Predicting Stock Volatility Using Domain Lexicon Enhanced Representation Learning

Abstract

Predicting stock price volatility using firm disclosed reports is an essential task in financial markets. Finance economists have developed finance-specific lexicons in order to analyze the sentiments and semantics of firm reports. Recent advance in representation learning has demonstrated its superior performance on many text analysis tasks. Thus, we frame our research question as: can we achieve better stock volatility prediction by incorporating financial lexicon with representation learning? In this work, we propose two methods for lexicon incorporation. The first method models lexicon word associations as must-link and cannot-link constraints and incorporate the constraints into `word2vec` objective function. The second method averages word embeddings from different lexicon categories separately to obtain document embeddings. Empirical results show that, by taking advantage of domain lexicon and representation learning, both methods significantly reduce the volatility prediction error, compared with baselines that without considering domain lexicon. We open source our models and financial-specific embeddings so that capital market participants can use them to better assess financial risks and design trading strategies.

1 Introduction

Predicting financial risks of publicly traded companies is of great interests to capital market participants, and it is also the essence of investment portfolio management. In finance, stock price volatility, which is the standard deviation of a stock's returns over a period of time, is often used as a measure of financial risks. Unlike directly predicting stock returns or prices, it is uncontroversial in the field of economics that one can predict a stock's volatility level using public information [Bernard *et al.*, 2007].

Traditionally, stock volatility is quantitatively modeled by historical trading data. Recently, volatility prediction using firm disclosures has drawn attention from academia and industry. In the United States, the Securities Exchange Commission (SEC) mandates all public companies to file annual reports, which contain comprehensive information about companies' business. As a result, it has been an active research topic that whether one can predict a company's risk from firm disclosed reports. Pioneering work such as [Kogan *et al.*, 2009] formulates the above question as a text regression problem: give a piece of firm disclosure, predict a real-valued quantity, i.e., stock volatility, associated with that text. Moreover, economists and finance researchers have developed finance-specific lexicon to facilitate financial text analysis. The lexicon groups financial terms into different categories, such as positive, negative, uncertainty, litigation, etc. Prior research along this line mainly uses bag-of-words, either a generic word list [Kogan *et al.*, 2009] or lexicon word lists, to represent firm disclosures [Tsai and Wang, 2014; Rekabsaz *et al.*, 2017] for volatility prediction.

However, bag-of-words model is limited by its representation power, and hand-crafted lexicons only contain a small set of financial terms. Inspired by recent advance in representation learning [Mikolov *et al.*, 2013; Le and Mikolov, 2014], this paper studies the following question: can we incorporate domain lexicon with representation learning and achieve improvement on stock volatility prediction? Toward this end, we propose two methods for incorporating domain-specific lexicon with representation learning. Prior research [Xu *et al.*, 2014; Tkachenko *et al.*, 2018] on knowledge incorporation word embedding only considers adding constraints for words belong to the same lexicon category, i.e., must-link constraints. Instead, we propose to model word associations in lexicon as must-link and cannot-link constraints, and we combine constraint objective with `word2vec` objective function. Moreover, there is little work that incorporates lexicon for document representations. We propose a simple yet effective solution to represent document vectors by averaging words from different lexicon categories. We thoroughly evaluate these two methods with various baselines. The empirical results show that by incorporating lexicon knowledge into representation learning, we can significantly reduce stock volatility prediction error.

The contribution of this work can be summarized below as two folds. Firstly, we are among the first to incorporate finance-specific lexicon with representation learning for stock volatility prediction. Secondly, we empirically show that the domain lexicon enhanced representation learning can indeed reduce prediction error, compared to bag-of-words models and vanilla word and document embedding. In the next section, we briefly provide institutional background on financial disclosure and finance-specific lexicon. In Section 3, we outline related work in financial text regression and knowledge incorporation representation learning. We then introduce details of our knowledge incorporation models in Section 4, and dataset and experiment results are shown and discussed in Section 5 and Section 6 respectively. We conclude this paper in Section 7.

2 Financial Disclosure and Finance-specific Lexicon

In the United States, the Securities Exchange Commission (SEC) mandates all publicly traded companies to file annual reports, known as **Form 10-K**. This document provides a comprehensive overview of the company's business and financial condition. The Form 10-Ks are audited by an approved accountancy firm, and they are publicly available and can be accesses from SEC website[1].

Specifically, Section *Item 1A - Risk Factors* of Form 10-K describes the most significant risks that could harm the business. For example, an offshore oil driller company may list the risk of losses from a major accident and oil spill. Prior research [Dyer *et al.*, 2017] suggests that 10-K reports are long and redundant, and only the *Risk Factors* section appears to be informative to the investor. Therefore, in this paper, we focus on volatility prediction using Section *Item 1A - Risk Factors* of Form 10-K.

