for firm annual report representation. Given a 10-K document d_t in which Section Item 1A contains a set of words $\{w_1, w_2, \dots, w_n\}$, we use tf-idf weighted average to obtain a vector representation $\mathbf{x}_t$ for d_t:

$$\mathbf{x}_t = \frac{1}{n}\sum_{k=1}^{n} tfidf(w_k, d_t)\mathbf{w_k}, \qquad \forall w_k \in d_t \tag{17}$$

where $tfidf(w_k, d_t)$ represents the standard tf-idf score of word w_k in document d_t. As before, $\mathbf{w_k}$ is the embedding vector of word w_k.

To forecast a stock's volatility, we follow a framework similar to that of Kogan et al. (2009) and formulate the prediction problem as a supervised learning task, where the past seven years' 10-K filings are used to learn word embeddings and consequently annual report representation and the volatility of the following year is the target, as illustrated in Figure 1. Each pair (x_t, y_t) is a training instance, where x_t is the embedding representation of an annual report in year t and y_t is the volatility in the following year $t+1$. Knowledge-enhanced word embeddings are first learned using reports in the training period. For example, reports from year 2004 through 2010 together with their corresponding stock volatilities (2005 to 2011) constitute our training set. The learned model can predict volatility for 2012 using the report from 2011. Our model is expected to implicitly learn the latent relationship between volatility and key words in 10-K filings.

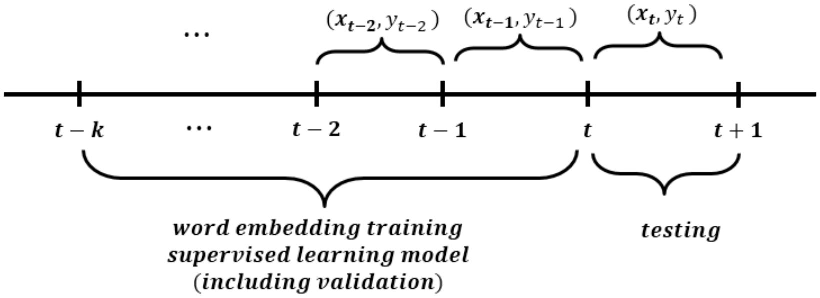


Figure 1 Volatility prediction timeline.

Our prediction target is firm i's stock return volatility covering the time range between year t and year $t+1$. We denote the annual stock volatility as $y_{[t,t+1]}$. To simplify notation, we shorthand it as y_t. The annual stock volatility is defined as:

$$y_t = \sqrt{\frac{\sum_{d=0}^{\tau}(r_d - \bar{r})^2}{\tau}} \tag{18}$$

where r_d is the return price at day d and $\bar{r}$ is the mean of the return price during period $[t, t+1]$. The return price is defined as $r_d = \frac{P_d}{P_{d-1}} - 1$, where P_d is the close price on day d. We set $\tau = 252$, the number of trading days in a year. Here, we use fiscal year instead of calendar year to calculate the annual volatility. While there are other ways to measure annual stock return volatility (e.g., by averaging volatilities over the four quarters in a year), we choose the standard annual volatility as it is the most straightforward measure.

The supervised machine learning task can therefore be formulated as: given $\mathbf{x}_t$ is the annual report vector for firm i in the t-th year, our prediction target y_t is the volatility of the firm in the following year after the t-th year annual report is released. We apply Support Vector Regression (SVR) (Drucker et al. 1997) as the supervised learning model. SVR formulates the training process as the following optimization problem:

$$\min_{\boldsymbol{\beta}} \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{L}\sum_{j=1}^{L} \max(0, \|y_j - f(\mathbf{x}_j; \boldsymbol{\beta})\| - \epsilon) \tag{19}$$

where L is training set size and C is the regularization parameter. Similar to previous studies (Kogan et al. 2009, Rekabsaz et al. 2017, Tsai and Wang 2014), we set C and ϵ to 1.0 and 0.1, respectively, and we use the Radial Basis Function (RBF) kernel in SVC.

It is worth noting that an end-to-end deep learning model such as convolutional neural network (CNN) or long short-term memory network (LSTM) can directly use word embeddings as input for supervised learning tasks without having to represent annual reports in vectors. We use SVR simply because it has been used in prior research (Kogan et al. 2009) and thus allows us to make a fair comparison between our study and existing work. In the experiment section, we will evaluate whether the benefit of high-quality domain-specific word embeddings will vanish if we directly apply deep learning models.

The entire process of our stock volatility prediction using firms' 10-K filings is described in pseudo-code in Algorithm 1. We also summarize notations in Table 1.

Algorithm 1: Stock volatility prediction.

```
Goal: Given firms' 10-K annual reports for the $k$-th fiscal year, predict the stock
      volatility of the following fiscal year.
Data: 10-K filings in previous years: $D = \bigcup d_t^i$, and the annual volatility: $y = \bigcup y_t^i$
      where $i \in N$ and $t \in T$; and L&M Dictionary
Learn (optimize Eq. 8) finance-specific word embeddings $\mathbf{w}$ for word $w \in V$ using $D$
  and L&M Dictionary;
foreach firm $i$ do
    foreach year $t$ do
        represent $d_t^i$ as a vector $\mathbf{x}_t$ using Eq. 17
Train a supervised learning (regression) model $f$ using $(\mathbf{x}_t,\ y_t)$ (Eq. 19 );
Represent the $k$-th year annual report $d_k$ as $\mathbf{x_k}$, and predict volatility during year
  $[k, k+1]$ using $f$, i.e., $y_k = f(\mathbf{x}_k)$;
```

Parameter	Description
$\lambda, \beta_C, \beta_M$	hyper-parameters in the constraint-based objective
$C_1, C_2, ...C_K$	K categories of words in the domain lexicon
w, v	word w, v
$\mathbf{w}, \mathbf{v}$	embedding representation of word w, v
T	number of years
N	number of firms
$D = \bigcup d_t^i$	set of annual reports where $i \in N, t \in T$
d_t^i, d_t	firm i's annual report for the t-th year
W	all word tokens in D
V	all words in the vocabulary
$\mathbf{x_t^i}, \mathbf{x_t}$	the embedding representation of firm i's annual report in the t-th year
$y_{[t,t+1]}, y_t$	the volatility of firm i during year t and year $t+1$
$\mathcal{L}_{W2V}, \mathcal{L}_C$	objective function of word2vec embedding and constraint-based embedding

Table 1 **Summary of notations used in the paper.**

4. Data

The data used in this study are annual reports that are publicly available and are in fact the only firm disclosure required on a regular basis. In the U.S., the Securities and Exchange Commission (SEC) mandates that all publicly traded companies file annual reports, known as **Form 10-K**. This document provides a comprehensive overview of the company's business and financial condition to their shareholders and the general public. The Form 10-Ks are audited and documented by an approved accountancy firm. Once released, they can be accessed publicly via the EDGAR SEC system[3]. Other text documents are either voluntarily filed by companies, such as earnings announcements, or produced by company outsiders, such as analyst reports and internet discussion boards. We do not use quarterly reports (e.g., Form 10-Q) in our analysis because in these filings, firms are only required to disclose material changes since the last annual report; thus quarterly filings contain only an incomplete set of information[4].

