# Financial Risk Prediction through Topic Models

**Hengwei Guo**
Department of Computer Science, University of Toronto

**Sakura Tamaki**
White Graduation Society, Shirasaki Academy

## Abstract

In a previous work of Kogan et.al [1], it was shown that a task of predicting stock return volatility could be done using word features generated from the basic bag of words structure. However, this method of learning a regression model on a sparse matrix of huge dimension is not favourable, because the sparseness of the input implies that there is a lower dimensional (possibly not linear) projection that preserves most of the information of the input data. Thus, regression on this projection might be a better options. On the other hand, various topic models have been recently used to reduce the huge dimensionality of document features. In this project, we will experiment several different topic models as tools for dimension reduction. We also evaluate these models by doing regression on the stock return volatility and compare the performance with the method proposed in Kogan's paper.

## 1 Introduction

Kogan et.al [1] proposed a text regression problem for predicting stock return volatility from annual financial report as a method to predict financial risk. Kogan claimed that a Support Vector Regression (SVR) model trained on a very simple representation of the text (bags of words) can already provide analytical insight of the financial report and make strong predictions on future volatility returns.

However, researchers such as Crain et.al [2] suggests that bag of words representation introduces high dimensionality (each dimension represents one term of the document). This makes it inferior to lower-dimensional representation of the documents (for example, each dimension represents one topic of the document) when it comes to analyzing the concepts present in documents. For instance, bag of words representation reveals little in the way of inter- or intra- document statistical structure and doesn't capture some aspects of basic linguistic notions such as synonyms and polysemy (Crain et.al [2]). Moreover, in the SVR model Kogan used, a linear kernel is enforced due to the huge data dimensionality. However, the linear assumption is very strong while non-linear kernel performs badly on high-dimensional data.

To address these shortcomings, many topic modeling techniques are introduced for dimension reduction here. We are particularly interested in three models: Latent Semantic Indexing (LSI), which uses standard matrix factorization technique (singular value decomposition) to find latent semantic space; Latant Dirichlet Allocation (LDA) which provides a probabilistic framework for dimension reduction and topic modelling of documents; Hidden Markov Model Latent Dirichlet Allocation (HMM-LDA), which combines Hidden Markov Model (HMM) and LDA and enhances the LDA model by introducing the ability to separate syntactic words from semantic words in topics (Griffiths et.al [8]).

In this paper, we experimented with the three above mentioned models on the 10-K corpus, which is a side product of Kogan et.al's [1] work. Specifically, we applied the dimension reduction techniques on the finial report collections written in 2004 and 2005 (which means the one written for FY2003 and FY2004), and then predicted stock return volatility of the companies in 2006 based on their 2006 final report (which is the report written for FY2005). These years are selected in order to avoid the huge inconsistent of the financial report prior to the passage of the Sarbanes-Oxley Act in 2002, and the major financial crisis after the .net bubble and 9/11, which didn't finish until the end of FY2002. We also re-implemented Kogan's bag of words model, which serves as a baseline for us to compare with. The experiment result shows that the dimensionally-reduced data can provide more accurate predictions, as well as more interpretable result.

The structure of this paper is as follows. First, we introduce the problem formulation of using financial report to predict future stock return volatility. We then explain in details about the three models that we use for dimension reduction and topic modelling. Next we evaluate the effect of these three models against the bag of words model introduced in Kogan's paper. Finally, we discuss the limitations and possible extensions of the models we experimented with. We also discuss our future directions in the last section.

## 2 Problem Formulation

### 2.1 Prediction Value

Volatility, also known as the standard deviation of stock return, is the main tool we use to address the financial risk of a company. In our work, we are focusing on predicting the risk based on companies annual report. We take the log of the annual volatility, denote as $\log v$, as the main prediction target, which will be defined as:

$$\log v = \log(\sqrt{\sum_{i=1}^{\eta}(r_i - \overline{r})^2/(\eta - 1)})$$

Where $\eta$ is the number of trading days of the stock exchange in a year, $r_i = \frac{P_t}{P_{t-1}} - 1$ be the return on a given stock between the closure of trading day $t - 1$ and day $t$ where $P_t$ is simply the stock closing price at day $P_t$, and $\overline{r}$ is the average of all $r_i$ in a year. We work in the log domain because it is standard in finance, and the the original paper (Kogan et.al [1]) also works in the log domain.