Economists and finance researchers have hand-crafted finance-specific lexicons to analyze the semantic and sentiment content in financial reports. The most widely used lexicon is Loughran & McDonald Word List **(L&M Dictionary)** [Loughran and McDonald, 2011]. This dictionary only contains words used by managers in Form 10-Ks, and it includes different categories (Positive, Negative, Uncertainty, Litigious, Constraining, Superfluous, etc). A large body of finance and accounting research (see survey paper [Loughran and McDonald, 2016]) have used the L&M dictionary to gauge the tone and semantics of Form 10-Ks. An important feature of L&M dictionary is that the word categorization are based on a financial context, for example the word "restatement" is generally more negative [2] when used in finance than when used in general English.

3 Related Work

Our work is closely related with the following two lines of research:

financial risk prediction with text: It is a received wisdom in economics and finance that one can predict a stock's risk using historical information [Bernard *et al.*, 2007]. Stock risks, as measured by variance of price over a period of time, displays an auto-regressive conditional heteroscedasticity effect [Engle, 1982], which means that the variance tends to change gradually. Various work has studies the problem of financial risk prediction using firm financial reports. A pioneer work [Kogan *et al.*, 2009] shows that simple bag-of-words features in Form 10-Ks combined with historical volatility can simply outperform statistical models that is built upon historical volatility only. Other work [Rekabsaz *et al.*, 2017] proposes to use weighted bag-of-words features to represent Form 10-Ks. Similarly, [Tsai and Wang, 2014] and [Theil *et al.*, 2018] use word embedding method to expand L&M Dictionary for volatility prediction. In addition to Form 10-K, recent research has investigated other sources of firm related documents, such as firm quarterly earning conference call transcripts [Wang and Hua, 2014], CEO letters and bank reports [Nopp and Hanbury, 2015]. Broadly speaking, our paper is also aligned to recent studies that make use of news articles and social media data to predict the stock market return [Tetlock, 2007; Schumaker and Chen, 2009; Ding *et al.*, 2015].

incorporating knowledge with representation learning: Despite our financial domain, our approach is relevant to representation learning with domain knowledge. Previous models [Tkachenko *et al.*, 2018; Xu *et al.*, 2014] show that the quality of word embedding *word2vec* can be improved by incorporating semantic knowledge from lexicons. Both work consider only constraints for words in the same lexicon category, i.e., must-links. Other work [Yu and Dredze, 2014; Faruqui *et al.*, 2015] also propose to refine the pre-trained word embeddings in a post-hoc manner. Our work differs from the above methods as we represent domain-specific lexicon as must-links and cannot-links constraints, and we design a new constraint objective functions based on the idea of using the algebraic property of the word embeddings. Moreover, we also incorporate domain lexicon with document representations. To the best of our knowledge, there is little work that incorporates domain lexicon with document representation learning. In prior work such as [Faruqui *et al.*, 2015], they simply represent document as the average of the word vectors for downstream classification tasks (such as sentiment analysis). In other words, domain knowledge is ignored in document representation in prior work.

4 Methodology

Our research question can be formulated as: given a lexicon and a collection of documents, how to learn good word or document representations to predict a real-valued quantity associated with the documents.

4.1 Incorporating Lexicon with Word Embeddings: LEX-w2v

The first method *LEX-w2v* aims to learn more domain-specific word embeddings by incorporating domain-specific lexicon. Inspired by the constrained clustering using the must-link and cannot-link relations, we introduce a new constraint objective to the original `Word2Vec` objective function.

Word2Vec We choose to incorporate lexicon with `Word2Vec` framework [Mikolov *et al.*, 2013] due to its wide adoption. Two training schemes, Skip-gram and CBOW, are

[1] http://www.sec.gov/edgar.shtml

[2] 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 of the stock price.

introduced in `Word2Vec` [Mikolov *et al.*, 2013]. Since both schemes have very similar objective function, for the sake of simplicity, we only consider incorporating lexicon with Skip-gram model. Here, we briefly review Skip-gram model.

The objective of Skip-gram is to maximize the probability of the context (output) words conditioning on a central (input) word:

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

where w_I is the target word and $w_{O_1}, ..., w_{O_C}$ are C context words. Thus, the objective function can be written as:

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

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

To approximate the conditional probability, two set of weights are kept as the input vectors $\boldsymbol{v}^0$ (the word embeddings) and output vectors $\boldsymbol{v}$. Then the probability of word w_O conditioning on word w_I is approximated by a softmax function:

$$P(w_O|w_I) = \frac{\exp(\boldsymbol{v}_O^T \boldsymbol{v}_I^0)}{\sum_{\boldsymbol{v}' \in V} \exp(\boldsymbol{v}'^T \boldsymbol{v}_I^0)} \tag{3}$$

However, summing over all the words in the vocabulary to calculate the softmax is very time consuming. Therefore, a negative sampling strategy is used which only samples a small set of words according to the word frequencies in the training set.

