Investment Manager Discussions and Stock Returns: a Word Embedding Approach

Lee Gao
Tepper School of Business & Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213
lilig@andrew.cmu.edu

Abstract

Background. It has been debated for a long time whether institutional investment managers have superior ability to pick stocks and to time the market. If so, the next question is whether the investment managers deliver their market insights to investors. As more and more investors delegate their portfolios to investment managers in the U.S. financial market, the questions above are critical to understanding the value created by investment professionals.

Aim. This paper investigates whether institutional investment managers are capable in predicting market aggregate returns and whether their public discussions contain valuable market information.

Data. The stock return data are from the Center for Research in Security Prices database, and the textual data are letters to shareholders extracted from N-CSR(S) files from the Security and Exchange Commission Electronic Data Gathering, Analysis and Retrieval database. The N-CSR(S) files are annual (semi-annual) certified shareholder reports of registered management investment companies.

Methods. I quantify textual documents by mapping words and documents into a low dimensional vector space using the continuous bag-of-words (CBOW) neural network model. Then I use the document vectors to predict value-weighted market portfolio returns using elastic-net.

Results. The out-of-sample predictions show that the root mean square error can be reduced by about 6.6% when document vectors are included in the prediction model, in comparison to benchmark models including a constant, a momentum factor and a value factor. The in-sample regressions show that when the proportion of risk aversion related words increases by 1%, the expected annual stock return increases by 1-5%, which is both statistically and economically significant.

Conclusions. Investment managers have insights to predict market aggregate returns, and they convey valuable information to their investors in the letters to shareholders in their regulatory reports. The CBOW neural network word-embedding model provides an efficient way to retrieve information from textual documents. Textual features that predict stock returns contain information about the degree of risk aversion of investors.

Key Words: Word Embedding, Neural Network, Investment, Stock Returns

1 Introduction

Financial economists have been debating whether institutional investment managers have superior ability of picking stocks and timing the market for a long time and empirical evidence are mixed in

the previous literature. A following up question is, if investment professionals have valuable insights about market performance, would they deliver such kind of information to their investors. As more and more investors delegate their portfolios to investment managers in the U.S. financial market, the questions above are important to understand the value created by investment professionals.

My research contributes to the literature by getting new evidence supporting the claim that investment managers are adding value to investors. My evidence comes from a textual dataset, which contains letters to shareholders written by investment managers. I apply the continuous bag-of-words (CBOW) neural network model to quantify the textual documents in a systematically way. I also investigate the economic intuition of the stock return predicting power of the investment manager discussions and find the textual documents contains information about the degree of risk aversion of the investors, which agrees with asset pricing theory.

2 Problem and Approach

The question I am trying to answer is whether the information delivered to investors by investment managers provides useful insights in predicting aggregate stock excess returns.

To answer my question, I construct a textual dataset which contains the letters to shareholders extracted from the semi-annual shareholder reports (N-CSR and N-CSRS[1]) that registered management investment companies file with the Security and Exchange Commission (SEC). In these letters, investment managers discuss the macroeconomic environment, explain the constitutions of their asset holdings and the related performance, compare the fund performance with benchmarks and competing funds, as well as express opinions of future plans. Intuitively, the forward-looking statements and subjective opinions of the investment professionals contained in the letters may provide relevant information for the investors to understand the concurrent investment conditions, or reflect sentiments of the investment managers.

To make statistical inferences using textual documents, I quantify the letters by mapping words and documents into a low dimensional (relative to vocabulary size) vector space using the continuous bag-of-words (CBOW) neural network model proposed in Mikolov et al. (2013a)[2]. These vector representations for the words are called word embeddings. The word vectors are trained based on unsupervised learning algorithm that tries to predict a word based on its neighbors. In downstream prediction tasks, we need a vector representation for each document, and a document vector is calculated as the average of word vectors representing individual words appearing in the document. This approach of generating document vectors is referred as CBOW-Average. This is fundamentally different from the word counting approach based on pre-built dictionaries that are commonly applied in previous finance literature (Tetlock (2007), Loughran & McDonald (2011), Jegadeesh & Wu (2013), etc.). The advantage of my approach is that it avoids the subjectivity of human readers involved in building word classifying dictionaries, and it quantifies documents in a systematic way such that it requires much less human labor and can be applied to textual data of different domains.

The word embedding approach is drawing a great deal of attention from researchers in computational linguistics in recent years. In comparison to the traditional bag-of-words model, it generates superior results in many natural language processing (NLP) tasks such as part of speeching tagging, sentiment analysis, speech recognition, etc.

To test the prediction power of the document vectors, I conduct out-of-sample (OOS) predictions. The dependent variable is the annual stock return of the Center for Research in Security Prices (CRSP) value-weighted market portfolio, which is calculated as the accumulated return covering a 252-day period starting from the day following the N-CSR(S) release date. The explanatory variables include two controlling variables, the annual stock return of the market portfolio and in the one year period before the N-CSR(S) release date, and the dividend yield of the market portfolio.

[1] N-CSR and N-CSRS basically contained the same information. N-CSR is released at the end of a fiscal year, while N-CSRS is released at the half-way of a fiscal year. They are treated in the same way in constructing the letters to shareholders dataset.

[2] A related neural network model introduced in Mikolov et al. (2013b,a) is called Skip-Gram, while in CBOW, word vectors are trained based on unsupervised learning algorithm that tries to predict a word based on its neighbors; in Skip-gram, word vectors are trained to predict the surrounding words of a word based on a target word.

The whole sample set include the 2,255 daily observations covering the period 2003-2015. I construct a training set and a test set in two ways. First, I pool all the 2,255 observations together ignoring their time stamp and randomly select 70% of the samples to form the training set, and use the rest samples to build the test set. I estimated a linear model based on the training set using elastic-net. Elastic-net is capable of dealing with high-dimension explanatory variables as the penalization in the L1 and L2 -norm of the coefficients could reduce overfitting. I find that including the document vectors can reduce the OOS prediction root mean square errors (RMSEs) significantly, by about 6.6%.

As constructing the training and test sets through random splitting may introduce looking ahead bias as the training set contain future information in comparison to the test set. Therefore, in the second way, I split the training and the test sets on rolling window basis. For every 6-year window, I estimate the predicting model using the data in the leading five years and make OOS predictions in the sixth year. In this approach, I still find that including the document vectors in the prediction can still reduce the OOS prediction RMSEs significantly. This rolling window based OOS predictions confirm that the letters to shareholders contain substantial return predicting information.

Generally speaking, the CBOW neural network model can be considered as a kind of dimension reduction technique that summarizes sparse information contained in documents into a low-dimensional vector. However, it is not the only way to learn low dimensional vector representations of words and documents. I compare the predictive power of document vectors generated by CBOW-Average with six other language models: CBOW-Doc, CBOW-Kmeans, CBOW-Spectral_Clustering, Sentiment_Counting, Latent Semantic Analysis (LSA), and Latent Dirichlet Allocation (LDA). Through the comparison, I find that CBOW-Average generates smallest OOS prediction RMSEs when the training and test set are split in a rolling window basis, and CBOW-Doc generates smallest OOS prediction RMSEs when the training and test set are split randomly.

In additional to stock returns, I also investigate the predicting power of textual features in predicting stock return volatilities and the growth rates of oil price, dollar index, and default spreads. I find that including the textual features into the model can reduce the OOS prediction RMSEs significantly, in comparison to benchmark models without the textual features.

Also, I investigate the economic meaning of the textual information that can predict stock returns. As previous research in asset pricing suggest that the predictive part of stock returns is risk premium, which is affected by the degree of risk aversion of a representative investor. I construct two measures of risk aversion based on counting the frequency of words that are related to investment uncertainties and business cycles. Notice that my approach of classifying words is based on the semantic distance measured by the cosine similarity of their embedding vectors learned based on CBOW, rather than human designed rules, which is free of subjective judgment and is easy to be applied to a different corpus. I find that my text-based risk aversion measure contains information in predicting stock returns. When the proportion of investment uncertainty related words increase by 1%, the expected annual stock returns increase by 5%, which is economically and statistically significant; and when the proportion of business cycle related words increase by 1%, the expected annual stock returns increase by 1%.

3 Background and Related Work

This paper is related to two strands of literature in finance. First, it is related to the literature of stock return predictability. The predictability of stock returns has been under debate for a long time (Campbell & Yogo (2006); Ang & Bekaert (2007); Cochrane (2011); Fama & French (1988)). Now many financial economists agree that long-term stock returns are predictable. In particular, the predictable part of stock returns is risk premium. As the risk aversion property of an investor is subjective in nature, the degree of risk aversion is difficult to measure empirically. However, the textual data of letters to shareholders, which reflect the subjective opinions of investment managers, provide a unique source to measure risk aversion. Intuitively, the risk aversion nature of an investment manager affects the information he/she puts into the letters to shareholders. I construct proxies that measure the risk aversion of investors to predict stock returns by retrieving the textual information in the letters. In addition, I also find the investment manager discussions contain information in predicting future stock return volatilities, as well as some macroeconomic indicators. This results agrees with the previous literature about stock return predictability such as Kogan et al. (2009).

Second, this paper is related to the literature about investment manager abilities. It has been discussed for a long time whether investment managers have superior abilities to pick stocks or to time the market and add value to their clients (Edwards & Caglayan (2001); Brands et al. (2005); Cremers & Petajisto (2009)). Understanding how investment managers add value is important because a significant and growing proportion of individual investors delegate their portfolio management to investment professionals. Kacperczyk et al. (2014) found that a small subset of funds persistently outperforms, due to their superior capabilities of picking stocks in expansions and timing the market in recessions. The prerequisite for an investment manager to outperform the market is to have insights about the market. My paper suggests that as the information delivered to fund investors indeed contains valuable information to predict market returns, it can be inferred that investment managers indeed have capabilities to understand the market and make informative investments.

4 Method

Textual documents come to econometricians in the format as strings of words, and we have to quantify the textual documents for downstream statistical analysis.

Many machine learning algorithms require the input to be represented as a fixed-length feature vector. In textual analysis, one of the most common fixed-length features is bag-of-words. Bag-of-words is popular because of its simplicity and robustness in many NLP applications. However, the bag-of-words model has two major weakness. First, the order information of a word is lost, and thus two different sentences could have the same representations. Although a derivation of the bag-of-words model, the bag-of-n-grams model, incorporates some local order information into the vector representation of a document, it suffers from data sparsity and high dimensionality. Second, the semantic information contained in a document is lost in a bag-of-words representation. For example, in a financial report corpus, a pair of words like "stock" and "returns" should be semantically closer to each other than a pair of words like "stock" and "Africa", because "stock" and "returns" are more likely to appear together. However, in a bag-of-words model, the three words are equally distant from each other.