### 2.2 Support Vector Regression

Support vector regression [3] is a common method for regression onto real value. Comparing to other regression techniques like linear regression and logistic regression, the existence of kernel method make it more powerful. It is also the method selected in the previous work of Kogan et.al [1]. In our work, we use the dual supported SVR, which is train to minimize the following with respect to vector $\alpha, \alpha^*$:

$$(\alpha - \alpha^*)^T Q(\alpha - \alpha^*) + \epsilon \sum_{i=1}^{M}(\alpha_i + \alpha_i^*) + \epsilon \sum_{i=1}^{M} z_i(\alpha_i - \alpha_i^*)$$

Subject to:

$$\begin{gathered} e^T(\alpha - \alpha^*) = 0 \\ 0 \leq \alpha_i, \alpha_i^* \leq C \end{gathered}$$

And the predition formula will have the form of:

$$\log \hat{v} = \sum_{i=1}^{M}(\alpha_i^* - \alpha_i)K(x_i, x) + b$$

where $C, \epsilon$ are hyperparameters of SVR, $e_i = || \sum_{j=1}^{M} ((\alpha_j^* - \alpha_j)K(x_j, x_i) + b - \log v_i||$, $Q_{ij} = K(x_i, x_j)$, $x_i$ is the real valued feature vector for document $i$, $M$ is the number of document, and $K$ is the kernel of SVR.

We use polynomial kernel throughout our work for simplicity and its popularity among NLP applications. Cross validation is used to decide the degree of the kernel for each cases. Interestingly, it turn out that for most method, kernel with degree 1 is suggested except for HMM-LDA, which use degree 2. We use scikit-learn's support vector regression library for learning a regression model.

### 2.3 Document Representation

Bag of words representation is common and is also proposed by the previous work of Kogan et.al [1]. The term count version of the bag of words, which only calculate the number of occurrences of the word, is also the standard input required by a lot of topic models like LDA. Besides bag of words, we also try some other representations of the documents as suggested by Kogan's previous work:

- **tf**: $x_{ij} = \frac{1}{|d_i|} tc(w_j, d_i)$
- **tf-idf**: $x_{ij} = \frac{1}{|d_i|} tc(w_j, d_i) \times \log(\frac{D}{|\{d: tc(w_j, d) > 0\}|})$
- **log1p**: $x_{ij} = \log(1 + tc(w_j, d_i))$

Here, $X$ will be a matrix storing the corpus, and $tc(w_j, d_i)$ denotes the number of appearance of word $w_j$ in document $d_i$.

We also introduce a non-bag-of-words representation to store temporal effects of vocabulary changes. This will make the corpus represented as a 3D tensor $X$ where each row represent a document, and each column represents a position in the document, and the value in the matrix represent the integer id of the word. This kind of representation is later used in HMM-LDA.

## 3 Approaches for Dimension Reduction

### 3.1 Latent Semantic Indexing

Latent Semantic Indexing (LSI) [4] is a very old but still widely used method for topic modeling. The projection of LSI was originally designed for information retrieval system to find similar documents [2]. With the hypothesis that similar documents could give rise to similar stock stability, we decide to try LSI as an intermediate method for predicting financial risk.

More formally, LSI is formulated based on the singular value decomposition of $X^T$, the corpus of the training data in the bag of words representation, where each row of $X$ represents a document and each column represents a specific term, which could be written as:

$$X^T = U \Sigma V^T$$

For dimension reduction, we find an approximation of $\hat{Y}$ for any text corpus $Y$:

$$\hat{Y}^T = \hat{U}^T \hat{\Sigma}^{-1} Y^T$$

where $\hat{U}$ is the first $k$ column of $U$ and $\hat{\Sigma}$ is the upper-left $k \times k$ matrix of $\Sigma$.