In details, for each update, we sample k words $w_1, ..., w_k$ from an empirical unigram distribution $P_W(c) = \frac{(\#c)^{\frac{3}{4}}}{Z}$, where Z is the normalization denominator. The probability of word w_O conditioning on word w_I is now approximated by the product of some sigmoid functions:

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

The target function can now be written as:

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

Constraint Objective Several prior research [Xu *et al.*, 2014; Tkachenko *et al.*, 2018] studies how to incorporate semantic knowledge with word embeddings. The general idea is to introduce knowledge as a constraint objective $\mathcal{L}_C$ in additional to the original `Word2Vec` objective function $\mathcal{L}_{W2V}$.

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

where λ is the hyperparameter controlling the strength of the constraint. However, they only consider adding constraints for words with in the same lexicon category. For example, [Tkachenko *et al.*, 2018] makes a strong constraint assumption that all positive words in the lexicon are centered around a center vector following a probability distribution.

Inspired by constraint clustering [Zhang *et al.*, 2007], we propose to represent lexicon as a set of must-link and cannot-link constraints. Words in the same lexicon category can be regarded as having a must-link constraint, and words in different lexicon categories can be regarded as having a cannot-link constraint. By incorporating both must-link and cannot-link constraints, we hope to learn better domain-specific word embeddings, so as to improve downstream prediction task.

Assume that a lexicon consists of K categories $C_1, C_2, \ldots, C_K$, and each category contains a set of semantically similar words, we propose constraint objective $\mathcal{L}_C$ as:

$$\begin{aligned} \mathcal{L}_C = &\sum_{\substack{k \neq h \\ k,h=1}}^{K} \beta_C \sum_{\substack{w_i \in C_k \\ w_j \in C_h}} \|\boldsymbol{v}_i^0 - \boldsymbol{v}_j^0\|^2 \\ &- \sum_{k=1}^{K} \beta_M \sum_{\substack{w_i \in C_k \\ w_j \in C_k}} \|\boldsymbol{v}_i^0 - \boldsymbol{v}_j^0\|^2 \end{aligned} \tag{7}$$

where $\|\cdot\|$ is the Euclidean distance, β_C and β_M are the hyperparameters controlling the strength of cannot-link relations and must-link relations respectively, $\boldsymbol{v}_i^0$ is the corresponding embedding for word w_i. By maximizing this constraint objective, we hope to decrease the distance between the words in the same category, but also increase the distance between the words in different categories.

Optimization The objective function $\mathcal{L}_{joint}$ (Eq. 4) allows stochastic gradient ascent optimization. The update function for input word vector and the corresponding gradient can be solved as:

$$\boldsymbol{v}_I^0 \leftarrow \boldsymbol{v}_I^0 + \alpha \frac{\partial \mathcal{L}_{joint}}{\partial \boldsymbol{v}_I^0} = \boldsymbol{v}_I^0 + \alpha \frac{\partial \mathcal{L}_{W2V}}{\partial \boldsymbol{v}_I^0} + \alpha\lambda \frac{\partial \mathcal{L}_C}{\partial \boldsymbol{v}_I^0} \tag{8}$$

where

$$\frac{\partial \mathcal{L}_{W2V}}{\partial \boldsymbol{v}_I^0} = \sum_{i=1}^{C} (\sigma(-\boldsymbol{v}_{O_i}^T \boldsymbol{v}_I^0) \boldsymbol{v}_{O_i} - \sum_{j=1}^{k} \sigma(\boldsymbol{v}_j^T \boldsymbol{v}_I^0) \boldsymbol{v}_j) \tag{9}$$

$$\begin{aligned} \frac{\partial \mathcal{L}_C}{\partial v_{ik}^0} = &\sum_{\substack{k \neq h \\ k,h=1}}^{K} 2\beta_C \sum_{w_j \in C_h} (\boldsymbol{v}_I^0 - \boldsymbol{v}_j^0) \\ &- \sum_{k=1}^{K} 2\beta_M \sum_{w_j \in C_k} (\boldsymbol{v}_I^0 - \boldsymbol{v}_j^0) \end{aligned} \tag{10}$$

The update function for output vector is the same as in Skip-gram model, i.e.