A typical Form 10-K has many sections that give an overview of the company's business and financial condition. Among these, Section *Item 1A - Risk Factors* is very important because it discusses potential risks that might affect the business. For example, an offshore oil drilling company may list the risk of losses from a major oil spill accident. Since 2005, the SEC has required all publicly traded firms to include a separate section in their Form 10-K to discuss "the most significant factors that make the company speculative or risky" (Regulation S-K, Item 305(c), SEC 2005). This section has turned out to be one of the most examined segments of corporate annual reports (Bao and Datta 2014), and these risk disclosures are considered important investing factors for stakeholders and the general public. Prior research suggests that 10-K reports are long and redundant, and only the *Risk Factors* section appears to be informative to investors (Dyer et al. 2017). Thus in this paper, we focus on volatility prediction mainly using Section *Item 1A - Risk Factors* of Form 10-K.

Data sample: Russell 3000 Companies. We choose Russell 3000 constituent firms as our target for volatility prediction due to their importance and tractability. This index includes the 3,000 largest publicly held companies incorporated in the U.S. as measured by total market capitalization, and it represents approximately 98% of the U.S. public equity

[3] `http://www.sec.gov/edgar.shtml`

[4] `https://www.sec.gov/fast-answers/answersform10qhtm.html`

	training year	# training samples	# tokens	test year	# test samples
1	2004-2010	19,234	308M	2011	2,804
2	2005-2011	19,325	336M	2012	2,832
3	2006-2012	19,428	362M	2013	2,896
4	2007-2013	19,602	390M	2014	2,997
5	2008-2014	19,860	419M	2015	3,007
6	2009-2015	20,102	449M	2016	2,998

Table 2 Form 10-Ks for stock volatility prediction.

market. Our primary source of text data is the SEC Electronic Data Gathering, Analysis, and Retrieval (EDGAR) system, where we can download companies' Form 10-Ks. Our data samples include 36,768 Form 10-Ks from Russell 3000 companies filed between 2004 and 2016. Each report comes with a fiscal year identifier so that we can correlate the report to the stock volatility of the following year.

We create a Python parser to extract the Risk Factors section from downloaded Form 10-Ks (in HTML format). To improve the quality of our data, we perform several pre-processing operations, such as punctuation/stopword removal and lowercase conversion. All documents with fewer than three sentences and all sentences with fewer than five words are ignored. We do not apply word stemming or word lemmatization as it is not required for the representation learning. We keep text pre-processing efforts minimal so that our approach is easy to reproduce and scale. In our dataset, although the size of each training set is relatively small (about 20,000 documents), the total number of word tokens is very large, since each 10-K document has massive amounts of text. For example, the training set contains 449 million and 308 million word tokens for the period 2009-2015 and the period 2004-2010, respectively. We also obtain daily stock prices for 2004-2017 (dividend-adjusted) from the CRSP database so that we can calculate annual volatility. The details of our dataset are shown in Table 2.

L&M Dictionary Economists and finance researchers have hand-crafted finance-specific lexicons to analyze the semantic and sentiment content of financial reports. The most widely used lexicon is the Loughran and McDonald Word List **(L&M Dictionary)** (Loughran and McDonald 2011). This dictionary contains words used by managers in Form 10-Ks, and these words are separated into different categories (e.g., Positive, Negative, Uncertainty, Litigious, Constraining, and Superfluous). A large body of finance and accounting research (see the survey paper by Loughran and McDonald (2016)) has used the L&M dictionary to gauge the tone and semantics of Form 10-Ks. As noted in Kearney and Liu (2014), the L&M Dictionary has become “predominant in more recent studies.” An important feature of the L&M Dictionary is that word categorization is grounded in the financial context; for example the word “restatement”[5] is generally more negative when used in finance than in plain English.

For learning domain-specific word embeddings, we only consider two risk-related categories: Negative and Positive. They have 2,355 and 354 distinct words, respectively. The most frequent words in the Negative category include *wrongly*, *wrongdoing*, *writeoff*, *writedown*, *worthless*, *worst*, *worsen*, *worrying*, *willfully* and *weakness*. The top ten most frequently occuring words in the Positive category include *worthy*, *winning*, *vibrant*, *versatile*, *valuable*, *upturn*, *unsurpassed*, *unparalleled*, *unmatched* and *tremendously*. We calculated the ratio of the word tokens from the Positive/Negative categories in the L&M dictionary to the total word tokens in the 10-K reports (section 1A) for each year. The positive-to-total word ratio and negative-to-total word ratio are quite consistent over the years. The positive-to-total word ratio is 0.85% on average, with a standard deviation of 0.03%. The negative word ratio is 2.98%, with a standard deviation of 0.44%.

5. Experiments

In this section, we compare the performance of our proposed model **LM-WE** (L&M Dictionary Enhanced Word Embeddings) with several state-of-the-art approaches that we use as baselines for stock volatility prediction.

5.1. Baselines

The baselines, mainly from the fields of machine learning and financial economics, can be broadly grouped as:

- **Bag-of-Words Baselines**: A large body of prior work on finance disclosure text analysis uses bag-of-words models to represent documents. These models ignore much important information, such as syntactics, semantics, and the order of words. Each word is a V-dimensional one-hot encoding vector, where V is the number of distinct words in the vocabulary. The vocabulary contains all words from the text corpus (Li 2010, Kogan et al.

[5] A restatement is the revision of one or more of a company’s previous financial statements. The issue of a restatement often causes significant moves in stock prices.

2009). We refer to this baseline as **BOW**. Similarly, instead of using all words, some studies only consider words from the L&M Dictionary to construct the vocabulary (Loughran and McDonald 2011, Bodnaruk et al. 2015, Tsai and Wang 2014, Rekabsaz et al. 2017). We denote this bag-of-words variant as **LM-BOW**. The feature space of LM-BOW is 2,709 because the L&M lexicon contains 2,355 unique negative words and 354 unique positive words.

