Multi-Representation Approach to Text Regression of Financial Risks

Roman Trusov
ITMO University
Saint-Petersburg, Russia
r.v.trusov@gmail.com

Alexey Natekin
Deloitte Analytics Institute
Moscow, Russia
anatekin@deloitte.ru

Pavel Kalaidin
VK
Saint-Petersburg, Russia
pavel.kalaidin@gmail.com

Sergey Ovcharenko
NTech Lab
Moscow, Russia
ovcharenko.ser@gmail.com

Alois Knoll
Technical University Munich
Garching, Germany
knoll@in.tum.de

Aida Fazylova
KL10CH
Moscow, Russia
ai@kl10.ch

***Abstract*—Different approaches for textual feature extraction have been proposed starting with simple word count features and continuing with deeper representations capturing distributional semantics. In recent publications word embedding methods have been successfully used as a representation basis for a large number of NLP tasks like text classification, part of speech tagging and many others. In this article we explore opportunities of using multiple text representations simultaneously within one regression task in order to exploit conventional bag of words approach with the more semantically rich embeddings. We investigate performance of this multi-representation approach on the financial risk prediction problem. Publicly available 10-K reports filled by US trading companies are used as the basis for predicting next year change in stock price volatility. Our study shows that models based on single representations achieve performance that is comparable to the previously published results on risk prediction and models with multiple representations benefit from complementary information and outperform both baseline and single representation models.**

I. Introduction

In order to build text regression and classification models, documents should be properly transformed into a meaningful representation, typically in the form of a feature vector of fixed length. The most commonly used text representation is the bag-of-words model [1], which maps each text document to a word-count vector, often supplemented with some word weighting schema.

These vectors can serve as the basis for methods like Latent Dirichlet Allocation [10] in topic modeling, which in their turn can be used as inputs to other machine learning tasks [24]. Linear models on top of bag-of-words proved to be a reliable tool [4], [5], used in different tasks like movie revenue and ratings predictions [30], [21]. Although such representation often turns out to be very high-dimensional with thousands of word-count features, regularization techniques [22] for generalized linear models [3] efficiently deal with this problem.

These simple yet efficient models have found many successful applications in Sociology, Psychology, Political Science and Linguistics [23], [19], [17]. Economics and Finance have also found many promising cases that leverage available textual information [2] for various regression and classification tasks:

- Fraud detection [28] in financial reports, based on decision trees, Locally Weighted Learning, Naive Bayes and SVM for classification.
- Sentiment analysis and popularity estimation [33], based on Artificial Neural Networks on top of n-gram scheme. Besides the obvious marketing application, the same framework can be reused for short-term financial forecasts.
- Determination of relationships between sentiments and trading volume, as well as other indicators like return volatility, fraud probability and unexpected earnings [27]. This research is based on 10-K annual financial reports with TF-IDF weighted bag-of-words features.

A notoriously interesting sub-domain of Finance is the field of Risk Management. Credit scoring[11] is the most well-known success story, yet there are other promising applications, in particular the ones based on stock price risk predictions [16].

Although financial forecasts are a challenging problem with much effort invested into it [34], financial volatility predictions are successfully built with reasonable accuracy. And most importantly, one can achieve accuracy gains by exploiting text data as well as financial features [16]. General idea behind this approach is that the language used in financial statements can be semantically mapped to risk awareness and therefore, used for volatility prediction [6], [7]. Moreover, it also allows to explore the relations between certain text terms and financial risk indicators.

One great advantage of these works is that they are based on publicly available data like the EDGAR database maintained by U.S. Securities and Exchange Commission (www.sec.gov/edgar.shtml). Specifically 10-K reports contain the most important disclosures about the company's performance, thus much research relies on them [28], [27], [16].

ISBN 978-952-68397-0-7 (paperback), ISBN 978-952-68397-1-4 (PDF)

Our research is inspired by the recent advances in learning rich feature representations by means of word embeddings [31]. Word embeddings are vectors whose relative similarities are learned to correlate with semantic similarity [20]. These vectors are used as a representational basis for Natural Language Processing tasks like text classification [25], part of speech tagging [15], sentiment analysis and others [35]. Word embeddings have become a viable alternative to the shallow bag-of-words representation due to excellent performance on a large number of applications [38], [15].