$$\boldsymbol{v}_j \leftarrow \boldsymbol{v}_j + \alpha \frac{\partial \mathcal{L}_{joint}}{\partial \boldsymbol{v}_j} = \boldsymbol{v}_j + \alpha \frac{\partial \mathcal{L}_{W2V}}{\partial \boldsymbol{v}_j} \tag{11}$$

Where

$$\frac{\partial \mathcal{L}_{W2V}}{\partial \boldsymbol{v}_j} = \begin{cases} \sigma(-\boldsymbol{v}_j^T \boldsymbol{v}_I^0)\boldsymbol{v}_I^0, \ if\ w_j\ is\ a\ context\ word \\ -\sigma(\boldsymbol{v}_j^T \boldsymbol{v}_I^0)\boldsymbol{v}_I^0, \ otherwise \end{cases} \tag{12}$$

In our implementation, we use the asynchronized stochastic gradient ascent and only sample a small fixed number of constraints from the lexicon for each update. We follow the same sampling strategy as the negative sampling for constraint objective, since the more frequent words have been trained more in Skip-gram and are more representative for the lexicon category to which they belong. Optimizing $\mathcal{L}_{joint}$ (Eq. 4) makes the words of the same lexicon category gravitate to the same semispace and words of different lexicon category to the opposite semispaces, while still traded off by $\mathcal{L}_{W2V}$ objective.

Algebraic Property Our design of constraint objective can be explained from the algebraic property of word embeddings. Nonlinear word embeddings models like Word2Vec and GloVe are well known for their good algebraic properties. For example, “king” - “man” + “woman” = “queen”. Most of them 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 [Levy and Goldberg, 2014; Arora *et al.*, 2016].

$$\boldsymbol{v}_w \cdot \boldsymbol{v}_{w'} \approx PMI(w, w') = \log \frac{P(w|w')}{P(w)} \tag{13}$$

Therefore, for a context word c, we have

$$\boldsymbol{v}_c \cdot (\boldsymbol{v}_{w_1} - \boldsymbol{v}_{w_2}) \approx \log \frac{P(c|w_1)}{P(c|w_2)} \tag{14}$$

If two words w_1 and w_2 have similar semantic meaning, 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 closed to 1, which means that

$$\boldsymbol{v}_c \cdot (\boldsymbol{v}_{w_1} - \boldsymbol{v}_{w_2}) = \|\boldsymbol{v}_c\| \|\boldsymbol{v}_{w_1} - \boldsymbol{v}_{w_2}\| \cos\theta_c \approx 0 \tag{15}$$

where θ_c is the angle between $\boldsymbol{v}_c$ and $\boldsymbol{v}_{w_1} - \boldsymbol{v}_{w_2}$. Since this equation holds for every context word c, we must have $\|\boldsymbol{v}_{w_1} - \boldsymbol{v}_{w_2}\|$ being considerably small.

If w_1 and w_2 are very different semantically, then $\frac{P(c|w_1)}{P(c|w_2)}$ should not be closed to 1 which means $\|\boldsymbol{v}_{w_1} - \boldsymbol{v}_{w_2}\|$ should not be small. Therefore, by adding our constraint objective Equation 7, we simply enforce this algebraic property by making the embeddings of semantically similar words closer in the embedding space while making the embeddings of semantically distinct words farther.

4.2 Incorporating Lexicon with Document Embeddings: LEX-d2v

Averaging word embeddings to obtain document embedding is simple and effective in the case where domain lexicon is not available. However, the disadvantage of averaging the word embeddings over the whole vocabulary to represent a document is that, by adding word embeddings together, we are eliminating the difference between them. For example, “king” - “man” + “woman” = “queen” can also be written as “king” + “woman” = “queen” + “man”. By adding “king” and “woman” together, we loss the information of gender difference. This side effect would be especially profound for documents that consist of words from semantically different lexicon categories. For example, it is very common that a financial report contains both positive words for profit and negative words for risks and losses. When domain lexicon is available, we can incorporate the lexicon as knowledge to improve the quality of document representations. Here we propose a simple strategy *LEX-d2v* that averages words from different lexicon categories separately, instead of pooling all the words together.

Suppose the documents with respect to the words in the k-th category form a $n \times m_k$ matrix $\boldsymbol{W}_k$, where n is the number of documents in the data set and m_k is the number of words in the k-th category. Also, all the word embeddings in the k-th category form a $m_k \times d$ matrix $\boldsymbol{V}_k$, where d is the length of the embedding. Then we have $\boldsymbol{D}_k = \boldsymbol{W}_k \boldsymbol{V}_k$, where $\boldsymbol{V}_k$ is a $k \times d$ matrix formed by the embedding of all the words in the k-th category. Then we can get a new representation of the documents by concatenating them together:

$$\boldsymbol{D}_{concat} = (\boldsymbol{D}_1, \boldsymbol{D}_2, \ldots, \boldsymbol{D}_K, \boldsymbol{D}^*) \tag{16}$$

where $\boldsymbol{D}^*$ corresponding to all the words not included in the lexicon.