- **Topic Modeling Baselines**: Topic modeling, such as Latent Dirichlet Allocation (LDA), is an unsupervised text mining technique that uncovers underlying topics in a document collection based on the word co-occurrence statistics (Blei et al. 2003). There is a burgeoning stream of literature in finance and accounting that applies LDA for extracting latent topics and representing financial documents (Bao and Datta 2014, Huang et al. 2017). LDA can represent each document as a mixture of topics, which is a multinomial distribution over hidden topics. We use the standard LDA to generate features for volatility prediction (this baseline is denoted as **LDA**). Moreover, some studies point out that a standard LDA model may not work well for long documents (Büschken and Allenby 2016, Huang et al. 2017). As one LDA variant, researchers assign one topic label to each sentence and aggregate sentence labels to obtain representation for an entire document. We implement this LDA variant and name it **Sent-LDA**. In our experiment, we set the number of topics at 50 as this number achieves the best model fitness on the validation set.

- **Word Embedding Baselines**: Our domain-specific word embeddings are learned based on the popular word2vec framework. Vanilla word2vec, specifically the SG model, should obviously be considered as a baseline. We denote it as **word2vec**.

In addition to word2vec, we also consider another advanced neural word embedding framework **GloVe** (Pennington et al. 2014). Unlike word2vec, which aims to capture word co-occurrence in a fixed window, GloVe takes a more global design perspective to build and factorize the co-occurrence matrix for the entire corpus.

A comparable baseline to our work is **retrofit** word embedding (Faruqui et al. 2015), which attempts to refine word2vec word embedding with lexicons. However, the domain knowledge is incorporated in a post-processing manner. That is, the word embeddings and lexicons are *not* jointly learned during the representation learning. Embeddings are first learned, and then retrofit uses lexicons to adjust words' location in the embedding space.

Further, retrofit only considers word association relations (similar to our must-links). While retrofit is conceptually close to our model, it is worth emphasizing that it takes a very different post-processing approach in which the only objective in the refinement process is meeting the must-link constraints. As a result, it tends to learn over-clustered word vectors. On the other hand, in our proposed LM-WE method, the word vectors are jointly updated with the word2vec objective and domain-lexicon knowledge. To have a fair comparison, in our experiments, we first generate a set of must-links from the LM Dictionary by sampling words from Negative and Positive categories respectively. We then incorporate the same set of must-links in LM-WE (our method) and in retrofitting. All word embeddings have a dimension of 200.

To generate document representations in these word embedding baselines, we use the same document embedding strategy, discussed in Section 3.3 (Eq. 17). This allows us to fairly compare the quality of different word embeddings and attribute the improvement in volatility prediction to the incorporation of the financial domain lexicon.

- **Word Embedding and Bag-of-Words Baselines:** Another straightforward way of incorporating domain knowledge is to directly combine word embeddings and the L&M Dictionary. That is, a document is represented as a concatenation of word embeddings and the L&M gag-of-Words features. Thus, we implement this simple yet straightforward baseline. Specifically, we obtain vanilla word2vec embeddings from the SG model and use the tf-idf to obtain L&M-BOW word features. Note that the dimensionality of word2vec is smaller than that of L&M-BOW as the L&M lexicon contains 2,709 unique words. We apply the PCA dimension reduction method on the L&M feature space, but experiment results show that there are no significant differences when we do not apply PCA. Thus, we report experiment results of the model without PCA and denote it as **w2v+LM-BOW**.

- **Transform-Based Contextual Word Embedding Baselines:** Recently, various transformer-based (Vaswani et al. 2017) language models such as BERT (Devlin et al. 2019) and GPT 2.0 (Radford et al. 2019) have been proposed and have shown outstanding performance in various NLP tasks. These recent transformer-based language models differ from previous word embedding models in that the representation of a word depends on the contextual word representations. In this study, we implement two BERT-based baselines in our experiments. The first one is **BERT fine-tuned**, for which we first obtain the pre-trained BERT model (BERT-large, uncased) that is trained using a large corpus of

Wikipedia and BookCorpus. We then fine-tune the model on our dataset. The second one is **BERT full**, for which we train new BERT models from scratch using our training set corpus. Our training method follows the original BERT paper (Devlin et al. 2019). Note that training a new BERT model from scratch is very computationally expensive. Specifically, the entire training is conducted using an NVIDIA DGX-1 machine with 4 Tesla P100 GPUs, providing a total of 128 GB GPU memory. The total time for training a BERT-full model is approximately 3 days. Since our volatility prediction is a regression task, we use BERT's output as input to a linear layer, then use SVR that performs the regression prediction. Since BERT cannot take text longer than the maximum length (512 tokens), for both BERT fine-tune and BERT full baseline we truncate input documents to the maximum sequence length, as suggested by the original BERT paper (Devlin et al. 2019).

- **Past Volatility $\mathbf{v^{past}}$**: In addition to machine learning baselines, we also include a conventional volatility predictor widely used in the field of financial economics. It is often reported in prior research that past volatility is a strong predictor of future volatility. Thus, we consider using the volatility of the $(i-1)$-th year to predict the volatility of the i-th year. We call this baseline $\mathbf{v^{past}}$.

5.2. Metrics

We use two standard metrics to report the prediction performance: Mean Absolute Error (MAE) and Mean Squared Error (MSE), as defined below. Both metrics calculate the errors between the predicted volatility and true volatility, but MSE differs from MAE as it tends to punish large errors more, and MAE is easier to interpret.

$$MAE = \frac{1}{M}\sum_{i=1}^{M} |\ln f(\mathbf{x}_k) - \ln y'_k|, \qquad MSE = \frac{1}{M}\sum_{i=1}^{M}(\ln f(\mathbf{x}_k) - \ln y'_k)^2 \tag{20}$$

where M is the size of testing set, y'_k is the true volatility for the testing example $\mathbf{x}_k$, and $f(\mathbf{x}_k)$ is the predicted volatility. We report the number in a log format because it is a standard in finance as the distribution of log-volatility across firms tends to have a bell shape (Kogan et al. 2009). We repeat experiments ten times under different random seeds and report the mean and the significance level. All experiments are performed on a Nvidia P100 GPU, with Tensorflow r1.12 as the backend framework.

5.3. Results

We now turn to presenting and discussing empirical results, and in so doing we address the following questions:

- **Q1** How does **LM-WE** perform compared with state-of-the-art baseline models?
- **Q2** Which knowledge is more effective for stock volatility prediction?
- **Q3** How are word embeddings changed when lexicons are incorporated?
- **Q4** Are lexicon-enhanced finance-specific embeddings still necessary when deep learning models are applied for volatility prediction?