To overcome the shortcomings of the bag-of-words model, a collection of word embedding models are proposed in the computational linguistic literature (Bengio et al. (2006); Collobert & Weston (2008); Mnih & Hinton (2009); Turian et al. (2010); Mikolov et al. (2013b,a); Tang et al. (2014)). The idea is to map words or phrases into a low dimensional vector space such that semantic similarity between words can be measured using vector distances.

4.1 CBOW

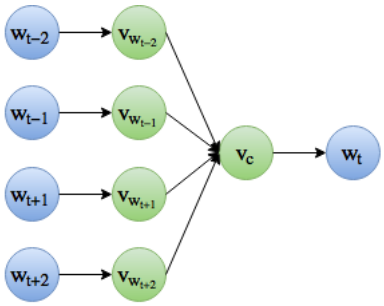
The CBOW word embedding model is a neural network model introduced by Mikolov et al. (2013b). It provides an efficient method for learning high-quality vector representations of words from large amounts of unstructured text data and has achieved great popularity in the computational linguistics. The idea of CBOW is to find word vector representations that are useful for predicting a target word using surrounding words in a paragraph. The architecture of CBOW is shown in Figure 1, which is essentially a convolutional neural network. Each surrounding word as an input is mapped to a word embedding vector, the average of surrounding word vectors forms the context vector, based on which we predict the target word.

More formally, using the notation in Levy & Goldberg (2014), Denote the vocabulary set of words in a corpus as V_W, and the set of contexts V_C. In CBOW, the contexts for word w_t are the surrounding words in a window with length $2l$: $c_t = (w_{t-l}, ..., w_{t-1}, w_{t+1}, ..., w_{t+l})$, containing l words ahead of w_t, and l words following w_t. Denote D as the set of observed words and context pairs. Consider a word-context pair (w, c), and let $p\left(w\middle|c\right)$ be the probability that w appears in context c.

The idea of CBOW is to associate each word $w \in V_W$ with a vector $v_w \in \mathbb{R}^r$, where r is the embedding's dimensionality, a hyper parameter chosen by researchers. And a context vector is $v_c = \frac{1}{2l}\sum_{i=1}^{l}(w_{t-i} + w_{t+i})$. The elements in the word vectors are latent parameters to be learned from the model. Denote $\#(w, c)$ as the counts of the pair (w, c) in D, $\#(w) = \sum_{c' \in V_C} \#(w, c')$ and $\#(c) = \sum_{w' \in V_w} \#(w', c)$ as the counts of w and c in D, respectively.

Figure 1: Architecture of CBOW

This figure demonstrates the neural network architecture of CBOW. Each word is mapped to a word embedding vector. The context vector is the average of surrounding word vectors. The distribution of a target word is determined by the inner product of its own embedding vector and the context vector.



In CBOW, the probability for a word w to appear in context c is modeled as a sigmoid function of the inner product of the word vector and context vector

$$p\left(w\middle|c\right) = \sigma\left(v_w \cdot v_c\right) \equiv \frac{1}{1+\exp\left(-v_w \cdot v_c\right)}.$$

The learning of CBOW employs the negative sampling technique, in which the objective for a single (w, c) is to maximize the average log probability

$$\log \sigma\left(v_w \cdot v_c\right) + k \cdot E_{w_N \sim P(w)} \sigma\left(-v_{w_N} \cdot v_c\right).$$

The idea of the objective function is to maximize $p\left(w\middle|c\right)$ for (w, c) that appears in the corpus, while minimizing $p\left(w_N\middle|c\right)$ for (w_N, c) not appearing in the corpus. k is the number of "negative" samples. When k is large, the objective puts more weight on penalizing unobserved (w_N, c) pairs; when k is small, the objectives puts more weight on maximizing the likelihood of observed (w, c) pairs. w_N denotes words drawn from the empirical distribution $P(w) = \frac{\#(w)}{|D|}$, the proportion of observed word w in set D. The global objective is to maximize the sum of the objective of single (w, c) pairs:

$$\mathcal{L} = \sum_{w \in V_W} \sum_{c \in V_C} \#(w, c) \frac{1}{T} \sum_{t=1}^{T} \left[\log \sigma\left(v_w \cdot v_c\right) + \sum_{i=1}^{k} E_{w_N \sim P(w)} \sigma\left(-v_{w_N} \cdot v_c\right) \right].$$

4.1.1 CBOW-Average

Training texts using CBOW only generates the embedding vectors for each word, but we need an embedding vector for each document in training downstream stock return prediction models. In CBOW-Average, a document vector is simply calculated as the average of the word vectors corresponding to words in the document. Otherwise explicitly specified, all the document vectors in this paper refer to vectors generated through CBOW-Average.

4.1.2 CBOW-Doc

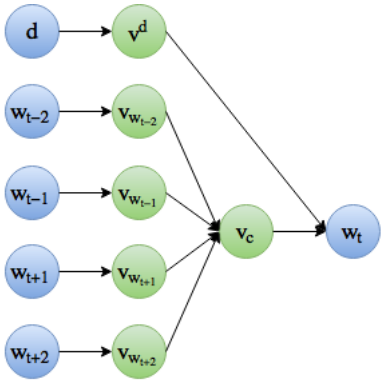
CBOW-Doc (Le & Mikolov (2014)) is a derivation of the original CBOW model, which directly encodes the co-occurrence of words and documents into the neural network structure and directly estimates a document vector. In CBOW-Doc, not only each word, but also each document is represented as a vector, and the probability for word w to appear in context c and document d is

$$p\left(w\middle|c, d\right) = \sigma\left(v_w \cdot \left(\alpha v_c + (1-\alpha) v_d\right)\right),$$

where $v_d \in \mathbb{R}^r$ is the vector representing document d, and $\alpha \in [0, 1]$ is the weight assigned to the context vector v_c in affecting word distributions. The architecture of the CBOW-Doc model is shown in Figure 2.

Figure 2: Architecture of CBOW-Doc

This figure demonstrates the neural network architecture of CBOW-Doc. Each word and document is mapped to an embedding vector. The context vector is the average of surrounding word vectors. The distribution of a target word is determined by the inner product of its own embedding vector and a weighted average of the context vector and document vector.



4.2 Matrix Factorization

The specification of the CBOW neural network has an intuition of coercing words surrounded by similar contexts to have similar embeddings. However, it does not provide intuition to understand the meanings of the embeddings. Levy & Goldberg (2014) justifies that neural word embedding can be considered as implicit word-context matrix factorization, and thus each dimension of the embedding spaces represents as a hidden topic of the corpus.

General word embedding models starts with a word-context matrix M. The process of learning word embedding vectors is to factorize the word-context matrix into a $|V_W| \times r$ word embedding matrix W and a $|V_C| \times r$ context embedding matrix C such that $M = W \cdot C'$, which embeds both words and their contexts into a low-dimensional space $\mathbb{R}^r$. Each row of W corresponds to a word, and each row of C corresponds to a context. Each element M_{wc} measures the association between a word and a context.

Levy & Goldberg (2014) proved CBOW is essentially factorizing a word-context matrix M that $M_{wc} = \log\left(\frac{\#(w,c)\cdot|D|}{\#(w)\cdot\#(c)}\right) - \log k$ in CBOW, and the procedure of maximizing the objective function $\mathcal{L}$ through stochastic gradient descent in Mikolov et al. (2013a) is similar to the symmetric singular value decomposition (SVD) of M. SVD factorizes M into the product of three matrices $U\Sigma V^T$, where the columns of U and V are the left and right singular vectors of M, and Σ is a diagonal matrix of singular values. Let Σ_r be the diagonal matrix containing the largest r singular values, and U_r, V_r be the matrices containing the corresponding singular vectors. The matrix $M_r = U_r\Sigma_r V_r^T$ is the matrix of rank r the best approximates M, measured in terms of Frobenius norm, $M_r = \arg\min_{Rank(M')=r} ||M' - M||^2_{Fro}$. The word embedding matrix W achieved by CBOW is similar to a symmetric SVD matrix $W^{SVD_{1/2}} = U_r \cdot \sqrt{\Sigma_r}$.

4.3 Predictive Model

After learning document embedding vectors from CBOW, I consider a linear predictive model $y = \beta_0 + \beta_X X + \beta_d v_d$, where X denotes the controlling variables and y is a general dependent variable. Because of the high dimensionality of v_d, I estimate the linear model using elastic-net, which penalizes the a convex combination of $L1$ and $L2$ norm of the parameters. The objective of elastic-net is

$$\min_{\beta_0,\beta_X,\beta_d} ||y - (\beta_0 + \beta_X X + \beta_d v_d)||_2^2 + \lambda\left[\rho\left(||\beta_X||_1 + ||\beta_d||_1\right) + (1-\rho)\left(||\beta_X||_2^2 + ||\beta_d||_2^2\right)\right]$$

where λ is the penalization parameter and ρ is the weight assigned to $L1$ norm. They are usually chosen through cross-validation.

5 Data

5.1 Letters to Shareholders

In the United States, the Securities and Exchange Commission (SEC) requires all registered management investment companies to file annual (N-CSR) and semiannual (N-CSRS) reports to shareholders. The N-CSR(S) files are publicly available from the SEC Edgar database, and the period covered is 2003-2014.

The N-CSR(S) files often start with a letter to shareholders written by the investment managers. However, the SEC only provides general instructions on N-CSR(S) filing, but there is no strict structured template for the companies to follow. Therefore, the structures of N-CSR(S) files across firms are heterogeneous, and there is no uniform boundaries between the letters and the rest part of a file. This fact makes extracting a certain section from the N-CSR(S) files much more challenging than extracting sections from well structured SEC files like 10-Ks, the corporate annual reports.

I extract the letters to shareholders through regular expression matching. As there is no separate section for letters to shareholders, I use the common letter starting words (e.g. "Dear Shareholders", "Letters to Shareholders", "Fellow Shareholders") to match the beginning of a letter and use ending words (e.g."Yours sincerely", "Respectfully","Best regards") to match the end of a letter. Table 5 shows the counts of the original N-CSR(S) files and the letters extracted from the original files, as well as the extraction rate, the proportion of letters extracted from the original files successfully. The total number of the N-CSR files is $37,862$, and the total number of letters extracted from the N-CSR files is $21,937$, with average extraction rate of 0.58. The total number of N-CSRS files is $31,139$, and the total number of letters extracted from the N-CSRS files is $15,077$, with average extraction rate of 0.48.