The main goal of this article is to explore opportunities of using word embedding representation for stock price volatility prediction. It is important to note that bag-of-words representation and word embeddings are not mutually exclusive and can be used simultaneously within the same learning task. Hence our key idea to further enhance predictive qualities lies in the basic feature and model fusion of these two representations. Our study show that each of these representations on it's own doesn't provide considerable accuracy improvement whereas their joint exploitation improves accuracy of next year volatility predictions from 5% to 12%.

The paper proceeds as follows. Section 2 provides the description of the data collection process. Section 3 specifies procedures of data processing target variable preparation. Section 4 describes text representation opportunities used in the course of our analysis. Section 5 outlines the machine learning framework used in experiments. Section 6 presents experimental results of the modeling and section 7 concludes this article with potential issues as well as opportunities of our approach.

II. Data collection

Financial risk prediction is a challenging problem because risks reflect not only company performance, but also industry dynamics, state of the market and even societal effects. Hence they can be affected by a tremendous amount of internal and external drivers with scarcely available data. Another problem is that many risk drivers are irregular and event-driven thus making conventional continuous modeling complicated.

To tackle these problems we follow the design of [16] and use ticker data about company stock returns as the principle reflection of market performance and form 10-K fillings as company's internal convoluted vision of potential market drivers. Form 10-K filing is an obligatory annual financial report for publicly traded companies in the United States. This report provides a comprehensive summary of a company's financial performance. The most important part of this report is Section 7A, "Quantitative and qualitative disclosures about market risk", part of the "Management's discussion and analysis" section that contains insights about company's own predictions and vision of the market.

Both 10-K fillings and daily stock prices are publicly available and are regularly updated thus providing consistent data for analysis.

To proceed to data collection we will consider a compiled list of companies that are listed on NASDAQ and NYSE. Our study is limited to reports from 2003 to September 2015, starting with the first collected report issued on January 13, 2003, and ending with reports from December 31, 2014. Although EDGAR database provides access to a significantly larger set of reports regarding observation history, we limited our time frame due to the following reasons:

- Dotcom era of 1997-2000 had a strong impact on stock markets thus bringing a large number of outliers and very specific market behavior that would affect predictive properties of text regression models.
- The **Sarbanes-Oxley Act** of 2002 [13] enhanced the standards for financial disclosure, which led to more elaborate reviews of company performances. Hence the period of 2001-2002 was excluded in order to reach a more homogeneous data quality.

Texts of 10-K reports were obtained via publicly available API provided by EDGAR system, maintained by U.S. Securities and Exchange Commission (SEC). A total of 28057 10-K filings from 3183 publicly traded companies were collected. For each 10-K report we collected a vector of adjusted closing stock prices across one year prior to report publishing. Stock prices were collected via Yahoo Finance public API.

III. Data processing

EDGAR database doesn't provide means for extracting specific parts of the fillings, hence the section extraction was automated. In order to increase the size of extracted textual sources of information we extracted the whole Section 7 despite the fact that Section 7A is considered more specific for this type of analysis. However based on the official definition, Section 7 can be used for "comparing the current period versus the prior period", which perfectly corresponds to our concept of risk prediction.

In order to have our data convenient and suitable for analysis, extracted Section 7 texts were processed according to the following steps:

- text was converted to lowercase and all non-whitespace tokens shorter than 3 symbols were removed.
- non-character symbols were removed, yet special cases of interest with hyphenation, such as "Sarbanes-Oxley" were left intact.
- texts with fewer than 1000 words were dropped from the dataset.

After processing the resulting dataset comprised of 25507 complete report and stock price vector bundles. Out of the 2550 lost entries, 2102 were lost due to ticker name collisions and insufficient stock price data while 448 short documents were omitted due to processing. A descriptive summary of the amount of reports by year is shown on Fig.1. One can see that the amount of report data is annually increasing with short drops in 2008 and 2012 due to the aftermaths of economical turnovers that led to bankruptcies.