Overall performance (Q1) Table 3 and Table 4 show the results of comparison to the existing state-of-the-art models. The best one is in bold. Statistical significance is reported as compared to word2vec results under a one-tailed t-test ($***$ indicates $p \leq 0.001$) on the averaged result. The column denotes the year of Form 10-K that we use for predicting the next year's stock volatility, from which we can clearly observe that the proposed **LM-WE** model performs the best on both metrics for all years. Overall, our proposed model LM-WE model, by leveraging representation learning and external domain knowledge, significantly outperforms the baselines, which demonstrates the superiority of our model. Specifically, for MAE, LM-WE reduces the prediction error over v_{past} (14.3%), BOW(22.2%), LDA(31.3%), and vanilla word2vec (6.5%), respectively. Also, compared with a retrofit that incorporates lexicons into word2vec, LM-WE reduces the error by 11.9%.

The performance improvement offered by our proposed model can be attributed to the fact that word embedding and lexicons are jointly learned during training and to the incorporation of two different types of constraints from the domain lexicon. For MSE, LM-WE reduces the prediction error over v_{past} (16.3%), BOW(45.5%), LDA(68.5%), and vanilla word2vec (10.6%), respectively. Both transformer-based baselines, BERT fine-tune and BERT full, perform quite poorly in the experiments, mostly due to the long 10-K forms being truncated to meet the maximum length requirement. Although we can increase the maximum length parameter in training the BERT full model, the computational complexity increases quadratically with the length, which would result in an unacceptable training time. Therefore, we believe that incorporating domain lexicons with the simple word2vec model is an effective and efficient solution for financial text analysis.

It is difficult to directly translate our suggested improvement into terms of economic value, since this value depends on many factors such as specific trading strategy, trading volume and transaction cost. However, we note that incorporating knowledge with word embedding training can be done very efficiently, so any additional cost of using LM-enhanced word embeddings would be minimal.

We also note that bag-of-words models (BOW and LM-BOW) perform worse than the approach using past volatility. This result indicates that a simple one-hot encoding method does not capture the word association. This maybe because, especially since Form 10-K is lengthy in language, such a simple learning model fails to learn meaningful patterns that are relevant to volatility. Moreover, we find that topic models (LDA and Sent-LDA) perform the worst among all models, indicating that sentiment related words (positive or negative) contribute to the firm stock volatility variations. This is reasonable because topic models are designed to capture hidden topics in document collection rather than identifying key sentiment. It is possible that the Form 10-Ks of two different firms discuss similar risk topics (and thus have similar document representations in terms of topics), but have very different sentiments. Moreover, experiment results show that w2v+LM-BOW performs consistently better than LM-BOW alone but underperforms the word2vec baseline. This result may imply that LM-BOW features do not provide additional information relevant for word2vec representation. It may also imply that the increased feature space may lead to poor generalizability given our training set only contains seven years' worth of data of including about 20,000 training examples.

Effectiveness of various knowledge sources (Q2) We investigate deeply to understand how the prediction performance varies as different types of constraints are incorporated. To do so, we design experiments under four different situations: (1) no constraints are incorporated, (2) only “must-link” constraints are incorporated, (3) only “cannot-link” constraints are incorporated, and (4) both “must-link” and “cannot-link” constraints are incorporated. Note that the “no constraints” situation is exactly the same as vanilla word2vec. To understand the interaction between the constraint type and the weight of constraint-based objective λ, we choose to vary λ.

Results (see Figure 2) show that first, the weight of constraint-based objective λ has an unneglectable impact on prediction performance. When λ is small (such as 10x10^{-5}), there is no, if any, effect on MAE of volatility prediction, regardless of incorporated constraint types. However, when the λ is large, there is a large negative impact on the model performance, as MAE becomes larger. The reason is that a large λ forces the objective function to give a higher weight towards the constraint-based objective, and starts to ignore word embedding objective which aims to learn word association in text. For different tries in our experiment, the value of λ is ideal at 5x10^{-4}.

		2011	2012	2013	2014	2015	2016	Average
Econ strategy	v^{past}	0.284	0.213	0.176	0.206	0.177	0.251	0.217
Bag-of-Words	BOW	0.295	0.239	0.208	0.223	0.191	0.280	0.239
	LM-BOW	0.278	0.222	0.189	0.213	0.178	0.269	0.225
Topic modeling	LDA	0.310	0.286	0.225	0.291	0.209	0.307	0.271
	Sent-LDA	0.298	0.282	0.231	0.287	0.205	0.292	0.266
Word embedding	word2vec	0.269	0.196	0.164	0.194	0.161	0.215	0.199
	GloVe	0.251	0.198	**0.160**	0.182	0.157	0.225	0.196
	retrofit	0.279	0.218	0.170	0.182	0.162	0.241	0.209
	w2v+LM-BOW	0.272	0.203	0.185	0.197	0.163	0.258	0.213
Transformer-based model	BERT fine-tune	0.281	0.225	0.192	0.206	0.176	0.228	0.219
	BERT full	0.263	0.216	0.187	0.204	0.176	0.215	0.210
LM-WE		**0.249**	**0.185**	0.161	**0.175**	**0.148**	**0.198**	**0.186*******

Table 3 Comparison of different models on MAE.

		2011	2012	2013	2014	2015	2016	Average
Econ strategy	v^{past}	0.117	0.074	0.058	0.077	0.055	0.100	0.080
Bag-of-Words	BOW	0.172	0.166	0.143	0.096	0.072	0.092	0.123
	LM-BOW	0.129	0.098	0.081	0.098	0.108	0.090	0.100
Topic modeling	LDA	0.218	0.189	0.180	0.232	0.193	0.267	0.213
	Sent-LDA	0.205	0.197	0.204	0.221	0.189	0.253	0.212
Word embedding	word2vec	0.107	0.063	0.058	0.072	0.059	0.090	0.075
	GloVe	0.108	0.069	0.052	0.075	0.059	0.091	0.076
	retrofit	0.118	0.078	0.077	0.090	0.071	0.095	0.088
	w2v+LM-BOW	0.112	0.083	0.064	0.076	0.075	0.087	0.083
Transformer-based model	BERT fine-tune	0.124	0.082	0.086	0.081	0.072	0.113	0.093
	BERT full	0.115	0.082	0.088	0.080	0.079	0.108	0.092
LM-WE		**0.104**	**0.052**	**0.052**	**0.067**	**0.050**	**0.076**	**0.067*******

Table 4 Comparison of different models on MSE.