After extracting the letters from the N-CSR(S) files, following Kogan et al. (2009), I tokenize the letters in six steps: 1. Eliminate HTML markups; 2. Downcase all letters (convert A-Z to a-z); 3. Separate letter strings from other types of sequences; 4. Delete strings not a letter; 5. Clean up whitespace, leaving only one white space between tokens. 6. Remove stopwords.

The summary statistics for the length of the tokenized letters are shown in Table 6 in the Appendix. We can see that the average length of a tokenized letter contains about 500 words and the length varies a lot from letter to letter.

As multiple N-CSR(S) files may be filed on the same day, I concatenate the letters to shareholders written by different investment managers on the same day together and treat it as a single document. Because the my research question is to test whether a representative investment manager has insights about market performance, there is no need to identify individual managers. In addition,for CBOW, the word embedding vectors are learned based on the co-occurrence of words in the same sentence, and thus the concatenation does not impair the learning of the word embedding vectors. For CBOW-Doc, this may add bias to the estimation of the word and document vectors as the concatenation procedure creates some fake co-occurrence of some words and documents.

5.2 Stock Returns

The daily stock return data of the value-weighted market portfolio come from the Center for Research in Security Prices (CRSP) dataset.

CRSP provides the market portfolio return data both including ($vwretd$) and excluding ($vwretx$) dividends. Denote the price of the market portfolio at time t as P_t, and its dividend as D_t. The market portfolio returns including and excluding dividends from period $t-1$ to t are $vwretd_t = (P_t + D_t)/P_{t-1} - 1$ and $vwretx_t = P_t/P_{t-1} - 1$ respectively. Therefore, the dividend yield $\log(D_t/P_t)$ can be constructed as

$$dividend\_yield_t = \log\left(\frac{1 + vwretd_t}{1 + vwretx_t} - 1\right).$$

To test whether the document vectors contains information in predicting the stock returns of the market portfolio. I use the document vector at date t, to predict the annual excess return of the market portfolio, which is calculated as the accumulated returns from $t+1$ to $t+252$. The excess

Table 1: Similar Words
This table demonstrates the top 10 similar words to "china", "oil", "politics" and "shareholder". The similarity between 2 words are measured as the cosine similarity of their word embedding vectors.

	china	oil	politics	shareholder
1	chinese	commodity	terrorism	shareholders
2	indonesia	energy	rhetoric	stockholders
3	brazil	gasoline	political	stockholder
4	russia	cotton	standoff	shareowner
5	japan	fuel	presidential	trustees
6	asia	gold	partisan	shareowners
7	turkey	brent	debate	classify
8	states	natural	threats	directors
9	population	food	uncertainties	mergers
10	india	ore	attacks	semiannual

return is gross return ($vwretd$) minus the risk-free rate. The risk-free is proxied by the interest rate of 3-Month Treasury Bills in this paper. The controlling variables are $dividend\_yield_t$ and $return\_leading_t$, where $return\_leading_t$ is the leading annual stock return of the value-weighted market portfolio, which is calculated as the accumulated returns from $t - 251$ to t.

The value and momentum factors are two of the most popular pricing factors in the asset pricing literature and are found to explain a significant proportion of variations in the cross-section of stock returns (Fama & French (1993); Carhart (1997)). In the market portfolio time series predictions, $dividend\_yield_t$ captures value factor, and $return\_leading_t$ captures the momentum factor. They are found to have significant power in predicting long-term stock returns (Lettau & Ludvigson (2001); Cochrane (2011); Fama & French (1988)), and thus I include $dividend\_yield_t$ and $return\_leading_t$ in my predicting models as a controlling variables.

6 Analysis

6.1 Word Vectors

I apply the CBOW model using the Python module Gensim (Řehůřek & Sojka (2010)). Gensim provides Python interface to the Word2Vec software of Google which originally implemented the CBOW model. It is recommended to represent words in a relative high-dimension vector space in literature (Mikolov et al. (2013b)) to achieve accurate word embedding estimates. In practice, a common choice of the dimension is $150 - 400$. In this paper, I choose the embedding dimension to be 300 and length of the context window l to be equal to 2, meaning the context of a word contains 2 leading and 2 following words.

Examples showing the top similar words to a few seed words are listed in Table 1. For example, the top 10 words that have highest semantic similarity to the word "china" are "chinese", "indonesia", "brazil", "russia", "japan", etc., which is sensible as Indonesia and Japan are countries geographically close to China, and Brazil, Russia, India are often referred as Gold BRICS countries in financial documents. The topic 10 words that have closest semantic similarity to the word "oil" are "commodity", "energy", "gasoline", "cotton" etc., which is also reasonable because these words often appear together in letters to shareholders written by investment managers that focus on commodity trading.

6.2 Word Clouds

The nonlinear dimension reduction technique t-SNE (Van der Maaten & Hinton (2008)) is a powerful dimension reduction method to project the high-dimension word vectors into a low-dimension space such that we can visualize the word locations in a 2-d graph.

The visualization of some sentiment words are demonstrated in Figure 8 in the Appendix. To generate the positive and negative word lists, I use the keywords "good" and "bad" as seed words, and find 30 words that have the highest semantic similarity to them. We can see the splitting between positive words like "good", "excellent", "superior" and negative words "bad", "terrible", "discouraging", and words with the same sentiment are close to each other.

The visualization of words classified by economic topics are demonstrated in Figure 9 in the Appendix. I include eight topics in the graph: regions, politics, macroeconomy, market index, commodity, industry, investment and shareholder. To generate the word list for each topic, I use the keywords "region", "politics", "macroeconomy", "index", "commodity", "industry", "investment", "shareholder" as seed words, and find 30 words that have the highest semantic similarity to the seed word for each topic. For example, the words having closest semantic meaning to "commodity" include "gold", "oil", "electricity", "copper" etc; the words having closest semantic meaning to "region" include "china", "japan", "russian", "asia" etc; the words having closest semantic meaning to "politics" include "politicians", "democracy", "presidential", "legislative" etc. The word lists agree with our linguistic intuition.

The distributed location of the economic topic word clouds in Figure 9 also generate intuitive results. First of all, words close to each other in semantic meaning indeed locate close to each other. Second, topics that are supposed to have a close linguistic relationship also locate close to each other. For example, in news articles or financial reports, people often tie politics to a certain region, like wars in the mid-east or presidential elections in the United States. In the words clouds, we indeed see the "politics" topic located close to the "region" topic. When institutional investors make investments, the macroeconomic condition is an important factor affecting their investment decisions, and the "macro" and "investment" topic are indeed close to each other in the word clouds.

7 Results

7.1 Out-of-sample Predictions

For out-of-sample (OOS) predictions, I construct the training and test datasets in two ways, random splitting and rolling window splitting.

7.1.1 Random Splitting

For random splitting, I first pool all the 2,255 observations together, and randomly select 70% of the observations to form the training set, and use the rest 30% observations to form the test set. I consider five linear models, which include different explanatory variables: (1). "Constant", the explanatory variable include only a constant, which is equivalent to prediction using training set mean; (2). "Mom"; the explanatory variables include a constant and the momentum factor $return\_leading_t$; (3). "Mom&Val", the explanatory variables include a constant, the momentum factor $return\_leading_t$ and value variable $dividend\_yield_t$; (4). "CBOW-Average"; the explanatory variables include the document vectors generated using the CBOW-Average model in addition to the controlling variables in "Benchmark"; (5). "CBOW-Doc", the explanatory variables include the document vectors generated using the CBOW-Doc model in addition to the controlling variables in "Benchmark".

I estimate the linear models using elastic-net. The penalization parameter of the elastic-net is selected through 10-fold cross validation. I measure of the prediction accuracy using OOS RMSEs.

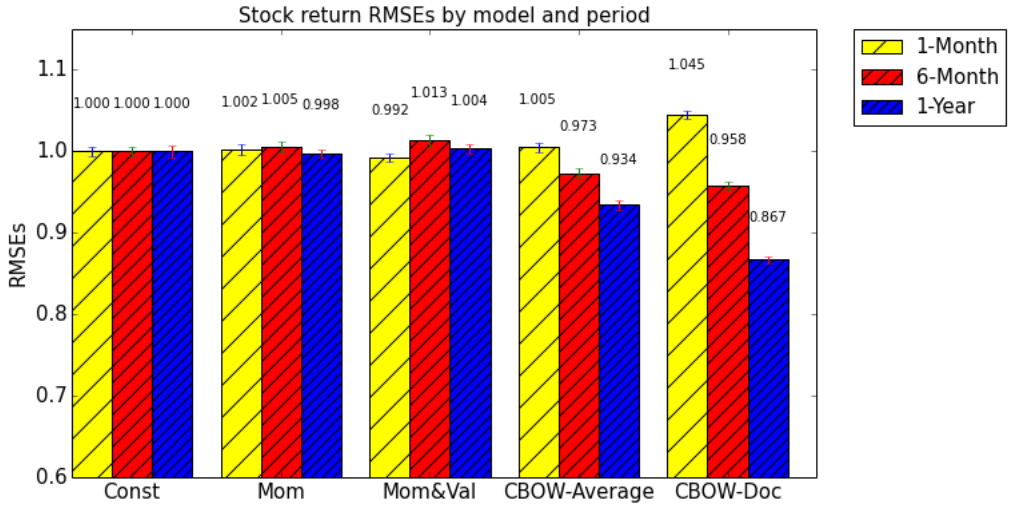
To reduce the random effect of the training-test set splitting, I follow a bootstrap approach by repeating the training-test splitting for 50 times. The OOS prediction RMSEs of the five models are shown in Figure 3. In addition to 1-year returns, I also checked the power of textual features in predicting 1-month and 6-month returns (the corresponding momentum factor $return\_leading_t$ is adjusted accordingly). To make the results for returns of different horizons comparable, I normalize the OOS RMSEs of the Const model to be equal to 1, and scale the RMSEs generated by other models correspondingly. In the bar chart, the height of the bars represent the average OOS RMSEs of the 50 experiments, and the standard errors are also demonstrated through the error bars.

We can see that by including document vectors generated by CBOW-Average in the stock return prediction model, we can reduce the OOS RMSEs by about 1.0% for 1-month returns, 2.7% for

6-month returns, and 6.6% for 1-year returns, in comparison to predicting using training set mean. It means that the textual features generated by CBOW indeed contains valuable information in predicting future stock returns, and the prediction power increases with the length of horizon. The textual features generated by CBOW-Doc is more powerful in predicting long-term stock returns, but it underperforms in predicting short-term returns.