A. Target variable

To finilize our dataset collection and processing we collapse adjusted stock price vectors to volatility estimates for

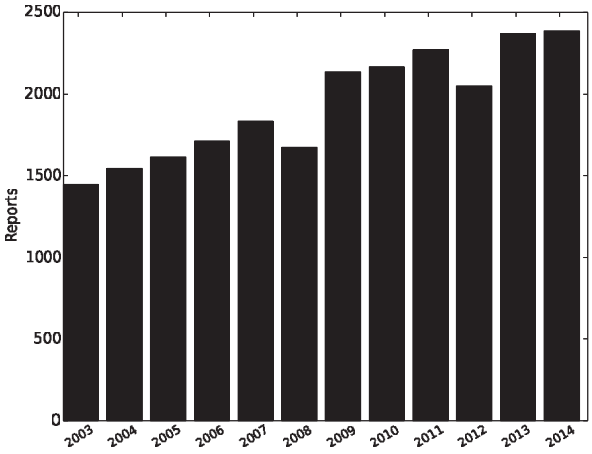

Fig. 1. Amount of reports by year

their corresponding annual periods. To be more specific, for each report from company i issued at year T we have a vector of daily adjusted for dividends stock prices $P_{i,t}$ across the prior year $\{P_{i,t}\}$, $t \in \overline{T-365, T}$. These prices were then converted to daily returns $r_{i,t} = \frac{P_{i,t}}{P_{i,t-1}} - 1$ and a standard deviation $\sigma_{i,T}$ of these returns was taken as the measure of risk:

$$\sigma_{i,T} = \sqrt{\frac{\sum_{t=T-365}^{T}(r_{i,t} - \overline{r_{i,T}}^2}{T_i}}$$

where T_i is the number of trading days of stock i at year T and $\overline{r_{i,T}}$ is the sample mean during this time period. Due to log-normal distribution of $\sigma_{i,T}$ which will be used as the target variable we followed the suggested [16] log-transformation thus leading to the final processed risk variable:

$$\hat{\sigma}_{i,T} = \log(\sigma_{i,T})$$

Stock price vectors were then substituted with $\hat{\sigma}_{i,T}$ estimates thus finalizing the text to target value correspondence. For each stock i we added the $\hat{\sigma}_{i,T+1}$ column to our dataset as the true target. The final dataset then had 25507 entries, consisting of a ticker name, publishing date, processed Section 7 text, prior volatility and the target volatility.

IV. Feature representations

A. Count-based models

Bag-of-Words (or Bag-of-n-Grams) model [1] represents each document with a sparse vector of the size equal to dictionary size, disregarding word order. **TF-IDF model** (term frequency-inverse document frequency) is similar to Bag-of-Words but with frequent words penalized.

A particularly successful application based on bag of words representation is topic modelling. Probabilistic topic model defines each topic by a multinomial distribution over words, and then represents each document with a multinomial distribution over topics.

An example of topic modelling is **Latent Dirichlet Allocation** [10]. The inferred topic probabilities provide an explicit compressed representation of a each document by a set of topics. Unfortunately these models require significant amount of documents for reasonable quality of the derived topics so in the case of financial risk prediction this model with it's features is not applicable. However given more data about company news one could make use it as an important information source.

In our scenario we find it hardly feasible to exploit all of the resulting terms in our TF-IDF feature matrix because we extracted nearly 200,000 different words. Therefore we will consider a truncated SVD projection of this large matrix. We will compare different options of this dimensionality reduction in our study, spanning a set of values from 25 to 500 reconstructed components.

B. Predictive models

Vector representations of words using neural networks got a lot of traction in machine learning community in the past years. In this formulation, each word is represented by a vector which is concatenated or averaged with other word vectors in a context, and the resulting vector is used to predict other words in the context.

Recently introduced neural network model of **Paragraph Vectors** [36] extends word models to go beyond word level to achieve document-level embeddings. It represents each document by a dense vector representations which are learned to predict the surrounding words in contexts sampled from the document. Paragraph Vectors are reported [37] to outperform bag-of-words and latent topic models on several text classification tasks.

C. Paragraph vectors

Paragraph Vectors framework comes in two fashions: preserving word order or discarding it. We use the former model, so called Distributed Memory Model of Paragraph Vectors. Every document is mapped to a unique vector and every word is also mapped to a unique vector. These vectors are used to predict the next word in a fixed-width context which is sampled from a sliding window over all documents. The document vector acts as a topic of the document which is missing from the current context and is learned in the process.

We follow the considerations and discussion from [36], [37] and define context length equal to 8 (maximum distance between the predicted word and context words used for prediction within a document).

Just like with SVD projections of our bag-of-words representation, we investigate performance of different amounts of extracted vector components, starting from 25 and ending with 500 to mirror our SVD experiment design.