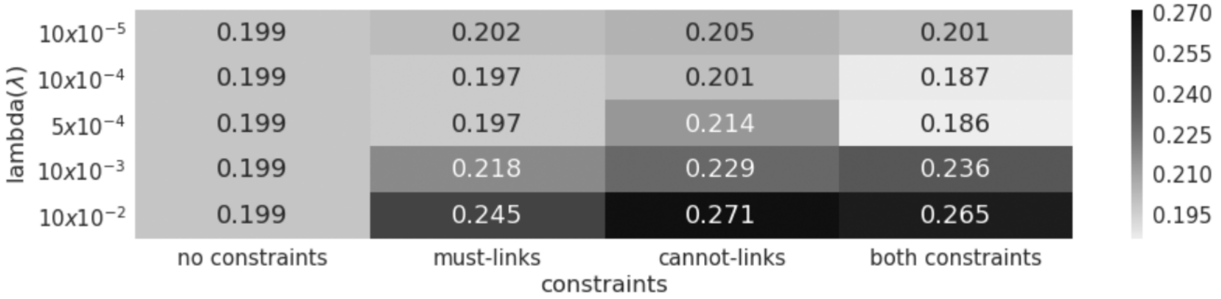
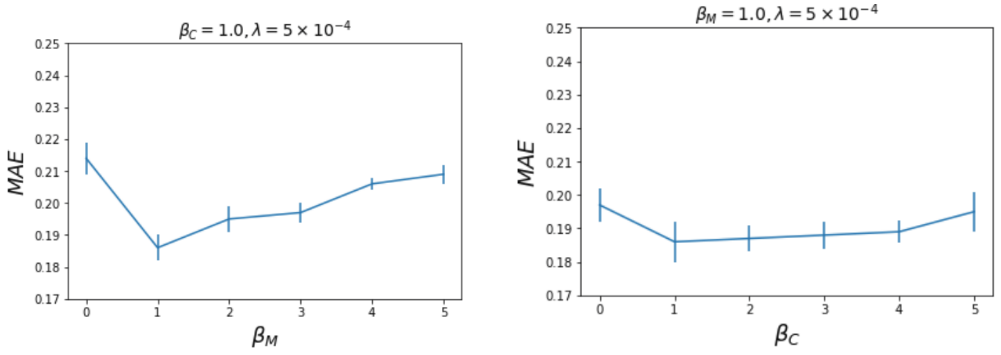


Figure 2 **Different types of constraints are incorporated with domain-specific word embedding learning. MAE is reported here. Lighter color indicates a lower MAE value.**

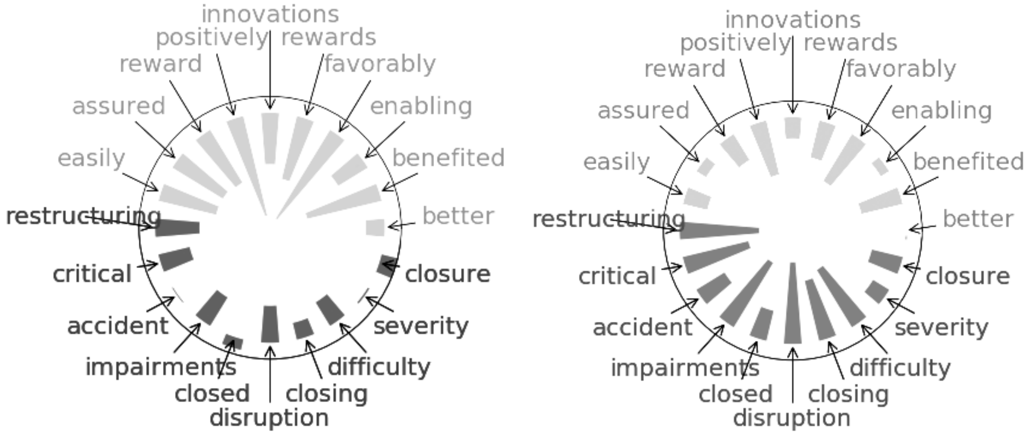


(a) Varying value of $beta_M$ (b) Varying value of $beta_C$

Figure 3 **Parameter sensitivity analysis by changing β_M and β_C.**

Second, adding only must-link or cannot-link constraints in general does not significantly improve the quality of word embeddings. Must-links tend to make words in the same category move closer in the embedding space, and this does not guarantee that words in different categories move further away in the embedding space. Similarly, cannot-links ensure that words in different categories will separate from each other in the embedding space, but as the constraint-based objective takes a batch of word pairs in the training set and does not optimize *all* word pairs at once, words in different categories are also likely to be pushed closer to each other in the embedding space.

In addition, we conduct sensitivity analysis on two hyperparameters, β_M and β_C. β_M controls the strength of must-link objectives, while β_C controls the strength of cannot-link objectives. We present the results in Figure 3. We observe that setting either β_M or β_C to 0 achieves the highest MAE. Therefore, it is important to consider must-links and cannot-links together. Both experiments guide us towards a better understanding of how prediction performance varies as different types of constraints are incorporated.



(a) benefit (b) loss

Figure 4 Change of Euclidean distances to (a) "benefit" and (b) "loss" after incorporating lexicon knowledge. Light color at the top denotes positive words, and dark color at the bottom denotes negative words.

Change of word embeddings (Q3) The main idea behind our domain-specific word embedding framework is that we intend to improve word embeddings to better capture the semantics encoded in text by incorporating domain lexicons that serve as external knowledge. That being said, the geometric locations of words in the embedding space are expected to change after lexicon knowledge is incorporated. To examine this effect, we visualize the change of Euclidean distances in the embedding space. Specifically, we measure the distance of word a to word b before and after the lexicon is incorporated using the following equation:

$$\|\boldsymbol{v}^a_{\text{LM-WE}} - \boldsymbol{v}^b_{\text{LM-WE}}\| - \|\boldsymbol{v}^a_{\text{word2vec}} - \boldsymbol{v}^b_{\text{word2vec}}\| \qquad (21)$$

The first term and second term measure the Euclidean distance between two words learned via LM-WE and vanilla word2vec methods, respectively. We choose two words, "benefit" and "loss," which are the most frequent words in the Positive category and in the Negative category, respectively in the L&M dictionary. Then we randomly pick other 10 Positive words and 10 Negative words from the L&M dictionary and compute their distances to these two reference words using Eq. (21).

Results in Figure 4 show that (1) the distance of each word to the reference word decreases as the direction is pointing inward[6]. (2) The positive words (light color at the top) become much closer to the reference word "benefit" than the negative words (see Figure

[6] The thickness indicates the distance.

4(a)), while the negative words (dark color at the bottom) become much closer to the reference word "loss" than the positive words (see Figure 4(b)). (3) The Euclidean distances between negative words and "benefit" decrease after the lexicon is incorporated. We want to emphasize that this does *not* mean that negative words and "benefit" are similar enough. By examining Figure 4(b), we can see that the Euclidean distances between negative words and "loss" decrease even more significantly after the lexicon is incorporated. This in fact indicates that the negative words become relatively more distant from "benefit," and positive words become relatively more distant from "loss." Therefore, words in the same category become closer while words in different categories become further away in the embedding space. This is consistent with our objective function.