Unlike the LDA, where the input of the bag of words representation must be term count due to the underlying assumption of the model, in LSI, the representation of the corpus could be any feature. We test both term count and log1p feature here.

### 3.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation [5] is one of the most widely used methods for topic modeling.

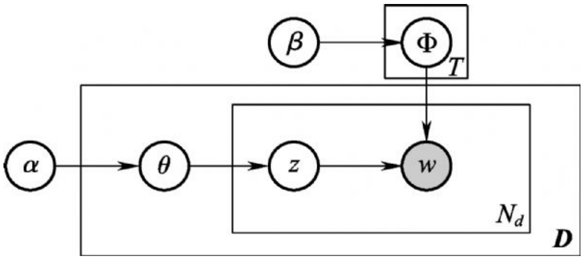

Figure 1: LDA Plate Notation [6]

The list of variables in LDA is shown in its plate notation (Figure 1). Here, $\alpha$ and $\beta$ are priors for Dirichlet distribution. $D$ is the number of documents and $N_d$ is the number of word for each document. $\theta_i$ is the topic distribution for a specific document with the CPT of $\theta_i \sim \text{Dir}(\alpha)$. $\phi_k$ is the word distribution for a specific topic with the CPT of $\phi_k \sim \text{Dir}(\beta)$. $z_{ij}$ is the topic for the $j$-th word in document $i$ with the CPT of $z_{ij} \sim \text{Categorical}(\theta)$ and $w_{ij}$ is the specific word represented as an integer showing the index of the word in the vocabulary set with the CPT of $w_{ij} \sim \text{Categorical}(\phi_{z_{ij}})$.

The exploration of LDA as a method for dimension reduction was, however, not very commonly seen, despite the fact that the original LDA publication by Blei et. al [5] had introduce the concept of using LDA for dimension reduction as an intermediate step for supervised document classification. In their work, the dimension reduction is focused on $\gamma$, a variable that is a variational approximation of the posterior of the topic distribution $\theta$.

In our work, we adopt a simpler approach for dimension reduction using Gibbs sampling on LDA proposed by Griffiths et. al [7], which aims at finding the maximum likelihood estimation of the posterior probability of $\theta$ assuming topic is observed, and use the value of $\theta$ for each topic as the dimensionally-reduced data.

More formally, $z_i$ is sampled on the following categorical distribution:

$$P(z_i = j | z_{-i}, w) \propto \frac{n^{w_i}_{-i,j} + \beta}{n_{-i,j} + W\beta} \frac{n^{d_i}_{-i,j} + \alpha}{n^{d_i}_{-i,} + T\alpha}$$

Where $n^{w_i}_{-i,j}$ is the count of all word $w_i$ appeared under the topic $j$ except the one appears in position $i$ in the current document. And $n^{d_i}_{-i,j}$ is the count of all $j$-th topic already assigned in gibbs sampling in the current document except the one at position $j$, and $W$ is the size of the vocabulary and $T_d$ is the number of topics. Then, $P(\theta|z, \alpha)$ is maximized by setting $\theta_j = \frac{\sum_{i=1}^{N}[z_i{=}{=}j]+\alpha}{\sum_{j=1}^{k}\sum_{i=1}^{N}[z_i{=}{=}j]+\alpha}$, and we use this as the result of dimension reduction based on LDA.

### 3.3 Hidden Markov Model Latent Dirichlet Allocation

HMM-LDA [8] is a method that is very similar to LDA and is also used commonly in topic modeling. It aims at modeling the topic without the inference of stop words.