Figure 3: OOS prediction RMSEs with random training-test splitting

This figure shows the OOS prediction RMSEs of five linear models estimated by elastic-net: (1). "Const", the explanatory variable include only a constant, which is equivalent to prediction using training set mean; (2). "Mom"; the explanatory variables include a constant and the momentum variable $return\_leading_t$; (3). "Mom&Val", the explanatory variables include a constant, the momentum factor $return\_leading_t$ and the value factor $dividend\_yield_t$; (4). "CBOW-Average"; the explanatory variables include the document vectors generated using the CBOW-Average model in addition to the controlling variables in "Mom&Val"; (5). "CBOW-Doc", the explanatory variables include the document vectors generated using the CBOW-Doc model in addition to the controlling variables in "Mom&Val". The training set is constructed by randomly selecting 70% observations from the whole sample set, and the rest 30% observations forms the test set. To avoid random splitting effect, this procedure is repeated 50 times. The height of the bars represents the average RMSEs of the 50 experiments, and the standard errors are shown through the error bars. The RMSEs generated by Const is normalized to 1.



7.1.2 Rolling Window Splitting

One possible concern about the forecasting results presented above is the potential "look-ahead" bias due to the fact the training set contains information in the future. This concern can be addressed by forming the training and test set in a rolling window basis and performing OOS forecasts where the parameters in the linear model are re-estimated every period, using only data available at the time of the forecast.

I consider rolling windows with length equal to six years. In every window, I use observations in the leading five years to form the training set to estimate the model parameters, and make predictions in the sixth year to calculate the OOS RMSEs. The RMSEs of the five models are shown in Table 2. As the data set covers the period 2003-2014, the first 6-year window is 2003-2008, and thus the RMSEs reported in the table starts from the year 2008.

We can see that "CBOW-Average" achieves the best rolling window OOS prediction performance. Overall, the improvement in the prediction accuracy by incorporating the document vectors into the explanatory variables is smaller in the rolling window training-test splitting approach in comparison to the random splitting approach. A possible explanation is that the correlations between the textual information in the letters to shareholders and market portfolio stock returns vary over time. Therefore, in the rolling window split approach, the linear model is more likely to overfit historical patterns. This point may justify the fact that CBOW-Doc is outperformed by the CBOW-Average in the rolling window approach, although it performs best in predicting annual stock returns when we

Table 2: OOS prediction RMSEs with rolling window training-test splitting

This table shows the OOS prediction RMSEs of five linear models based on elastic-net: (1). "Const", the explanatory variable include only a constant, which is equivalent to prediction using historical mean; (2). "Mom"; the explanatory variables include a constant and the momentum factor $return\_leading_t$; (3). "Mom&Val", the explanatory variables include a constant, the momentum factor $return\_leading_t$ and the value factor $dividend\_yield_t$; (4). "CBOW-Average"; the explanatory variables include the document vectors generated using the CBOW-Average model in addition to the controlling variables in "Mom&Val"; (5). "CBOW-Doc", the explanatory variables include the document vectors generated using the CBOW-Doc model in addition to the controlling variables in "Mom&Val". The training set is constructed in a rolling window basis. In every 6-year window, I estimate the parameters using observations in the first five years, and make predictions in the sixth year.

Year	Const	Mom	Mom&Val	CBOW-Average	CBOW-Doc
2008	0.349	0.375	0.376	0.328	0.290
2009	0.327	0.355	0.355	0.360	0.385
2010	0.149	0.275	0.275	0.135	0.131
2011	0.097	0.098	0.098	0.096	0.137
2012	0.186	0.196	0.196	0.175	0.171
2013	0.090	0.107	0.106	0.090	0.124
2014	0.128	0.111	0.111	0.116	0.118

split the dataset into a training set and a test set randomly. Because the word vectors built through CBOW-Average are solely based on co-occurrence of neighboring words, which do not depend on document level information which may contain time-varying text patterns, and thus CBOW-Average is less likely to overfit.

7.2 Other Language Models

In this section, I compare the CBOW-Average and CBOW-Doc results with five other language models, CBOW with clustering (k-means and spectral clustering), Sentiment Words Counting, Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA).

7.2.1 CBOW with Clustering

In the discussions above, the document vectors in CBOW-Average is calculated as the average of the word vectors. As CBOW provides a way to identify clusters of semantically related words through their embedding vectors, another way to exploit word similarities is to cluster words based on their locations in the embedding vector space, and to represent a document using a bag-of-clusters. I consider two clustering algorithms, k-means (CBOW-Kmeans) and spectral clustering (CBOW-Spectral). The advantage of representing documents using clusters is to reduce the idiosyncratic noises introduced by each word.

In both k-means and spectral clustering[3], I first classify the words into 20 clusters based their word vectors. Then I quantify each document using a bag-of-clusters model, where each document is represented as a 20-dimension vector, with each entry of the vector corresponds to a unique word cluster, and the value of each element is the counts of the words in the corresponding cluster.

7.2.2 Sentiment_Counting

The concurrent popular approach of textual analysis in the financial economics literature rely on a word counting approach based on pre-built sentiment dictionaries (Tetlock (2007); Tetlock et al. (2008); Loughran & McDonald (2011); Jegadeesh & Wu (2013)). Therefore, I also the test the return predictive power of two sentiment measures $negative_t$ and $positive_t$, which are calculated as the proportion of negative and positive words in the concatenated letter on day t, where the negative and positive words are classified using the Loughran & McDonald (2011) sentiment dictionaries.

[3]I use the Python module Scikit-Learn to implement k-means and spectral clustering, and the module Gensim to implement LSA and LDA.

7.2.3 Latent Semantic Analysis

LSA (Dumais (2004)) is a method for discovering hidden topics in a document data. LSA is essentially the singular value decomposition of the word-document matrix that represents a bag-of-words model using matrix notation. LSA is popularly used to reduce the dimension of the bag-of-words model and has a long history of applications in the information retrieval literature.

I use LSA to recover 20 hidden topics from the corpus. Each document is represented as a 20-dimension vector, with each entry of the vector corresponding to a hidden topic, and the value of each entry represents the loading on a hidden concept of the document. Sample topics generated by LSA is shown in Figure 10 in the Appendix.

7.2.4 Latent Dirichlet Allocation

LDA (Blei et al. (2003)) is a three-level hierarchical Bayesian Network model that describes the data generating process of textual documents. The idea of LDA is that documents are represented as random mixtures over latent topics, where each topic is characterized by a random distribution over words. Since introduction, LDA is popularly used in learning the hierarchical structures of documents and reducing the dimension of a bag-of-words model.

I use LDA to construct 20 topics from the corpus. Similar to LSA, each document is represented as a 20-dimension vector, with each entry of the vector corresponding to a topic, and the value of each entry represents the proportion of words in the topic. Sample topics generated by LSA is shown in Figure 11 in the Appendix.

The OOS prediction RMSEs comparing different language models are shown in Figure 4 (random training-test splitting) and Table 3 (rolling-window training-test splitting). We can see that CBOW-Average and CBOW-Doc generate smaller OOS prediction RMSEs than features generated using other language models in most cases in the random training-test splitting, and in most years in the rolling-window training-test splitting.

Table 3: OOS RMSEs, CBOW vs. other language models, rolling window

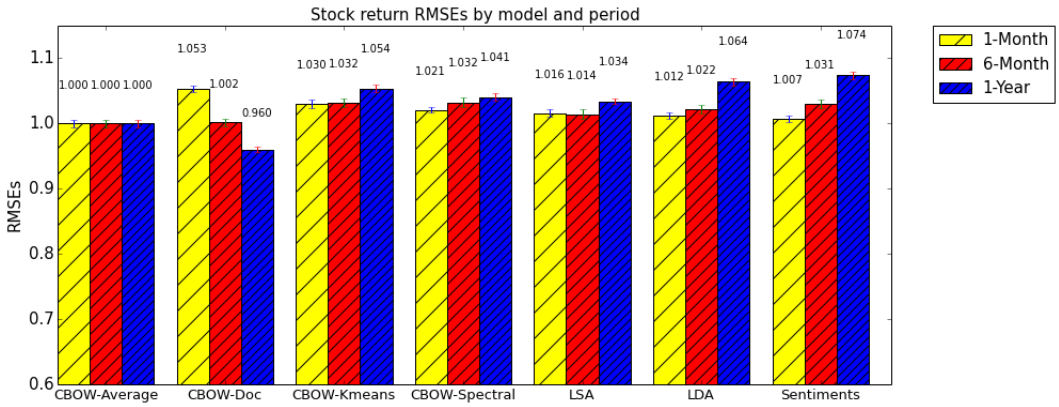
This table compares the elastic-net OOS prediction RMSEs between models using document vectors generated using CBOW-Average/CBOW-Doc with models using features generated using other language models. In CBOW-Average, a document vector is the average of the word embedding vectors for all individual words appearing in the document. In CBOW-Doc, the document vectors are directly estimated from the neural network model. In both k-means (CBOW-Kmeans) and spectral clustering (CBOW-Spectral), I first classify the words into 20 clusters based their CBOW word vectors, and then I quantify each document using a bag-of-cluster model, where each document is represented as a 20-dimension vector, with each element of the vector corresponds to a unique word cluster, and the value of each element is the counts of the words in the corresponding cluster. In LSA, the document features are loadings on 20 hidden topics recovered by singular value decomposition of term-document matrix. In LDA, the document features are distributions over 20 hidden topics learned from hierarchical structure of the documents. In Sentiments, the document features are the proportion of negative words and positive words based the Loughran & McDonald (2011) sentiment word classification dictionary. The training set is constructed by in a rolling window basis. In every 6-year window, I estimate the parameters using observations in the first five years, and make predictions in the sixth year.

Year	CBOW-Average	CBOW-Doc	CBOW-Kmeans	CBOW-Spectral	LSA	LDA	Sentiments
2008	0.328	0.290	0.327	0.327	0.344	0.346	0.342
2009	0.360	0.385	0.374	0.353	0.369	0.329	0.377
2010	0.135	0.131	0.150	0.151	0.171	0.150	0.150
2011	0.097	0.137	0.110	0.128	0.140	0.103	0.089
2012	0.175	0.171	0.186	0.183	0.160	0.188	0.191
2013	0.094	0.124	0.102	0.113	0.135	0.089	0.100
2014	0.116	0.118	0.117	0.119	0.110	0.128	0.121

I also check the power of textual features in predicting stock return volatilities and macroeconomic factors. The results generated by models estimated by elastic-net are shown in Figure 5 and Figure 6 in the appendix. I find that including textual features into the predicting models can reduce the OOS prediction RMSEs significantly, implying that the investment manager discussions also contain valuable information in predicting stock return volatilities and macroeconomic conditions.