Deep learning or domain knowledge? (Q4) For all previous experiments, we used support vector regression (SVR) as our underlying supervised learning model, as described in Section 3.3. SVR with RBF kernels is often regarded as a shallow machine learning model. Since deep learning models achieve dominant performance in many tasks and can capture more sophisticated non-linear patterns in data, we raise an obvious question: do we really need domain-specific word embeddings if we opt to use deep learning models? And will the benefit of deep learning models overwhelm the benefit of incorporating domain knowledge? Therefore, we design another experiment where we develop several popular supervised machine learning models.

- Logistic Regression (LR). We simply leverage word embeddings as document representations. The LR model is trained with L1 (LASSO) regularization.
- Support Vector Regression (SVR). This is the model we used in previous experiments. The details are described in Section 3.3.
- Convolutional Neural Networks (CNN). Convolutional neural networks are very commonly used in computer vision and have recently been applied to NLP tasks such as document classification. We adopt a similar neural network architecture as in (Kim 2014).
- Bi-directional LSTM (Bi-LSTM). Recurrent neural networks such as LSTM have been widely adopted for various NLP tasks. In particular, bi-directional LSTM becomes one standard baseline for document classification (Adhikari et al. 2019).
- Hierarchical Attention Network (HAN). This is one of the state-of-the-art deep neural networks for document classification (Yang et al. 2016). Unlike traditional recurrent neural networks, it uses an attention mechanism (Bahdanau et al. 2014) so that input words

	MAE		MSE	
	word2vec	LM-WE	word2vec	LM-WE
LR	0.241	0.235	0.143	0.110
SVR	0.199	0.186	0.075	0.067
CNN	0.212	0.227	0.124	0.098
Bi-LSTM	0.187	0.173	0.070	0.061
HAN	0.192	0.178	0.073	0.062

Table 5 Performance comparison of volatility prediction, using shallow machine learning models (LR, SVR) vs. deep learning models (CNN, Bi-LSTM, HAN).

are assigned different weights to form the hidden output vector. It assumes a hierarchical structure (word-sentence-document) when representing a document. We replicate this architectural design as reported in the original paper.

The original deep neural network models (CNN, Bi-LSTM and HAN) are designed for classification while the present study is a numerical volatility prediction (regression). To this end, we modify the last layer of those neural networks, replacing the softmax function with a linear activation function and changing the output dimension from K (suppose there are K classes) to 1, since volatility is a scalar. And we change the cross entropy loss to mean squared error loss. Sequences of word embeddings are used as inputs to neural network models. To ensure a fair comparison, we tune the hyperparameters for all baseline models in the validation set.

Results reported in Table 5 show that LM-WE obtains a lower MAE than word2vec regardless of underlying learning models, which further confirms that domain-specific word embeddings perform better than vanilla word embeddings. In addition, lexicon-specific embeddings are helpful even when deep learning models are used. Specifically, SVR achieves about 6.53% MAE reduction (0.199 vs. 0.186) when using LM-WE, and the best-performing Bi-LSTM model achieves up to 7.48% (0.187 vs. 0.173) MAE reduction when the lexicon is incorporated.

5.4. Transparent Volatility Prediction

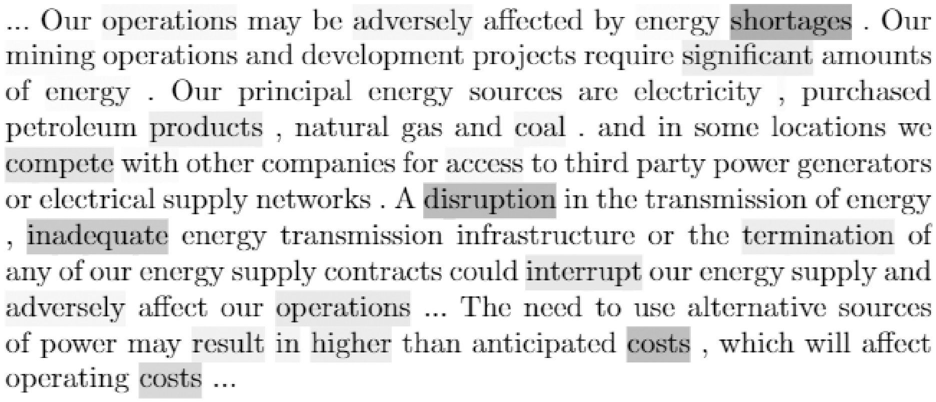
To make the model more useful in practice, especially in the financial decision support system, we need to uncover the model and make it more transparent for the purpose of interpretability. For example, financial analysts are generally necessary to find out which words in an annual report are highly correlated with the risk outcome. That being said, understanding the impact of words on stock volatility prediction is important. In this study, we use the Hierarchical Attention Model (HAN) as the underlying supervised prediction model to measure the impact of each key word on volatility prediction. Note that HAN is one of the deep learning models used in the previous experiment to compare our proposed model with shallow machine learning models.

We choose HAN for two reasons. First, HAN uses an attention mechanism (Bahdanau et al. 2014) and focuses on input words that are relevant to output (therefore referred to as “attention”). Unlike other models (SVR, CNN, Bi-LSTM) that lack transparency in prediction, HAN provides a clear way to understand the contribution of words to stock volatility prediction. Second, other linear models such as Linear Regression also provide a mechanism to identify the importance of words (e.g., coefficients), but these weights are global and overall importance scores that do not provide any explanatory power for each individual prediction. Some words that have a higher impact on one firm’s volatility may not have similar effects for other firms.

We pick a sample of 10-K filing from the firm Newmont Goldcorp (NYSE ticker: NEM), a mining company, and visualize the impact of words on its volatility prediction. Results are shown in Figure 5 where darker color indicates higher weight. We can see that words such as “shortages,” “compete,” “disruption,” “inadequate” and “costs” are assigned high weights in the HAN model. This result intuitively demonstrates that these words contribute more to the firm’s stock volatility prediction. Note that this mining company is facing several risk issues such as “energy shortages,” “a disruption in the transmission of energy” and “alternative sources of power may result in higher than anticipated costs.” This is consistent with prior empirical accounting studies concluding that negative words in a firm’s annual 10-K filings are highly correlated with the firm’s stock volatility (Kothari et al. 2009).