A elaborated example is that, if we average two word vectors that are from different lexicon categories (say one positive and one negative), it is likely that they would cancel out each other on some dimensions. But if we average the words from the two word lists separately, both semantics would be preserved, so are the corresponding document representations.

4.3 Evaluation Task: Stock Volatility Prediction

The stock volatility prediction problem is formulated following [Kogan *et al.*, 2009]. The volatility is defined as:

$$v_{[t-\tau,t]} = \ln\Big(\sqrt{\frac{\sum_{i=0}^{\tau}(r_{t-i} - \bar{r})^2}{\tau}}\Big) \tag{17}$$

where r_t is the return price at day t and $\bar{r}$ is the mean of the return price over the period of day $t - \tau$ to day t. The return price is defined as $r_t = \frac{P_t}{P_{t-1}} - 1$, where P_t is the close price on day t. Similar to [Rekabsaz *et al.*, 2017], we use $\tau = 64$ which is the number of trading days in a quarter. The yearly volatility is calculated by averaging over the four quarters in the year.

Suppose $\boldsymbol{x}_i$ is the document vector for a company in the k-th year, then our prediction target y_i is the volatility of the company in the $(k+1)$-th year. Given the document vector $\boldsymbol{x}_i$, we apply Support Vector Regression (SVR) [Drucker *et al.*, 1997] to predict y_i. SVR formulates the training as the following optimization problem:

$$\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{N}\sum_{i=1}^{N} \max(0, \|y_i - f(\boldsymbol{x}_i; \boldsymbol{w})\| - \epsilon) \tag{18}$$

where N is training set size. Similar to previous studies[Kogan *et al.*, 2009; Rekabsaz *et al.*, 2017; Tsai and Wang, 2014],

	train year	# train data	test year	# test data
1	2004-2010	1,994	2011	362
2	2005-2011	2,350	2012	365
3	2006-2012	2,470	2013	373
4	2007-2013	2,507	2014	370
5	2008-2014	2,533	2015	375
6	2009-2015	2,589	2016	378

Table 1: Form 10-Ks dataset for stock volatility prediction task.

we set C and ϵ to 1.0 and 0.1 respectively. We use Radial Basis Function (RBF) kernel as it is reported to perform the best [Rekabsaz *et al.*, 2017].

We report the performance using the Mean Squared Error (MSE) between the predicted volatility and true volatility:

$$MSE = \frac{1}{M} \sum_{i=1}^{M} (f(\boldsymbol{x}'_i; \boldsymbol{w}) - y'_i)^2 \tag{19}$$

where M is the size of the test set, and y'_i is the true volatility associated with testing example $\boldsymbol{x}'_i$.

5 Data

S&P 500 companies and Form 10-Ks We choose S&P 500 constituent firms as the target for volatility prediction for reasons of importance and tractability. Firms in the S&P 500 index encompass roughly three-quarters of the total U.S. market capitalization. We collect 17,107 annual reports published over the period of 1998-2018 for S&P 500 firms. Section *Item 1A - Risk Factors* of Form 10-K is extracted as it contains more information on business risks, and it is used to predict stock volatility. We also obtain daily stock prices of 2004-2017 (dividend-adjusted) from CRSP database. The dataset detail is shown in Table 1.

For preprocessing, all the punctuation is removed, and all the words are converted to their lower cases. All the documents with less than 3 sentences and all the sentences with less than 5 words are ignored.

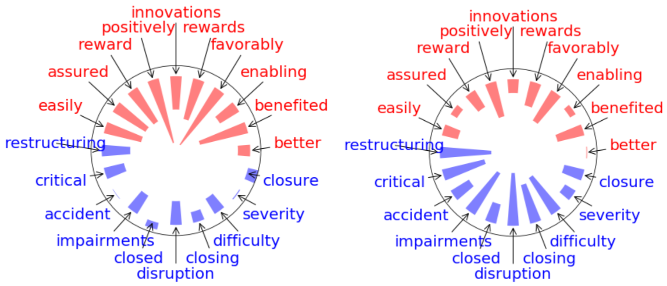
L&M Dictionary This finance-domain specific lexicon group words into categories including Negative, Positive, Uncertainty, Litigious, Constraining, Interesting, etc. For our volatility prediction, we only consider incorporating risk-related categories: Negative, Positive and Uncertainty. Each category contains 2,355, 354, 257 distinct words respectively.