D. Paragraph vectors

Visualization of the resulting embedding vectors by means of tSNE is presented on Fig.2. One can note that the resulting embedding preserves industry-specific context by grouping industries together with reasonable compactness. Also the learned distance is consistent due to having reports of the same company in the close proximity. To make the visualization more insightful we also added and calculated embedding for several reports from 2015, originally not included into our dataset.

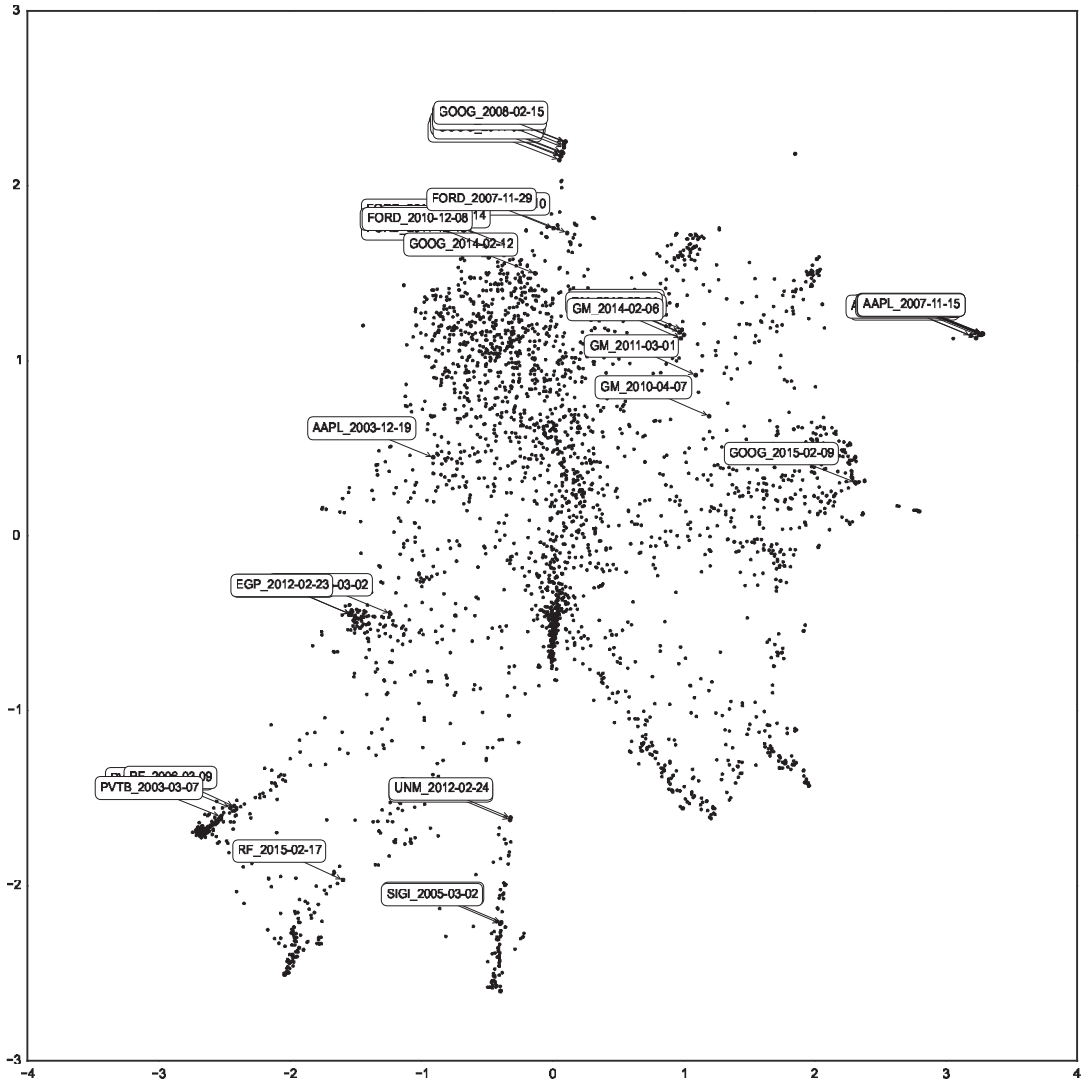

Fig. 2. tSNE embeddings of document vectors. Not only reports by the same company (as AAPL or GOOG), but reports from the same industry tend to form clusters (see bottom left clusters which are insurance companies (UNM, SIGI) and banks (RF, PVTB), and car manufacturers (GM, FORD) at the top