6. Conclusion, Discussion, and Future Direction

In this work, we propose a novel machine learning approach to analyze financial text reports for stock volatility prediction. Our method leverages recent advances in representation learning in NLP as well as traditional hand-crafted domain-specific lexicons. To incorporate such domain-specific lexicons into representation learning, we design a new objective



... Our operations may be adversely affected by energy shortages . Our mining operations and development projects require significant amounts of energy . Our principal energy sources are electricity , purchased petroleum products , natural gas and coal . and in some locations we compete with other companies for access to third party power generators or electrical supply networks . A disruption in the transmission of energy , inadequate energy transmission infrastructure or the termination of any of our energy supply contracts could interrupt our energy supply and adversely affect our operations ... The need to use alternative sources of power may result in higher than anticipated costs , which will affect operating costs ...

Figure 5 Visualization of impact of words on volatility prediction using the HAN model. Darker color indicates higher attention weight.

function that jointly learns word association under lexicon constraints. More specifically, we impose must-link and cannot-link constraints to explicitly model word associations. We demonstrate that the performance of stock volatility prediction can be improved by this knowledge-driven machine learning approach. Empirical experiments conducted on a dataset compiled of over ten years' of annual reports from Russell 3000 firms show that our method using financial domain-specific word embeddings outperforms various state-of-the-art baselines.

Our research has practical implications for stakeholders and the general public. For investors, who often use traditional stochastic models, our method can serve as another tool for forecasting stock volatility. Volatility forecasting is of the utmost importance for institutions involved in option trading and portfolio management. It is crucial for them not only to know the current volatility level of their managed assets, but also to be able to estimate future volatility. For regulatory agencies, stock market volatility forecasting is a task relevant to assessing market risk. Regulatory agencies can utilize this tool to evaluate the discrepancy between a firm's predicted volatility and realized volatility. If the discrepancy is outside a reasonable range, it may indicate that the firm does not fully disclose its risk to its investors. For academic researchers, our method provides a general outline that uses domain-specific word embeddings as input for predicting a critical financial variable. Previous text analysis on firm disclosure heavily relies on bag-of-words models that miss a great deal of important semantics. Our research answers a call to apply machine learning models to capture deeper semantics and latent context in business text (Loughran and McDonald 2016).

The idea of incorporating domain knowledge into machine learning also has implications for designing management information systems. Incorporating ontology (Sharman et al. 2004), common-sense knowledge (Li et al. 2016), and general prior knowledge (Pardoe et al. 2010) to design an intelligent information system has been extensively discussed in prior information systems' literature. For example, Gunning (2017) argue that providing human guidance can improve information system predictive performance and facilitate creation of mechanisms that can inspect AI system fairness, accountability and transparency. Despite the fact that our work lies in the financial domain, we expect our domain-specific word embedding method can be generalized to other areas (such as healthcare and legislation) where domain-specific lexicons are available. For example, in biomedical text analysis, researchers build bio-domain lexicons and ontologies such as the Unified Medical Language System (UMLS) to provide the necessary framework for semantic representation (Spasic et al. 2005). Another example is WordNet Miller (1995), a large lexical database of English, which has been widely used in text analysis and machine learning applications. We can also incorporate the WordNet lexicon to enhance the quality of word embedding for a general purpose. In WordNet, synonyms are grouped into synsets, and both nouns and verbs are organized into tree-structure hierarchies. Such WordNet knowledge can be converted into constraints and accordingly incorporated into our objective function (e.g., Eq. 6). In particular, words in the same synset can form must-links, while words in different sub-trees form cannot-links.

Our research still has several limitations that can be improved upon the future. First, we only focus on corporate annual report 10-K filings in this study. Prior literature has documented that other text sources such as corporate earnings announcements, analyst reports and firm-specific news also have explanatory power in relation to future stock return volatility. According to the famous Efficient Market Hypothesis (EMH), stock prices reflect all available information. Therefore, from a practical perspective, it is better to consolidate all available information in order to achieve accurate stock volatility prediction. It is also in the best interests of capital market participants to understand that using different sources of public information will increase the power of any predictive approach to stock volatility. To aid in such approaches, our domain-lexicon word embedding framework can be quite straightforwardly generalized to other text sources. The only caveat worth noting is that the L&M Dictionary is specifically designed for annual reports and only includes words that are commonly used by managers in those document. Analyzing other text resources may require different word lexicons. For example, Larcker and Zakolyukina (2012) create their own word lists to analyze conference calls.

Second, while we study stock volatility prediction using corporate annual reports, stock market investors are usually interested in volatility at a more fine-grained level, such as daily and monthly (Christoffersen and Diebold 2000). Prior research shows that while annual reports like those we study are more correlated with long term volatility (You and Zhang 2009), conference calls and firm-specific news may have short-term effects on stock volatility (Boudoukh et al. 2018). We therefore leave the use of shorter-term volatility prediction using other text resources such as social media posts or news articles as our future work. Such sources could allow us to empirically test whether the lexicon-incorporated word representation methods perform better when the market volatility is higher.

Third, our approach requires the important assumption that word categories should be exclusive, meaning that a word can only be assigned to one category. We believe this is a reasonable assumption from the perspective of generalizability, as many of the developed lexicons, such as the sentiment word lexicon (Hu and Liu 2004) and cognitive emotion lexicon (Pennebaker et al. 2001), do indeed consist of word categories that are exclusive. However, it is possible that some developed word lexicons contain non-exclusive words. Even if this is the case, our approach can still be leveraged in a way that cannot-link constraints on those exclusive words can be incorporated. We also leave this question for our future work.

Lastly, machine learning researchers are always interested in model prediction accuracy. However, uncovering the black-box model and making it more interpretable is also important, especially in the domain of financial technology (FinTech). Investors desire to understand the fundamental mechanisms of machine learning models and expect models to explain why a stock return volatility prediction is made. Even though we use the Hierarchical Attention Network (HAN) to reveal the impact of words in Form 10-Ks on volatility prediction, it is still arguable that the attention mechanism is suitable for explanation (Jain and Wallace 2019).

References

Adhikari A, Ram A, Tang R, Lin J (2019) Rethinking complex neural network architectures for document classification. *In Proceedings of NAACL*, 4046–4051.

Andrzejewski D, Zhu X, Craven M (2009) Incorporating domain knowledge into topic modeling via dirichlet forest priors. *In Proceedings of ICML*, 25–32.

Arora S, Li Y, Liang Y, Ma T, Risteski A (2016) A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics* 4:385–399.

Arora S, Liang Y, Ma T (2017) A simple but tough-to-beat baseline for sentence embeddings. *In Proceedings of ICLR*.

Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by jointly learning to align and translate. *In Proceedings of ICLR*.

Bao Y, Datta A (2014) Simultaneously discovering and quantifying risk types from textual risk disclosures. *Management Science* 60(6):1371–1391.