We will also release our processed Form 10-Ks and stock volatility data for readers who are interested in back-testing trading strategies based on our volatility prediction model.

6 Experiment Results

Baselines We consider several stock volatility prediction baselines as described below. The first baseline is a stock's past volatility. 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 v^{past}.

Prior work use different variants of bag-of-words model to represent a financial document. These baselines are:



(a) benefit

(b) loss

Figure 1: Euclidean distance changes after incorporating semantic knowledge to (a) "benefit" and (b) "loss". Red bars and blue bars belong to positive words and negative words respectively. The positive direction is pointing outward.

- *tf-idf*: used in [Kogan *et al.*, 2009]. The feature value is classic tf-idf score.
- *tf-idf+*: used in [Tsai and Wang, 2014]. The feature space only contains a set of keywords derived from L&M dictionary.
- *BM25*: used in [Rekabsaz *et al.*, 2017]. They use a weighting strategy called *BM25*, instead of tf-idf score. The feature space also only contains a set of keywords derived from L&M dictionary.

To evaluate the performance of incorporating knowledge into word embedding, we consider the following baselines:

- *Word2Vec*: It is vanilla Skip-gram [Mikolov *et al.*, 2013] word embeddings trained on all Form 10-Ks.
- *SentiVec*: used in [Tkachenko *et al.*, 2018]. They derive a set of constraints only for words in the same lexicon category, i.e., must-link constraints.
- *Retrofit*: used in [Faruqui *et al.*, 2015]. They use lexicon to refine pre-trained word2vec in a post-hoc manner.

After we obtain word representations using the above embedding methods, we use a simple tf-idf weighted averaging method to obtain document representations. By using this simple averaging method, we can fairly compare the performance between embedding methods and bag-of-words models, as well as among different embedding methods. This is also a common practice for evaluating the quality of word embeddings in prior work [Faruqui *et al.*, 2015].

To evaluate the performance of incorporating knowledge into document embedding, in addition to bag-of-words baselines, we consider the following baselines:

- *WeightedAverage*: used in [Arora *et al.*, 2017]. The document embedding is the tf-idf weighted average over all vanilla Skip-gram word embeddings.
- *Doc2Vec*: We also use *Doc2Vec* [Le and Mikolov, 2014] as it is a commonly used document embedding baseline.

Results It is worth noting that predicting stock volatility is a rather challenging task given the noisiness of the stock market. Therefore, prior research [Kogan *et al.*, 2009] reports volatility number in the 4-th decimal. All experiments are

	2012	2013	2014	2015	2016	2017	Average
v^{past}	0.0745	0.0650	0.0817	0.0999	0.0772	0.0706	0.0782
tf-idf	0.0966	0.0878	0.1108	0.1381	0.0748	0.0686	0.0961
tf-idf+	0.0702	0.0668	0.0801	0.0952	0.0558	**0.0606**	0.0714
BM25	0.0741	0.0648	0.0819	0.1019	0.0560	0.0616	0.0734
Word2Vec	0.0533	0.0652	0.0709	0.0868	0.0550	0.0642	0.0659
SentiVec	**0.0532**	0.0646*	0.0711	0.0870	0.0545*	0.0635*	0.0656*
Retrofit	0.0544	0.0665	0.0718	**0.0859***	0.0551	0.0634*	0.0662
LEX-w2v	0.0534	**0.0645***	**0.0706**	0.0864*	**0.0541***	0.0642	**0.0655***

Table 2: MSE of using different word embedding methods to predict the stock volatility. An asterisk means statistically significant result compared to *Word2Vec* at 5% level under a one-tailed t-test ($p < 0.05$).

	2012	2013	2014	2015	2016	2017	Average
WeightedAverage	**0.0533**	0.0652	0.0709	**0.0868**	0.0550	0.0642	0.0659
Doc2Vec	0.0650	0.0652	0.0762	0.1005	0.0523*	0.0626*	0.0703
LEX-d2v	0.0551	**0.0591***	**0.0692***	0.0922	**0.0521***	**0.0620***	**0.0650***

Table 3: MSE of using different document embedding methods to predict the stock volatility. An asterisk means statistically significant result compared to WeightedAverage at 5% level under a one-tailed t-test ($p < 0.05$).

	10×10^{-4}	6×10^{-4}	4×10^{-4}
LEX-w2v	0.0657*	0.0656*	0.0655*

Table 4: Average MSE using different λ.

	16	32	64
LEX-w2v	0.0656*	0.0655*	0.0658

Table 5: Average MSE of using different number of constraints.

repeated for 10 times by setting different random seeds, and the mean number and significance level are reported.