Figure 4: OOS RMSEs, CBOW v.s. other language models, random splitting

This figure compares the elastic-net OOS prediction RMSEs between models using document vectors generated using CBOW-Average/CBOW-Doc with models using features generated using other language models. In CBOW-Average, a document vector is the average of the word embedding vectors for all individual words appearing in the document. In CBOW-Doc, the document vectors are directly estimated from the neural network model. In both k-means (CBOW-Kmeans) and spectral clustering (CBOW-Spectral), I first classify the words into 20 clusters based their CBOW word vectors, and then I quantify each document using a bag-of-cluster model, where each document is represented as a 20-dimension vector, with each element of the vector corresponds to a unique word cluster, and the value of each element is the counts of the words in the corresponding cluster. In LSA, the document features are loadings on 20 hidden topics recovered by singular value decomposition of term-document matrix. In LDA, the document features are distributions over 20 hidden topics learned from hierarchical structure of the documents. In Sentiments, the document features are the proportion of negative words and positive words based the Loughran & McDonald (2011) sentiment word classification dictionary. The training set is constructed by randomly selecting 70% observations from the whole sample set, and the rest 30% observations forms the test set. To avoid random splitting effect, training-test split is repeated 50 times. The height of the bars represents the average RMSEs of the 50 experiments, and the standard errors are shown through the error bars. The RMSEs generated by CBOW-Average is normalized to 1.



8 Discussion

In this section, the aim is to understand the economic foundation that explains why investment manager discussions contain information in predicting stock returns.

Financial economists find that long-term stock returns are predictable. In particular, numerous studies report that predictive part of the stock returns is risk premium (Pástor & Stambaugh (2009); Cochrane (2008); Campbell & Shiller (1988)). According to standard asset pricing theory, risk premium is determined by the degree of the risk aversion of a representative investor. The degree of risk aversion, which reflects the subjective opinions of an investor, is often difficult to be measured accurately in practice. However, the textual data of investment manager discussions, which incorporates subjective mental information of the investors, provide a unique source to measure risk aversion.

I constructed two measures of risk aversion based on the textual data. The first measure $uncertain$ is the proportion (in percentage) of top 100 words having closest semantic meaning (highest word vector cosine similarity) to the word "uncertain" (the full list of words related to the seed words "uncertain", "risk" and "recession" are shown in Table 7, 8 and 9 in the Appendix). In theoretical works, economists usually distinguish uncertainty aversion and risk aversion (Dow & da Costa Werlang (1992)). Risk describes unknown outcomes whose odds of happening can be measured or learned about, while uncertainty refers to events that we do not know how to describe. However, in empirical works, the distinction between risk uncertainty is subtle, and researchers often ignore it. I adopted the notation of empiricists, where "risk aversion" referred by empiricists often included both risk aversion and uncertainty aversion, and the empirically measured risk premium often include a premium for both risk aversion and uncertainty aversion.

Although I use "uncertain" here as the seeding word, the word list I generate does not exclusively measure uncertainty aversion only. Checking the full list of words related to "uncertain", based on

Table 4: Risk aversion and stock returns

This table reports the in-sample OLS regression results. The dependent variable $return$ is the annual stock returns calculated for the 252-day period starting from the day following the release date of N-CSR(S) files. $return\_leading$ is the annual stock returns calculated for the 252-day period ending at the release date of the N-CSR(S) files. $dividend\_yield$ is the dividend yield, which is log of the dividend to price ratio. $uncertain$ is the proportion of the top 100 words that having the closest semantic relationship with the seeding word "uncertain", and $recession$ is the proportion of the top 100 words that having the closest semantic relationship with the seeding words "recession". The Newey-West Newey & West (1986) HAC robust t-statistics are shown below the estimated coefficient. t-statistics significant at the 1% level are shown in bold.

	Const + Risk Aversion			Mom + Risk Aversion			Mom&Val + Risk Aversion		
$intercept$	0.062	0.077	0.057	0.076	0.090	0.072	0.112	0.094	0.090
t-stat	**4.590**	**6.298**	**4.280**	**3.751**	**5.857**	**3.581**	**4.466**	**3.523**	**3.412**
5% CI	(0.050, 0.073)	(0.063, 0.090)	(0.045, 0.070)	(0.064, 0.088)	(0.072, 0.104)	(0.055, 0.088)	(0.076, 0.153)	(0.058, 0.134)	(0.051, 0.132)
$return\_leading$				-0.120	-0.125	-0.120	-0.125	-0.119	-0.120
t-stat				-1.371	-1.426	-1.372	-1.420	-1.367	-1.368
5% C.I.				(-0.176, -0.066)	(-0.178, -0.062)	(-0.181, -0.068)	(-0.178, -0.072)	(-0.172, -0.069)	(-0.172, -0.063)
$dividend\_yield$							0.230	0.190	0.188
t-stat							1.381	1.156	1.141
5% C.I.							(-0.106, 0.596)	(-0.146, 0.558)	(-0.143, 0.56)
$uncertain$	0.050		0.048	0.047		0.055		0.047	0.044
t-stat	**4.617**		**4.285**	**4.354**		**4.001**		**4.311**	**3.963**
5% C.I.	(0.033, 0.068)		(0.029, 0.066)	(0.032, 0.062)		(0.026 0.066)		(0.028, 0.065)	(0.027, 0.065)
$recession$		0.011	0.005		0.010	0.005	0.010		0.005
t-stat		**2.398**	1.026		**2.347**	1.017	**2.327**		1.009
5% C.I.		(0.002, 0.020)	(-0.003, 0.013)		(0.003, 0.019)	(-0.004, 0.015)	(0.001, 0.018)		(-0.005 0.014)
R^2_{adj}	0.012	0.002	0.012	0.027	0.019	0.027	0.027	0.019	0.027
$Obs.$	2255	2255	2255	2255	2255	2255	2255	2255	2255

linguistic intuition, it reasonably contains both risk aversion and uncertainty aversion information. I do not use "risk" as the seeding word because many words in the list related to "risk" does not have clear risk aversion related interpretation.

The second measure $recession$ is the proportion (in percentage) of top 100 words having closest semantic meaning to the word "recession". Previous literature on asset pricing found that risk aversion correlated with business cycles (Campbell (1999), Boldrin et al. (2001)). In particular, investors usually require a high risk premium. Therefore, when investors start to talk more about recessions, we can expect the future stock return to be higher.

The OLS regressions results are shown in Table 4. We can see that when regressing $return$, the annual stock returns post the release date of N-CSR(S) files on $uncertain$ and $recession$, both measures of risk aversion predict high returns in the future, which agrees with our economic intuition that when aversion is high, the expected stock returns is high, implying high risk premium. In particular, I consider three benchmark models controlling different variables, the momentum factor $return\_leading$ and the value factor $dividend\_yield$.

Below the coefficients in the Table, I demonstrate the Newey & West (1986) robust t-test and 5% confidence intervals constructed through bootstrapping of 1,000 times. All three models generate similar estimates and significant level for $uncertain$ and $recession$, indicating that the information contained in these two measures is orthogonal to the momentum measure $return\_leading$ and value measure $dividend\_yield$.

When we include $uncertain$ and $recession$ separately, both measures are statistically significant. When $uncertain$ increases by 1 unit, meaning when the proportion of the words related to "uncertain" increases by 1%, the expected future annual stock returns increases by 5%, which is economically significant. When $recession$ increases by 1 unit, meaning when the proportion of the words related to "recession" increases by 1%, the expected future annual stock returns increase by 1%. When we include both $uncertain$ and $recession$, only $uncertain$ is significant, which implies the collinearity between $uncertain$ and $recession$. I find the correlation between $uncertain$ and $recession$ is 0.257, indicating that $uncertain$ and $recession$ indeed contains common information.

9 Limitations

In the previous sections, all the documents in the corpus are used to learn the word embedding vectors. The advantage of this approach is that the estimation of word vectors is more accurate given more observations, and there is no out-of-vocabulary problem because the embedding vector

for every word in the corpus is learned through the neural network. The disadvantage is that to estimate the embedding vector for a document in a given period, future textual information is used, making it difficult to justify the causality between the textual features and document and stock returns. Therefore, in this subsection, I consider learning word embedding vectors in a rolling window basis. For a given year, I use all the documents in all the previous years to learn the word vectors, based on which we can calculate the document vectors as average of word vectors for words in the documents. Note that this approach is only applicable to CBOW-Average because in CBOW-Doc, we have to use the documents in the current period to learn the word and document vectors.

Compared with the results in Figure 3, Figure 7 in the Appendix shows that the OOS prediction RMSEs generated by CBOW-Average when the document vectors are learned in a rolling window basis is much smaller than the RMSEs generated by CBOW-Average when the document vectors are learned using all available documents. However, compared with the results in Table 2, Table 12 in the Appendix demonstrates that the RMSEs generated by CBOW-Average when the document vectors are learned in a rolling window basis is larger than the RMSEs generated by CBOW-Average when the document vectors are learned using all available documents.

The above results mean that when we learn the word embedding vectors using only historical documents, the embedding vectors are more likely to overfit history and thus leading to inaccurate predictions for future periods. It implies that there is time varying patterns of language usage for investment managers, and thus some time varying information is lost when we learn word embedding vectors using static models like CBOW.

10 Conclusion

In this paper, I construct a textual dataset containing 37,014 letters to shareholders written by investment managers to test whether investment managers discussions contain useful information in predicting market aggregate stock returns. I quantify the textual documents using the CBOW neural network word embedding model introduced in Mikolov et al. (2013a), which represents words and documents in a low-dimensional vector space. My out-of-sample prediction results using elastic-net show that the investment manager discussions indeed provide valuable information in predicting stock returns, stock return volatilities, as well as the growth rates of oil price, dollar index and default spreads. I find that the textual data reveals information about the degree of risk aversion of institutional investors, which agrees with previous literature in asset pricing that risk premium is predictable.

11 Future

The dataset of letters to shareholders provides a unique source to quantify subjective opinions of investment managers. In this paper, I investigated the relationship between investment managers discussions and stock returns. Related topics for future research include the relationship between fund performance and fund capital flow. Fund manager discussions may manipulate investors' interpretation of fund performance, which may explain why we observe different fund flow activities among funds with similar performance.

12 Acknowledgements

I thank Alan Montgomery, Bryan Routledge, Geoff Gordon, Roy Maxion, Stefano Sacchetto, Steve Karolyi, and Yiming Yang for very helpful guidance and insightful suggestions.

References

Ang, Andrew and Bekaert, Geert. Stock return predictability: Is it there? *Review of Financial studies*, 20(3):651–707, 2007.