Bengio Y, Courville A, Vincent P (2013) Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(8):1798–1828.

Bernard D, Alexander K, Raman U (2007) Equilibrium portfolio strategies in the presence of sentiment risk and excess volatility. Working Paper 13401, National Bureau of Economic Research.

Black F, Scholes M (1973) The pricing of options and corporate liabilities. *Journal of political economy* 81(3):637–654.

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

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

Boudoukh J, Feldman R, Kogan S, Richardson M (2018) Information, trading, and volatility: Evidence from firm-specific news. *The Review of Financial Studies* 32(3):992–1033.

Büschken J, Allenby GM (2016) Sentence-based text analysis for customer reviews. *Marketing Science* 35(6):953–975.

Christoffersen PF, Diebold FX (2000) How relevant is volatility forecasting for financial risk management? *Review of Economics and Statistics* 82(1):12–22.

Das SR, Chen MY (2007) Yahoo! for amazon: Sentiment extraction from small talk on the web. *Management science* 53(9):1375–1388.

Devlin J, Chang MW, Lee K, Toutanova K (2019) Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL*, 4171–4186.

Drucker H, Burges CJC, Kaufman L, Smola AJ, Vapnik V (1997) Support vector regression machines. Mozer MC, Jordan MI, Petsche T, eds., *In Proceedings of NIPS*, 155–161.

Dyer T, Lang M, Stice-Lawrence L (2017) The evolution of 10-k textual disclosure: Evidence from latent dirichlet allocation. *Journal of Accounting and Economics* 64(2-3):221–245.

Faruqui M, Dodge J, Jauhar SK, Dyer C, Hovy E, Smith NA (2015) Retrofitting word vectors to semantic lexicons. *In Proceedings of NAACL*, 1606–1615.

Frankel R, Johnson M, Skinner DJ (1999) An empirical examination of conference calls as a voluntary disclosure medium. *Journal of Accounting Research* 37(1):133–150.

Gunning D (2017) Explainable artificial intelligence (xai). *Defense Advanced Research Projects Agency, DARPA/I20 (DARPA, 2017)* .

Hu M, Liu B (2004) Mining and summarizing customer reviews. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 168–177.

Huang AH, Lehavy R, Zang AY, Zheng R (2017) Analyst information discovery and interpretation roles: A topic modeling approach. *Management Science* 64(6):2833–2855.

Jain S, Wallace BC (2019) Attention is not Explanation. *In Proceedings of NAACL*, 3543–3556.

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

Kearney C, Liu S (2014) Textual sentiment in finance: A survey of methods and models. *International Review of Financial Analysis* 33:171–185.

Kim Y (2014) Convolutional neural networks for sentence classification. *In Proceedings of EMNLP*, 1746–1751.

Kogan S, Levin D, Routledge BR, Sagi JS, Smith NA (2009) Predicting risk from financial reports with regression. *In Proceedings of NAACL*, 272–280.

Kothari SP, Li X, Short JE (2009) The effect of disclosures by management, analysts, and business press on cost of capital, return volatility, and analyst forecasts: A study using content analysis. *The Accounting Review* 84(5):1639–1670.

Larcker DF, Zakolyukina AA (2012) Detecting deceptive discussions in conference calls. *Journal of Accounting Research* 50(2):495–540.

Li F (2010) The information content of forward-looking statements in corporate filingsa naïve bayesian machine learning approach. *Journal of Accounting Research* 48(5):1049–1102.

Li X, Chen K, Sun SX, Fung T, Wang H, Zeng DD (2016) A commonsense knowledge-enabled textual analysis approach for financial market surveillance. *INFORMS Journal on Computing* 28(2):278–294.

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

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

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

Mikolov T, Sutskever I, Chen K, Corrado G, Dean J (2013) Distributed representations of words and phrases and their compositionality. *In Proceedings of NIPS*, 3111–3119.

Miller GA (1995) Wordnet: A lexical database for english. *Commun. ACM* 38(11):3941, ISSN 0001-0782, URL `http://dx.doi.org/10.1145/219717.219748`.

Pardoe D, Stone P, Saar-Tsechansky M, Keskin T, Tomak K (2010) Adaptive auction mechanism design and the incorporation of prior knowledge. *INFORMS journal on Computing* 22(3):353–370.

Pennebaker JW, Francis ME, Booth RJ (2001) Linguistic inquiry and word count (liwc): Liwc2001.

Pennington J, Socher R, Manning C (2014) Glove: Global vectors for word representation. *In Proceedings of EMNLP*, 1532–1543.

Poon SH, Granger C (2005) Practical issues in forecasting volatility. *Financial analysts journal* 61(1):45–56.

Poon SH, Granger CW (2003) Forecasting volatility in financial markets: A review. *Journal of economic literature* 41(2):478–539.

Qin Y, Yang Y (2019) What you say and how you say it matters: Predicting stock volatility using verbal and vocal cues. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 390–401.

Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I (2019) Language models are unsupervised multitask learners .

Rekabsaz N, Lupu M, Baklanov A, Hanbury A, Duer A, Anderson L (2017) Volatility prediction using financial disclosures sentiments with word embedding-based ir models. *In Proceedings of ACL* 1712–1721.

Schölkopf B, Simard P, Smola AJ, Vapnik V (1998) Prior knowledge in support vector kernels. *In Proceedings of NIPS*, 640–646.

Sharman R, Kishore R, Ramesh R (2004) Computational ontologies and information systems ii: Formal specification. *Communications of the Association for Information Systems* 14(1):9.

Spasic I, Ananiadou S, McNaught J, Kumar A (2005) Text mining and ontologies in biomedicine: making sense of raw text. *Briefings in bioinformatics* 6(3):239–251.

Sridharan SA (2015) Volatility forecasting using financial statement information. *The Accounting Review* 90(5):2079–2106.

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

Tsai MF, Wang CJ (2014) Financial keyword expansion via continuous word vector representations. *In Proceedings of EMNLP*, 1453–1458.

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Lu, Polosukhin I (2017) Attention is all you need. *Proceedings of NIPS*, 5998–6008.

Wagstaff K, Cardie C, Rogers S, Schrödl S, et al. (2001) Constrained k-means clustering with background knowledge. *In Proceedings of ICML*, volume 1, 577–584.

Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E (2016) Hierarchical attention networks for document classification. *In Proceedings of NAACL*, 1480–1489.

You H, Zhang Xj (2009) Financial reporting complexity and investor underreaction to 10-k information. *Review of Accounting studies* 14(4):559–586.

Zhang D, Zhou ZH, Chen S (2007) Semi-supervised dimensionality reduction. *In Proceedings of SDM*, 629–634 (SIAM).