All of the word embeddings and *Doc2Vec* have a length of $d = 200$. *Word2Vec*, *LEX-w2v*, *SentiVec* and *Doc2Vec* are trained using 10 threads with a same learning rate.

Word Embedding with Lexicon Incorporation Table 2 shows that using the word embeddings significantly outperforms the best weighting scheme baseline, *tf-idf+*, by over 7%. Our word embedding method *lex-w2v* outperforms all the other baselines in general. *SentiVec* also outperform vanilla embeddings on average but not as significantly as our method. *Retrofit* outperforms the vanilla embedding in some years, but it performs worse than the vanilla embeddings on average. For our proposed word embedding method *LEX-w2v*, the parameter we used are: $\beta_M = \beta_C = 1$ and $\lambda = 4 \times 10^{-4}$. The results show that word embeddings achieve better volatility prediction than prior bag-of words models, and incorporating domain-specific knowledge can further boost prediction performance. Moreover, our model *LEX-w2v* that incorporates both word must-link and cannot-link constraints achieves the best, and statistically significant, average performance.

To check the robustness of our method, as shown in Table 4 and Table 5, we change the value of λ (4×10^{-4}, 6×10^{-4}, and 10×10^{-4}) and the number of constraints sampled from the lexicon (16, 32, 64) for each update. The results show that our method can still significantly outperform the baselines.

$\beta_M = 1, \beta_C = 0$	$\beta_M = 0, \beta_C = 1$	$\beta_M = 1, \beta_C = 1$
0.0659	0.0660	0.0655*

Table 6: Average MSE of *LEX-w2v* only incorporating must-link relations or cannot-link relations.

We also investigate the effect of must-link constraints and cannot-link constraints on the performance. As shown in Table 6, we keep the other parameters unchanged only setting $\beta_M = 0$ or $\beta_C = 0$. In general, only adding must-link or cannot-link constraints cannot improve the quality of word embeddings significantly, compared with baselines.

Visualize Changes To qualitatively examine the effect of our method *LEX-w2v*, We visualize the Euclidean distance change in the embedding space after lexicon is incorporated. i.e. for two words a and b, we measure

$$\|\boldsymbol{v}^a_{\text{LEX-w2v}} - \boldsymbol{v}^b_{\text{LEX-w2v}}\| - \|\boldsymbol{v}^a_{\text{Word2Vec}} - \boldsymbol{v}^b_{\text{Word2Vec}}\| \tag{20}$$

We choose two reference words, “benefit” and “loss”, which are the most frequent word in Positive word list and Negative word list respectively. Then we choose 10 Positive words and 10 Negative words from the L&M Dictionary and compute their distance to these two reference words by Equation 20.

In Figure 1, as most of the bars are pointing inward, we can see that the distances between most of the chosen words and reference words decrease. This could because that the embedding length is decreased as the average length of *LEX-w2v* (3.1071) is significantly shorter than that of *Word2Vec* (3.1782). But for the positive words, their distances to “benefit” decrease more than their distances to “loss”. For the negative words, their distances to “loss” decrease more than their distances to “benefit”. So relatively speaking, words in the same category become closer while words in different categories become further in the embedding space.

Document Embedding with Lexicon Incorporation Table 3 shows that by incorporating semantic knowledge into document embeddings, we can boost the performance even more than the weighted average document embedding methods. On average, our proposed *LEX-d2v* outperforms *tf-idf*+ by over 9%, *Word2Vec* by about 1.5% and *Doc2Vec* by over 7%, all statistically significant. One thing worth noting is that we also try a hybrid method, that is we obtain word embeddings with *LEX-w2v* method and then represent document embeddings with *LEX-d2v* method. The experiment result shows that this hybrid method significantly outperforms *LEX-w2v* but does not significantly outperform *LEX-d2v*. One explanation is that *LEX-w2v* already incorporates domain lexicon knowledge and it will not help the document representation as much by incorporating lexicon knowledge again in *LEX-d2v*. The results are similar when combining other constrained word embedding methods (*SentiVec* and *Retrofit*) with *LEX-d2v*, which suggests that incorporating the same lexicon twice (word embedding and document embedding level) will weaken the effect of *LEX-d2v*.

7 Conclusion

In this work, we have demonstrated that the performance of stock volatility prediction can be improved by taking advantage of representation learning and domain lexicon. We propose two methods to incorporate lexicon with representation learning, one at word embedding level and one at document embedding level. Empirical results show that our word embedding and document embedding with finance-specific lexicon incorporation outperforms various baselines. Despite our financial domain, we hope our two different methods of incorporating domain knowledge with representation learning can also be useful in other areas (such as healthcare and legislation) where domain-specific lexicon is also available.