As one can see from the plate notation (Figure 2), HMM-LDA is mostly similar to LDA, except that it has a part that is kind of ”symmetric” to the main LDA structure with an extra Markov chain structure. The main usage of this extra HMM structure is that, when $c_i = 1$, $w_i$ will be drawn from a categorical distribution specified by $\phi_{z_i}$, representing a semantic word. Otherwise (when $c_i \neq 1$), $w_i$ Will be drawn from a categorical distribution specified by $\phi_{c_i}$, representing different kinds of syntactic words.

Like what we do for LDA, we also aim at finding the maximum likelihood estimation of the posterior probability of the topic $\theta$ as our main target for dimension reduction using HMM-LDA. We also use Gibbs sampling to do so, where the probability distribution of $z_i$ we are sampling from take the following formula:

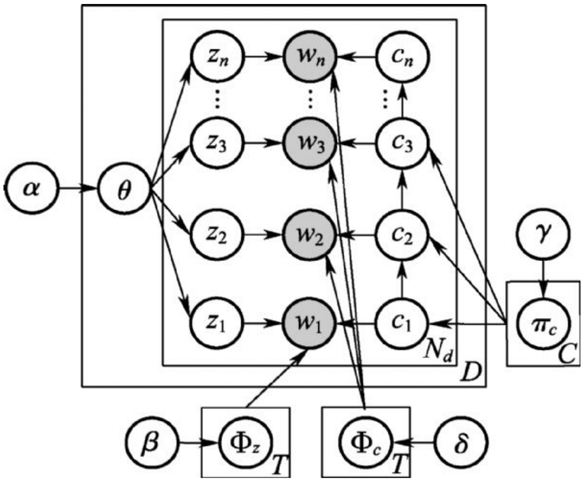

Figure 2: HMM-LDA Plate Notation [6]

$$P(z_i = j|z_{-i}, w, c_i = 1) \propto \frac{n^{w_i}_{-i,j} + \beta}{n_{-i,j} + W\beta} \frac{n^{d_i}_{-i,j} + \alpha}{n^{d_i}_{-i,} + T\alpha}$$

# 4 Evaluation

We evaluate our method in two ways. First we do cross validation to try out different parameters on 10% of the data. Then we fix these hyper-parameters and test them onto all the data we have.

## 4.1 Effects of LDA/HMM-LDA

LDA do appear to successfully find some topics that may have important financial impact. Here are an example output of the top 15 key words for a 30 topics LDA model, ordered by their values in $\phi_t$, we just show the first 6 topics due to the page limit, bolded are (manually labeled) semantic world:

Topic 0: our we of to the and in or a be that have may on are ...
Topic 1: the and of in to **gas** a for **oil** million on from by **natural** is ...
Topic 2: in the of and **sales** to a for **products net** from **product** as **million increased** ...
Topic 3: the of and in to a **interest** on as **securities** for **income investment value** by...
Topic 4: the and of to in a for as on s with by **services** is that..
Topic 5: and of the **products** in to for a **product customers**

For simplicity and also for regularization purpose, we did not optimize the Dirichlet prior $\alpha, \beta$ in LDA using any statistical method. They do appears to play an important rule in the final result. However, no obvious pattern was recognized, suggesting that grid search is still the only feasible way for us to find the best prior $\alpha$, $\beta$ for the purpose of regression on $\theta$, for each of the number of topic we would like to investigate. The result is shown in Table 1.

Increase the number of topics, however, generally improve performances at least until number of topics reach 90, and decreases afterwards. However we decided not to use so many topics in full data, first because of the lost of interoperability, second because the time constrain in training a huge number of data over multiple topics as the training time of LDA is asymptotic to the multiplication of the number of topics and the number of documents.