Bengio, Yoshua, Schwenk, Holger, Senécal, Jean-Sébastien, Morin, Fréderic, and Gauvain, Jean-Luc. Neural probabilistic language models. In *Innovations in Machine Learning*, pp. 137–186. Springer, 2006.

Blei, David M, Ng, Andrew Y, and Jordan, Michael I. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

Boldrin, Michele, Christiano, Lawrence J, and Fisher, Jonas DM. Habit persistence, asset returns, and the business cycle. *American Economic Review*, pp. 149–166, 2001.

Brands, Simone, Brown, Stephen J, and Gallagher, David R. Portfolio concentration and investment manager performance*. *International Review of Finance*, 5(3-4):149–174, 2005.

Campbell, John Y. Asset prices, consumption, and the business cycle. *Handbook of macroeconomics*, 1:1231–1303, 1999.

Campbell, John Y and Shiller, Robert J. The dividend-price ratio and expectations of future dividends and discount factors. *Review of financial studies*, 1(3):195–228, 1988.

Campbell, John Y and Yogo, Motohiro. Efficient tests of stock return predictability. *Journal of financial economics*, 81(1):27–60, 2006.

Carhart, Mark M. On persistence in mutual fund performance. *The Journal of finance*, 52(1):57–82, 1997.

Cochrane, John H. The dog that did not bark: A defense of return predictability. *Review of Financial Studies*, 21(4):1533–1575, 2008.

Cochrane, John H. Presidential address: Discount rates. *The Journal of Finance*, 66(4):1047–1108, 2011.

Collobert, Ronan and Weston, Jason. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pp. 160–167. ACM, 2008.

Cremers, KJ Martijn and Petajisto, Antti. How active is your fund manager? a new measure that predicts performance. *Review of Financial Studies*, pp. hhp057, 2009.

Dow, James and da Costa Werlang, Sergio Ribeiro. Uncertainty aversion, risk aversion, and the optimal choice of portfolio. *Econometrica: Journal of the Econometric Society*, pp. 197–204, 1992.

Dumais, Susan T. Latent semantic analysis. *Annual review of information science and technology*, 38(1):188–230, 2004.

Edwards, Franklin R and Caglayan, Mustafa Onur. Hedge fund performance and manager skill. *Journal of Futures Markets*, 21(11):1003–1028, 2001.

Fama, Eugene F and French, Kenneth R. Dividend yields and expected stock returns. *Journal of financial economics*, 22(1):3–25, 1988.

Fama, Eugene F and French, Kenneth R. Common risk factors in the returns on stocks and bonds. *Journal of financial economics*, 33(1):3–56, 1993.

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

Kacperczyk, Marcin, NIEUWERBURGH, STIJN VAN, and Veldkamp, Laura. Time-varying fund manager skill. *The Journal of Finance*, 69(4):1455–1484, 2014.

Kogan, Shimon, Levin, Dimitry, Routledge, Bryan R, Sagi, Jacob S, and Smith, Noah A. Predicting risk from financial reports with regression. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 272–280. Association for Computational Linguistics, 2009.

Le, Quoc V and Mikolov, Tomas. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*, 2014.

Lettau, Martin and Ludvigson, Sydney. Consumption, aggregate wealth, and expected stock returns. *the Journal of Finance*, 56(3):815–849, 2001.

Levy, Omer and Goldberg, Yoav. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, pp. 2177–2185, 2014.

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

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

Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Greg S, and Dean, Jeff. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119, 2013b.

Mnih, Andriy and Hinton, Geoffrey E. A scalable hierarchical distributed language model. In *Advances in neural information processing systems*, pp. 1081–1088, 2009.

Newey, Whitney K and West, Kenneth D. A simple, positive semi-definite, heteroskedasticity and autocorrelationconsistent covariance matrix, 1986.

Pástor, L'uboš and Stambaugh, Robert F. Predictive systems: Living with imperfect predictors. *The Journal of Finance*, 64(4):1583–1628, 2009.

Řehůřek, Radim and Sojka, Petr. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pp. 45–50, Valletta, Malta, May 2010. ELRA.

Tang, Duyu, Wei, Furu, Yang, Nan, Zhou, Ming, Liu, Ting, and Qin, Bing. Learning sentiment-specific word embedding for twitter sentiment classification. In *ACL (1)*, pp. 1555–1565, 2014.

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

Tetlock, Paul C, SAAR-TSECHANSKY, MAYTAL, and Macskassy, Sofus. More than words: Quantifying language to measure firms' fundamentals. *The Journal of Finance*, 63(3):1437–1467, 2008.

Turian, Joseph, Ratinov, Lev, and Bengio, Yoshua. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pp. 384–394. Association for Computational Linguistics, 2010.

Van der Maaten, Laurens and Hinton, Geoffrey. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

Table 5: Document counts and letter extraction rates

This table shows the numbers of N-CSR(S) files and letters to shareholders extracted from those files in each year. The extraction rate is the proportion of letters extracted from the N-CSR(S) files.

	N-CSR			N-CSRS		
	Files	Letters	Extraction Rate	Files	Letters	Extraction Rate
2003	2801	1527	0.55	984	460	0.47
2004	3968	2338	0.59	2615	1286	0.49
2005	3527	2154	0.61	2727	1415	0.52
2006	3361	1997	0.59	2738	1353	0.49
2007	3326	1967	0.59	2806	1248	0.44
2008	3293	1989	0.60	2787	1270	0.46
2009	3163	1938	0.61	2743	1345	0.49
2010	2927	1619	0.55	2772	1360	0.49
2011	2964	1652	0.56	2720	1393	0.51
2012	2832	1554	0.55	2742	1333	0.49
2013	2851	1583	0.56	2723	1307	0.48
2014	2849	1619	0.57	2782	1307	0.47
Total	37862	21937	0.58	31139	15077	0.48

Table 6: Letter length summary statistics

This table shows the summary statistics of the length (number of words) of the tokenized letters in each year. Count is the number of letters extracted from N-CSR(S) files in each year. Mean is the average number of words in the letters. Std is the standard deviation of the letter lengths. $X\%$ are the X percentile of the letter lengths.

Year	Count	Mean	Std	5%	50%	95%
2003	1987	407	548	67	255	1169
2004	3624	413	554	99	262	1091
2005	3569	451	591	98	294	1265
2006	3350	428	570	85	258	1213
2007	3215	463	608	99	274	1380
2008	3259	523	640	67	340	1633
2009	3283	515	601	75	312	1556
2010	2979	472	576	78	267	1511
2011	3045	487	597	73	285	1555
2012	2887	482	560	60	295	1516
2013	2890	515	843	60	305	1625
2014	2926	496	585	57	305	1622

Table 7: Risk aversion words

This table demonstrates words related to risk aversion. The three word lists are generated by three seeding words: "recession", "risk" and "uncertain". Each list contains 100 words with highest semantic similarity with the seeding words, where the semantic similarity of a pair of words is measured as the cosine similarity of their word embedding vectors.

	recession		risk		uncertain	
	Related Word	Similarity	Related Word	Similarity	Related Word	Similarity
1	depression	0.556	risks	0.481	unsettled	0.507
2	slump	0.499	risky	0.426	challenging	0.504
3	contraction	0.495	riskier	0.398	turbulent	0.487
4	recessions	0.483	volatility	0.374	unstable	0.451
5	downturn	0.478	quality	0.371	skeptical	0.445
6	slowdown	0.451	beta	0.320	unclear	0.445
7	crisis	0.440	swaps	0.314	uncertainty	0.440
8	deflation	0.439	coupons	0.309	tough	0.435
9	officially	0.409	sensitivity	0.306	cloudy	0.434
10	crunch	0.396	yielding	0.305	constructive	0.423
11	recessionary	0.396	potential	0.295	uncertainties	0.421
12	correction	0.390	exposure	0.295	evolving	0.413
13	patch	0.381	seasonally	0.286	vigilant	0.411
14	economists	0.374	potentially	0.283	accommodating	0.409
15	contagion	0.372	float	0.279	fragile	0.406
16	wwii	0.370	flexibility	0.272	changing	0.403
17	recovery	0.370	attractiveness	0.272	cautious	0.397
18	winter	0.368	safety	0.268	flux	0.394
19	mess	0.367	probability	0.267	sanguine	0.393
20	collapse	0.364	defensive	0.262	tenuous	0.382
21	meltdown	0.361	traditional	0.259	murky	0.381
22	sars	0.361	thereby	0.259	choppy	0.379
23	epidemic	0.360	correlation	0.259	dangerous	0.374
24	catastrophe	0.352	compensate	0.255	stormy	0.372
25	shock	0.352	conviction	0.255	perplexing	0.371
26	war	0.352	likelihood	0.255	mindful	0.370
27	storm	0.352	option	0.248	optimistic	0.368
28	technically	0.352	rated	0.247	clouded	0.366
29	landing	0.349	exposures	0.245	adapting	0.364
30	deflationary	0.349	fluctuation	0.245	confusing	0.362
31	economy	0.347	actively	0.243	tense	0.359
32	breakup	0.346	willing	0.242	volatile	0.353
33	malaise	0.346	environment	0.242	unsettling	0.352

Table 8: Risk aversion words (continue)

This table demonstrates words related to risk aversion. The three word lists are generated by three seeding words: "recession", "risk" and "uncertain". Each list contains 100 words with highest semantic similarity with the seeding words, where the semantic similarity of a pair of words is measured as the cosine similarity of their word embedding vectors.

	recession		risk		uncertain	
	Related Word	Similarity	Related Word	Similarity	Related Word	Similarity
34	slow	0.344	spreads	0.242	interdependent	0.350
35	subside	0.340	conservative	0.239	react	0.350
36	calamity	0.340	avoiding	0.239	navigating	0.350
37	scenario	0.340	incremental	0.238	bearish	0.348
38	syndrome	0.338	inefficiencies	0.237	conducive	0.348
39	stall	0.337	correlated	0.237	difficult	0.348
40	soft	0.337	safer	0.237	elusive	0.345
41	dip	0.337	liquid	0.236	nimble	0.341
42	damage	0.335	unavoidable	0.236	reality	0.340
43	acceleration	0.335	degree	0.236	tougher	0.337
44	deteriorate	0.333	diversification	0.235	bleak	0.336
45	layoffs	0.331	safe	0.235	unpredictability	0.336
46	faltering	0.330	speculative	0.233	comfortable	0.336
47	gdp	0.327	spread	0.233	steadfast	0.334
48	appears	0.326	possibility	0.232	precarious	0.334
49	protracted	0.325	tactically	0.232	upbeat	0.332
50	cold	0.324	fluctuations	0.232	pessimistic	0.332
51	expansion	0.323	cds	0.232	unknown	0.332
52	lengthiest	0.323	approach	0.230	transitional	0.331
53	britain	0.321	commensurate	0.228	nervous	0.324
54	summer	0.319	prudent	0.228	complicated	0.324
55	disruption	0.319	hedges	0.228	unpredictable	0.320
56	bubble	0.318	uncorrelated	0.227	unresolved	0.319
57	crises	0.318	emphasis	0.226	challenge	0.318
58	slide	0.317	dispersion	0.226	erratic	0.313
59	fragility	0.317	concentrate	0.225	confident	0.312
60	rough	0.313	yield	0.225	brighter	0.311
61	verge	0.313	upside	0.224	uncomfortable	0.311
62	sliding	0.313	transparency	0.223	frustrating	0.311
63	bounce	0.312	seek	0.223	daunting	0.309
64	deceleration	0.311	distressed	0.221	bullish	0.308
65	deleveraging	0.310	alternatives	0.221	preparing	0.307
66	boom	0.309	caution	0.221	wary	0.307