References

[Arora *et al.*, 2016] Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399, 2016.

[Arora *et al.*, 2017] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. In *In Proceedings of ICLR*, 2017.

[Bernard *et al.*, 2007] Dumas Bernard, Kurshev Alexander, and Uppal Raman. Equilibrium portfolio strategies in the presence of sentiment risk and excess volatility. Working Paper 13401, National Bureau of Economic Research, September 2007.

[Ding *et al.*, 2015] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. Deep learning for event-driven stock prediction. In *In Proceedings of IJCAI*, pages 2327–2333, 2015.

[Drucker *et al.*, 1997] Harris Drucker, Christopher J. C. Burges, Linda Kaufman, Alex J. Smola, and Vladimir Vapnik. Support vector regression machines. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *In Proceedings of NIPS*, pages 155–161. 1997.

[Dyer *et al.*, 2017] Travis Dyer, Mark Lang, and Lorien Stice-Lawrence. The evolution of 10-k textual disclosure: Evidence from latent dirichlet allocation. *Journal of Accounting and Economics*, 64(2-3):221–245, 2017.

[Engle, 1982] Robert Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, 50(4):987–1007, 1982.

[Faruqui *et al.*, 2015] Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. Retrofitting word vectors to semantic lexicons. In *In Proceedings of NAACL*, pages 1606–1615, 2015.

[Kogan *et al.*, 2009] Shimon Kogan, Dimitry Levin, Bryan R. Routledge, Jacob S. Sagi, and Noah A. Smith. Predicting risk from financial reports with regression. In *In Proceedings of NAACL*, pages 272–280, 2009.

[Le and Mikolov, 2014] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *In Proceedings of ICML*, pages II–1188–II–1196, 2014.

[Levy and Goldberg, 2014] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *In Proceedings of NIPS*, pages 2177–2185, 2014.

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

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

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *In Proceedings of NIPS*, pages 3111–3119, 2013.

[Nopp and Hanbury, 2015] Clemens Nopp and Allan Hanbury. Detecting risks in the banking system by sentiment analysis. In *In Proceedings of EMNLP*, pages 591–600, Lisbon, Portugal, 2015.

[Rekabsaz *et al.*, 2017] Navid Rekabsaz, Mihai Lupu, Artem Baklanov, Allan Hanbury, Alexander Duer, and Linda Anderson. Volatility prediction using financial disclosures sentiments with word embedding-based ir models. *In Proceedings of ACL*, pages 1712–1721, 2017.

[Schumaker and Chen, 2009] Robert P. Schumaker and Hsinchun Chen. Textual analysis of stock market prediction using breaking financial news: The azfin text system. *ACM Trans. Inf. Syst.*, 27, 2009.

[Tetlock, 2007] Paul C. Tetlock. Giving content to investor sentiment: The role of media in the stock market. *Journal of Finance*, 62(3):1139–1168, 2007.

[Theil *et al.*, 2018] Christoph Kilian Theil, Sanja Stajner, and Heiner Stuckenschmidt. Word embeddings-based uncertainty detection in financial disclosures. In *In Proceedings of the First Workshop on Economics and Natural Language Processing*, pages 32–37, 2018.

[Tkachenko *et al.*, 2018] Maksim Tkachenko, Chong Cher Chia, and Hady Lauw. Searching for the x-factor: Exploring corpus subjectivity for word embeddings. In *IN Proceedings of ACL*, pages 1212–1221, 2018.

[Tsai and Wang, 2014] Ming-Feng Tsai and Chuan-Ju Wang. Financial keyword expansion via continuous word vector representations. In *In Proceedings of EMNLP*, pages 1453–1458, 2014.

[Wang and Hua, 2014] William Yang Wang and Zhenhao Hua. A semiparametric gaussian copula regression model for predicting financial risks from earnings calls. In *In Proceedings of ACL*, volume 1, pages 1155–1165, 2014.

[Xu *et al.*, 2014] Chang Xu, Yalong Bai, Jiang Bian, Bin Gao, Gang Wang, Xiaoguang Liu, and Tie-Yan Liu. Rc-net: A general framework for incorporating knowledge into word representations. In *In Proceedings of CIKM*, pages 1219–1228, 2014.

[Yu and Dredze, 2014] Mo Yu and Mark Dredze. Improving lexical embeddings with semantic knowledge. In *In Proceedings of ACL*, pages 545–550, 2014.

[Zhang *et al.*, 2007] Daoqiang Zhang, Zhi-Hua Zhou, and Songcan Chen. Semi-supervised dimensionality reduction. In *In Proceedings of SDM*, pages 629–634. SIAM, 2007.