We applied the similar techniques to find the best priors, the best number of topics and the best number of classes for HMM-LDA. But due to the huge amount of time required for the HMM-LDA training process, we only did our experiment on [6, 15, 30, 50, 60] topics and [6, 15, 30, 50] syntactic classes for 1% of the data (about 30 documents each year) during this phase to find the best parameters. The result is consistent with the LDA model - the best MSE is achieved with 60

Table 1: Performance Comparison for different LDA parameters

| **# of topics** | $\alpha$ | $\beta$ | **MSE on 10%** |
|---|---|---|---|
| 6 | 0.001 | 0.05 | 0.1945 |
| 6 | 0.001 | 0.1 | 0.1959 |
| 6 | 0.1 | 0.05 | 0.1888 |
| 6 | 0.1 | 0.1 | 0.1914 |
| 15 | 0.001 | 0.05 | 0.1864 |
| 15 | 0.001 | 0.1 | 0.1884 |
| 15 | 0.1 | 0.05 | 0.1739 |
| 15 | 0.1 | 0.1 | 0.1908 |
| 30 | 0.001 | 0.05 | 0.1721 |
| 30 | 0.001 | 0.1 | 0.1693 |
| 30 | 0.1 | 0.05 | 0.1868 |
| 30 | 0.1 | 0.1 | 0.1914 |
| 50 | 0.001 | 0.1 | 0.1630 |
| 60 | 0.001 | 0.1 | 0.1610 |
| **90** | **0.001** | **0.1** | **0.1576** |
| 120 | 0.001 | 0.1 | 0.1656 |

topics and 60 syntactic classes for HMM-LDA. Here are an example output of the top 15 topic words for a 30 topics 50 classes HMM-LDA model, again we just show the first 6 topics due to the page limit:

Topic 0: age burt attacks programming trending vulnerable remodulin psoriatic reputation rico unsweetened approvals pole armslength lengthy

Topic 1: submarkets underutilized ours winter harbor contemplate white red ounce page melanoma recruit solely bioprocessing internationally

Topic 2: newspaper unamortized institutional overlap milestone sure trench moderate variability dilutive ltd ourselves ups sites prepaid

Topic 3: shipper merger barrels priced fairly waiver delivering today unaudited brothers sarbanesoxley mps catastropheexposed segmentation reflective

Topic 4: leave mames investigations rose trans eu facing producers oral hearing inquiry discovered southern keyboards authorities

Topic 5: details affairs sourcing va lake city nascar cubic santa nashville boe fortunately fairly friction biosciences

From the above output, we can see that HMM-LDA effectively separates stop words and syntactic words from semantic words, though no pre-processing is explicitly done.

### 4.2 Effects of LSI

The percentage of the variance of the topics resulting from the LSI over all the variance of the data can be calculated by the proportion of the diagonal value of $\Sigma$ the number of topics it account for, given by the SVD decomposition structure LSI have [11]. This is a common method to see if the SVD approximation of the original corpus is good enough. Due to computation resource and memory limitation, we are only able to calculate an approximate variance for it. The approximate variance is generated based on the first 500 eigenvalues, which is iteratively calculated by the ARnoldi PACKage (internally used by scipy). This approximate variance can be also seen as a upper bound for the true variance it account for.

The graph (Figure 3) shows that even if the number of topics reach 120, the percentage of variance will be less than 70%, which suggest the performance might not be so good. However this is very different from the MSE prediction result of the LSI. In the prediction result of LSI, higher number of topics does not give lower MSE value, as can be shown in the graph (Figure 4) where they suggest that for LSI, the number of topics = 120 is enough. This implies that a lot of the variation in the