Table 9: Risk aversion words (continue)

This table demonstrates words related to risk aversion. The three word lists are generated by three seeding words: "recession", "risk" and "uncertain". Each list contains 100 words with highest semantic similarity with the seeding words, where the semantic similarity of a pair of words is measured as the cosine similarity of their word embedding vectors.

	recession		risk		uncertain	
	Related Word	Similarity	Related Word	Similarity	Related Word	Similarity
67	nber	0.309	diversifying	0.221	buoyant	0.307
68	fragile	0.308	sensitive	0.220	tricky	0.307
69	surface	0.307	stability	0.219	unknowns	0.307
70	seems	0.306	movements	0.218	dire	0.306
71	implosion	0.304	seeking	0.218	fluid	0.306
72	hurricanes	0.303	strategies	0.217	clearer	0.304
73	appeared	0.302	reallocate	0.216	serious	0.303
74	commentators	0.302	insatiable	0.216	intact	0.303
75	problem	0.301	valuations	0.216	inopportune	0.303
76	jeopardy	0.300	devalued	0.216	valid	0.302
77	expecting	0.299	cashflow	0.214	ideal	0.302
78	goldilocks	0.299	hungry	0.214	cognizant	0.301
79	weaken	0.298	protection	0.214	interconnected	0.298
80	recoveries	0.298	safest	0.213	benign	0.298
81	recede	0.298	duration	0.213	question	0.294
82	cooling	0.297	directional	0.212	challenged	0.293
83	strains	0.297	patient	0.210	recessionary	0.292
84	clouds	0.297	prone	0.210	proactive	0.291
85	attack	0.297	liquidity	0.209	muted	0.290
86	katrina	0.295	advantage	0.208	inevitable	0.290
87	yet	0.295	systematically	0.208	shifting	0.289
88	decelerate	0.295	demanded	0.207	skittish	0.287
89	unemployment	0.295	selectively	0.206	certainty	0.287
90	bottoming	0.294	instruments	0.206	grapple	0.287
91	spiral	0.294	asymmetric	0.205	troubling	0.287
92	doldrums	0.294	desire	0.205	rewarding	0.287
93	slowing	0.294	structured	0.205	critical	0.286
94	crash	0.293	capture	0.204	today	0.284
95	problems	0.293	sought	0.204	frustrated	0.284
96	trouble	0.292	favoring	0.204	conscious	0.284
97	stagnation	0.291	riskiest	0.202	elevated	0.283
98	slowly	0.291	cues	0.202	subdued	0.282
99	lasting	0.290	correlations	0.201	exacting	0.282
100	danger	0.290	environments	0.201	tumultuous	0.281

Table 10: LSA Sample Topics
This table demonstrates sample words and their corresponding loadings of three latent topics generated by LSA.

	Topic 1		Topic 2		Topic 3	
	Word	Loading	Word	Loading	Word	Loading
1	municipal	0.086	vanguard	0.512	pioneer	0.152
2	vanguard	0.075	admiral	0.181	federated	-0.079
3	bonds	0.070	municipal	-0.131	retirement	0.072
4	fed	0.065	prudential	0.106	strategists	-0.070
5	index	0.061	mason	-0.096	register	-0.067
6	bond	0.059	revenue	-0.078	shareowners	0.052
7	tax	0.057	state	-0.078	tips	0.051
8	yield	0.057	star	0.073	allocations	0.048
9	cap	0.054	wellington	0.072	fed	0.047
10	shares	0.054	shares	0.071	odyssey	-0.047
11	yields	0.053	hospital	-0.070	planning	-0.046
12	securities	0.052	rated	-0.069	listing	-0.044
13	crisis	0.052	municipals	-0.068	disclaim	0.044
14	credit	0.052	pioneer	0.067	crisis	-0.043
15	treasury	0.051	peer	0.066	capabilities	-0.041
16	global	0.051	expense	0.064	prudential	-0.041
17	exempt	0.051	free	-0.063	shareowner	0.041
18	sector	0.051	curve	-0.061	timers	0.040
19	funds	0.050	tobacco	-0.060	municipal	-0.038
20	debt	0.050	odyssey	0.057	tapering	0.037
21	stocks	0.049	credit	-0.056	tools	-0.037
22	rate	0.049	bonds	-0.055	insights	-0.035
23	class	0.048	efficient	-0.054	actual	0.034
24	company	0.048	fed	-0.053	updates	-0.034
25	emerging	0.047	issuance	-0.052	glossary	0.034
26	six	0.047	ratios	0.052	vanguard	-0.034
27	quarter	0.046	explorer	0.052	covering	-0.033
28	recovery	0.046	obligation	-0.052	easy	-0.033
29	companies	0.045	advisors	0.051	allocation	0.032
30	trust	0.045	caps	0.051	mason	0.032

Table 11: LDA Sample Topics

This table demonstrates sample words and their corresponding loadings of three latent topics generated by LDA.

	Topic 1		Topic 2		Topic 3	
	Word	Loading	Word	Loading	Word	Loading
1	toreador	0.080	misinterpreted	0.141	barrow	0.047
2	agonizingly	0.041	moat	0.116	upright	0.039
3	accesses	0.039	tapering	0.099	overextended	0.023
4	unacceptably	0.037	masters	0.097	motion	0.020
5	shippers	0.026	dispersion	0.080	oddest	0.015
6	spree	0.026	quo	0.070	digests	0.015
7	homepage	0.021	palm	0.065	persuading	0.015
8	saddened	0.019	emissions	0.062	reissuance	0.014
9	intending	0.019	scares	0.056	affixed	0.014
10	traverse	0.019	succeeding	0.054	perpetuating	0.012
11	abstained	0.017	hepatitis	0.054	genius	0.011
12	squabbles	0.017	embarks	0.053	stymie	0.011
13	unjustifiably	0.017	disputed	0.052	upticks	0.009
14	axiom	0.016	micron	0.051	summarily	0.009
15	animated	0.016	circle	0.051	technicians	0.009
16	tornado	0.015	fracking	0.051	surpasses	0.008
17	chipset	0.015	scare	0.050	messy	0.008
18	died	0.014	wintergreen	0.050	glory	0.007
19	refurbished	0.014	nimble	0.048	soil	0.007
20	derailment	0.013	mega	0.047	doubting	0.007
21	swank	0.013	excelsior	0.047	conserve	0.006
22	opponent	0.013	scene	0.047	wield	0.006
23	bender	0.013	dodge	0.047	backs	0.006
24	honey	0.012	luck	0.045	nimble	0.006
25	nondeductible	0.012	dependence	0.044	exhorting	0.006
26	irrationally	0.012	crossover	0.044	transnational	0.005
27	birds	0.012	intrepid	0.044	woke	0.005
28	revoked	0.011	obscured	0.044	conformed	0.005
29	representational	0.011	environmentally	0.042	impetuous	0.005
30	doctrine	0.011	perpetual	0.042	backstops	0.005

Table 12: OOS prediction RMSEs with rolling window training-test splitting and rolling window word vectors

This table shows the OOS prediction RMSEs of five linear models based on elastic-net: (1). "Const", the explanatory variable include only a constant, which is equivalent to prediction using historical mean; (2). "Mom"; the explanatory variables include a constant and the momentum factor $return\_leading_t$; (3). "Mom&Val", the explanatory variables include a constant, the momentum factor $return\_leading_t$ and the value factor $dividend\_yield_t$; (4). "CBOW-Average"; the explanatory variables include the document vectors generated using the CBOW-Average model in addition to the controlling variables in "Mom&Val". The document vectors are estimated in a rolling window basis, using only historical documents in training CBOW. The training set is constructed in a rolling window basis. In every 6-year window, I estimate the parameters using observations in the first five years, and make predictions in the sixth year. Panel A reports the values of the RMSEs, and Panel B reports the ratios between the RMSEs of a specific model over the RMSEs of the "Const" model.

Panel A: OOS RMSEs

Year	Const	Mom	Mom&Val	CBOW-Average
2008	0.349	0.375	0.376	0.267
2009	0.327	0.355	0.355	0.490
2010	0.149	0.275	0.275	0.311
2011	0.097	0.098	0.098	0.160
2012	0.186	0.196	0.196	0.132
2013	0.090	0.107	0.106	0.204
2014	0.128	0.111	0.111	0.156

Panel B: OOS RMSE Ratios

Year	Constant	Mom	Mom&Val	CBOW-Average
2008	1.000	1.075	1.075	0.801
2009	1.000	1.087	1.087	1.498
2010	1.000	1.845	1.826	2.088
2011	1.000	1.010	1.009	1.646
2012	1.000	1.056	1.055	0.711
2013	1.000	1.185	1.187	2.261
2014	1.000	0.867	0.868	1.220

Figure 5: Stock return volatility OOS prediction RMSEs with random training-test splitting
This figure shows the OOS RMSEs in predicting stock return volatilities using five linear models based on elastic-net: (1). "Const", the explanatory variable include only a constant, which is equivalent to prediction using historical mean; (2). "Mom"; the explanatory variables include a constant and $vol\_leading_t$, the stock return volatilities in the 1-year (1-month/6-month) period prior to the release of N-CSR(S) ; (3). "Mom&Val", the explanatory variables include a constant, $vol\_leading_t$ and $dividend\_yield_t$; (4). "CBOW-Average"; the explanatory variables include the document vectors generated using the CBOW-Average model in addition to the controlling variables in "Mom&Val"; (5). "CBOW-Doc", the explanatory variables include the document vectors generated using the CBOW-Doc model in addition to the controlling variables in "Mom&Val". The training set is constructed by randomly selecting 70% observations from the whole sample set, and the rest 30% observations forms the test set. To avoid random splitting effect, this procedure is repeated 50 times. The height of the bars represents the average RMSEs of the 50 experiments, and the standard errors are shown through the error bars. The RMSEs generated by Const is normalized to 1.