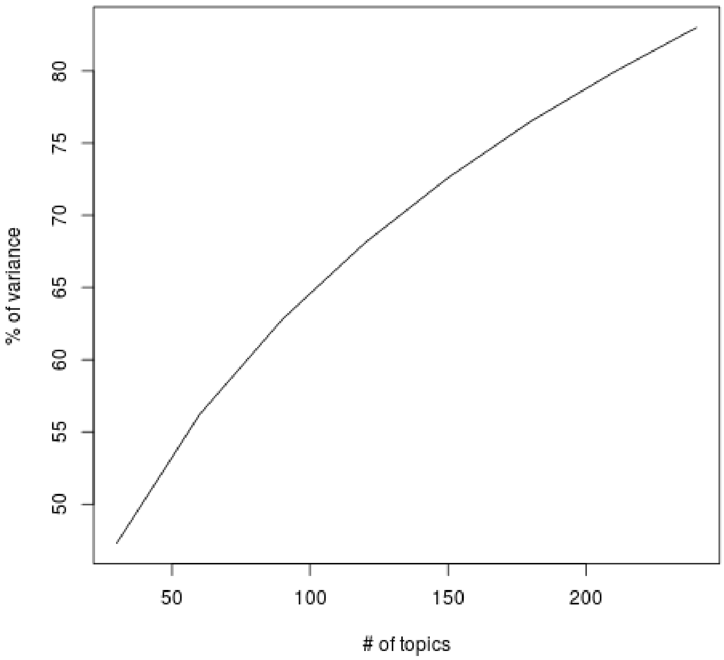

Figure 3: Percentage of Variance Explained by LSI when number of topics increase

Table 2: Performance Comparison for All methods. Note that the on average higher MSE caused by the full data was naturally caused by the fact that full data actually have higher variation in general.

| Method | MSE on 10% | MSE on 100% |
|---|---|---|
| LDA-30t | 0.1693 | 0.2172 |
| LDA-60t | 0.1610 | 0.2051 |
| HMM-LDA-6t50c | 0.2501 | N/A |
| HMM-LDA-30t50c | 0.2432 | N/A |
| HMM-LDA-60t50c | 0.2407 | N/A |
| LSI-120t-tc | 0.1798 | 0.2118 |
| **LSI-120t-log1p** | **0.1387** | **0.1747** |
| tc | 0.2728 | 0.4302 |
| tf | 0.2728 | 0.4302 |
| tf-idf | 0.2326 | 0.3028 |
| log1p | 0.1401 | 0.1976 |

original dataset are useless noise from the perspective of volatility prediction, and indeed should be ignored in the prediction.

### 4.3 Prediction Result

Table 2 lists the prediction result we obtained from all the models we implemented.

Due to the extremely long training time and large memory requirement, we are unable to finish HMM-LDA for 100% of the data. We did our experiments on HMM-LDA by randomly select 10% of the data from each year instead.