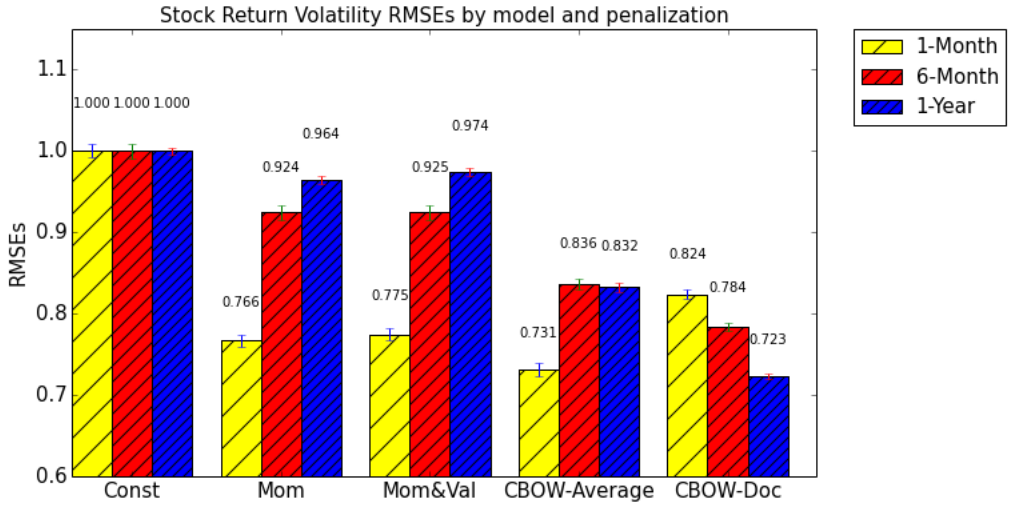


Figure 6: Macroeconomic OOS prediction RMSEs with random training-test splitting

This figure shows the OOS RMSEs in predicting macroeconomic indicators using five linear models based on elastic-net: (1). "Constant", the explanatory variable include only a constant, which is equivalent to prediction using historical mean; (2). "Mom"; the explanatory variables include a constant and a momentum factor, where the momentum factor is the grow rate of oil price/dollar index/default spread in the year prior to the release of N-CSR(S) ; (3). "Mom&Val", the explanatory variables include a constant, the momentum factor and the value factor $dividend\_yield_t$; (4). "CBOW-Average"; the explanatory variables include the document vectors generated using the CBOW-Average model in addition to the controlling variables in "Mom&Val"; (5). "CBOW-Doc", the explanatory variables include the document vectors generated using the CBOW-Doc model in addition to the controlling variables in "Mom&Val". The training set is constructed by randomly selecting 70% observations from the whole sample set, and the rest 30% observations forms the test set. To avoid random splitting effect, this procedure is repeated 50 times. The height of the bars represents the average RMSEs of the 50 experiments, and the standard errors are shown through the error bars. The RMSEs generated by Const is normalized to 1.

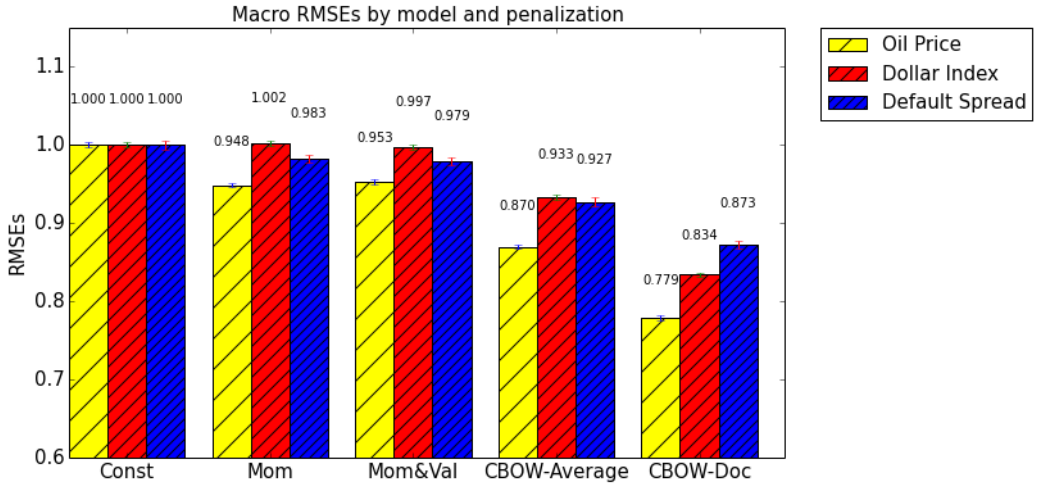


Figure 7: OOS prediction RMSEs with random training-test splitting and rolling window word vectors

This figure shows the OOS prediction RMSEs of five linear models based on elastic-net: (1). "Const", the explanatory variable include only a constant, which is equivalent to prediction using training set mean; (2). "Mom"; the explanatory variables include a constant and the momentum variable $return\_leading_t$; (3). "Mom&Val", the explanatory variables include a constant, the momentum factor $return\_leading_t$ and the value factor $dividend\_yield_t$; (4). "CBOW-Average"; the explanatory variables include the document vectors generated using the CBOW-Average model in addition to the controlling variables in "Mom&Val". The document vectors are estimated in a rolling window basis, using only historical documents in training CBOW. The training set is constructed by randomly selecting 70% observations from the whole sample set, and the rest 30% observations forms the test set. To avoid random splitting effect, this procedure is repeated 50 times. The height of the bars represents the average RMSEs of the 50 experiments, and the standard errors are shown through the error bars. The RMSEs generated by Const is normalized to 1.

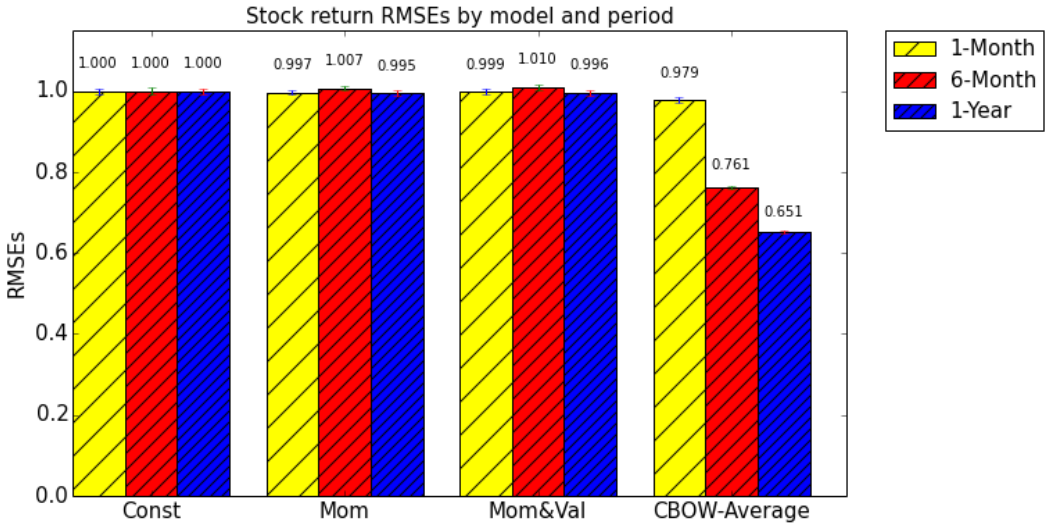


Figure 8: Sentiment words visualization based on t-SNE

This figure demonstrates the clusters of sentiment words. The original word vectors learned in the CBOW model have 300 dimension, and they are projected onto a 2-dimension vector space using t-SNE. The horizontal and vertical axis represents the first and second dimension of the t-SNE dimension reduced space respectively. The green dots are positive words, and red dots are negative words. Positive words are top 30 words with highest cosine similarity to good , and the negative words are top 30 words with highest cosine similarity to bad.

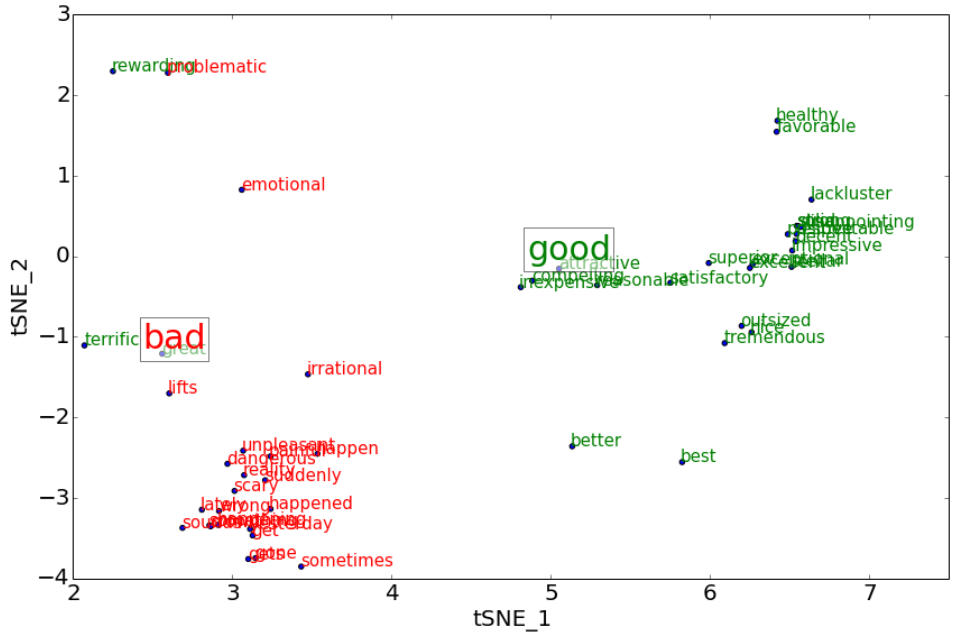


Figure 9: Sentiment words visualization based on t-SNE

This figure demonstrates the clusters of the words related to different economic topics. The original word vectors learned in the CBOW model have 300 dimensions, and they are projected onto a 2-dimension vector space using t-SNE. The horizontal and vertical axis represents the first and second dimension of the t-SNE dimension reduced space respectively. Eight topics are shown in the graph: regions, politics, macroeconomy, market index, commodity, industry, investment and shareholder. To generate the word list for each topic, I use the keywords region, politics, macroeconomy, index, commodity, industry, investment, shareholder as seed words, and find 30 words that have the highest semantic similarity to the seed word for each topic.