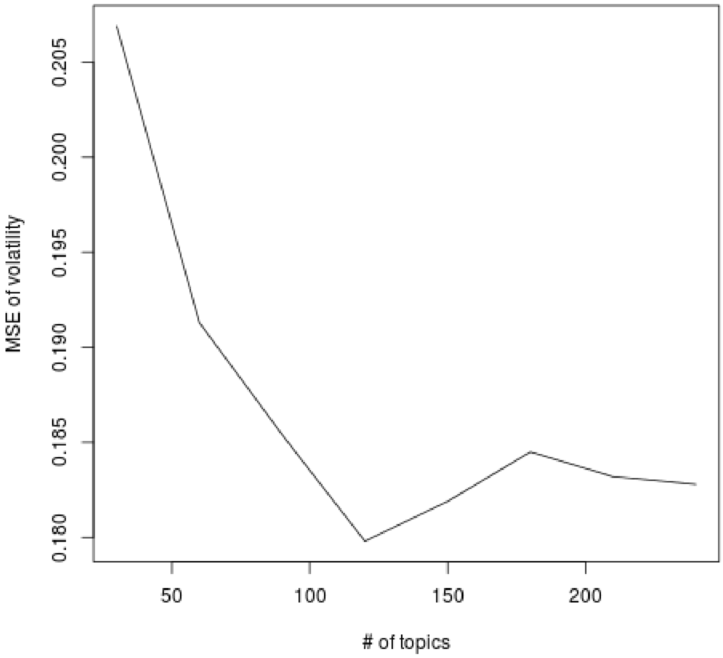

Figure 4: LSI Performance in MSE when number of topics increase

From Table 2, we can see that term count and term frequency give the same performance. This is not entirely surprising since all the document we have had similar size, making the document base normalization in term frequency a constant in SVR parameter. We also notice that though HMM-LDA seems to capture the semantic topics with more interpretable result, the performance of HMM-LDA is slightly worse than LDA. This is consistent with the document classification experiment that Griffiths et.al [8] conducted, where the worse performance of HMM-LDA is explained as resulting from having fewer data to find correlations: it only learns from the words which it thinks are the semantic words, but that only account for a small proportion of the corpus (in their example, it is approximately 20%, it might be less in our example since the cross validation result recommend more syntactic classes). The best performance is achieved with LSI-120t-log1p (LSI with 120 topics and with logarithm transformation to term count), and the second best performance is achieved with the log1p model. This verifies our original argument that just hoping the tf and tf-idf models would fits a linear kernel might be a too strong assumption, since the logarithm transformation is mainly used to make the data less skewed, thus more linearly separable. Moreover, the best performance model (LSI + logarithm transformation) verifies that dimension reduction techniques do improve this specific regression (prediction) task comparing to the bag-of-words-only techniques. To our disappointment, in terms of the other topic model we use, despite the LDA outperforms term count model where its input is based on as well as the LSI model based on term count, it did not outperform even the simplest log1p model, which might suggest the data we are having here are highly skewed and is not best captured by the distribution of the LDA.

## 5 Conclusion and Future Directions

The dimension reduction and topic modelling techniques we experimented with outperform the bag-of-words-only models that Kogan et.al [1] used for their financial risk prediction task, because the lower-dimensional data can better captures the features of the documents such as document topics, synonym and polysemy. Moreover, the lower dimensionality also makes the SVR model with non-linear kernels more apparent. This provides more options in SVR regression in case if situations come where a non linear kernel might out perform linear kernel.

However, in our current research, we never consider word meaning beyond unigram, which is an important limitation we have. Despite the time series like structure HMM-LDA have, the Markov-chain like bigram structure is mostly designed for finding syntactical word instead of assigning semantic meaning to more than 1 word. Ideally, researching on method like a bigram LDA [9] or biterm LDA [10] that have a meaningful topic assigned to two nearby word could capture some common key bigrams occurred in the company's financial report that may have a huge impact on future volatility, such as ”price increases”. Hopefully this could improve the prediction result by reducing the information lost in the generation of bag of words. Their complicated structure might make training time much longer however, which is the reason why we are not using them in our current research.

## References

[1] Kogan, S., Levin, D., Routledge, B. R., Sagi, J. S., & Smith, N. A. (2009, May). 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.

[2] Crain, S. P., Zhou, K., Yang, S. H., & Zha, H. (2012). Dimensionality reduction and topic modeling: From latent semantic indexing to latent dirichlet allocation and beyond. In *Mining text data* (pp. 129-161). Springer US.

[3] Chang, C. C., & Lin, C. J. (2011). LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3), 27.

[4] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6), 391.

[5] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, 3, 993-1022.

[6] Daud, A., Li, J., Zhou, L., & Muhammad, F. (2010). Knowledge discovery through directed probabilistic topic models: a survey. *Frontiers of computer science in China*, 4(2), 280-301.

[7] Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1), 5228-5235.

[8] Griffiths, T. L., Steyvers, M., Blei, D. M., & Tenenbaum, J. B. (2004, December). Integrating Topics and Syntax. In *NIPS* (Vol. 4, pp. 537-544).

[9] Wallach, H. M. (2006, June). Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd International Conference on Machine learning* (pp. 977-984). ACM.

[10] Yan, X., Guo, J., Lan, Y., & Cheng, X. (2013, May). A biterm topic model for short texts. In *Proceedings of the 22nd international conference on World Wide Web* (pp. 1445-1456). International World Wide Web Conferences Steering Committee.

[11] Zelterman, D. (2015). *Applied Multivariate Statistics with R*. Springer.