V. TEXT REGRESSION MODELS

Despite the fact that we deal with text data, we consider the regression problem in the following form. For a set of documents $\{D_i\}$ and associated target variables $\{\sigma_i\}$ we build a parametrized function $f(D, p) : d_i \rightarrow \sigma_i$ in the context of classic Least Square Estimate with squared L_2 loss that minimizes the Mean Squared Error:

$$\text{MSE}(\sigma_{true}, f(d_i, w)) = \frac{1}{M}\sum_{i=1}^{M}(\sigma_{true} - f(d_i, w))^2$$

where M denotes the number of document-value pairs.

A. Baseline model

Prior research [16], [6] states that conventional state-of-the-art in volatility prediction is based on linear autoregressive models of order 1. Hence we will use this model as our baseline. It is indeed a strong baseline to compete with for text regression models hence our goal here is to improve upon it.

B. Support Vector Regression

Support Vector Machines are a popular and successful approach in risk-prediction literature [6], [7]. Our research focuses on efficient exploitation of feature representations rather than trying out complex machine learning models and squeezing the resulting accuracy at all costs via fine tuning.

Following prior research in this domain we will focus on applications of vanilla linear-kernel Support Vector Regression (SVR) as the conventional successful tool. SVR is trained by solving the following optimization problem:

$$\min_{w\in\mathbb{R}^d}\frac{1}{2}\|w\|^2 + \frac{C}{N}\sum_{i=1}^{N}\max(0, |v_i - f(d_i, w)| - \epsilon)$$

Parameterized function $f(d_i, w)$ is defined as

$$f(d, w) = \sum_{i=1}^{N}\alpha_i K(d, d_i)$$

Where $K(d, d_i)$ is a kernel function (linear in our case). The cost hyper parameter C is chosen based on grid search via cross-validation.

C. Random Forest

Random forest is an ensemble learning method successfully applied for classification and regression tasks. Method builds a number of decision trees in the process of training and averages the prediction of all base decision tree estimators. This method has shown excellent results in numerous machine learning applications [9], [14] where a simple to handle strong non-linear model is required.

In our study we chose default parameters provided by scikit-learn python library [26] implementation of Random Forests with a total of 100 decision trees learned thus leaving us from any internal hyper parameter optimization via cross-validation procedures.

VI. EXPERIMENTAL STUDIES

A. Learning protocol

We carry our regression modeling experiments according to the following design:

- Every discussed feature representation option is built thus bringing text regression problem to it's conventional form with a feature matrix and target variable.
- Data is separated into global training and test sets. All entries from 2014 go into our test set which comprises of 2807 entries. The remaining 22700 entries form our training set.
- For each combination of representations and models considered a learning task with respect to 10-fold cross-validation on training set is performed.
- Two metrics are reported for each combination: MSE from out-of-fold cross-validation scores MSE_{CV} and test set MSE_{Test}.

B. Single representation models

During our first experiment for each representation option we investigate capabilities to capture the volatility dependence independently. We will encode bag-of-words representation with TF-IDF and SVD with n number of components as "TFIDF, svd, n". Paragraph embedding components will be coded similarly with "pv, n". Results of this experiment are presented in Table.I.

TABLE I. RAW FEATURES

Representation	SVR MSE_{CV}	RF MSE_{CV}	SVR MSE_{Test}	RF MSE_{Test}
Prev. value	0.204 ± 0.015	0.203 ± 0.017	0.205 ± 0.002	0.186 ± 0.001
TFIDF, svd 100	0.347 ± 0.057	0.312 ± 0.054	0.419 ± 0.002	0.360 ± 0.001
TFIDF, svd 250	0.309 + 0.055	0.306 + 0.055	0.364 + 0.002	0.357 + 0.003
TFIDF, svd 500	0.301 ± 0.054	0.307 ± 0.056	0.357 ± 0.002	0.356 ± 0.003
pv, 100	0.408 ± 0.029	0.387 ± 0.026	0.439 ± 0.001	0.452 ± 0.001
pv, 250	0.388 ± 0.028	0.398 ± 0.028	0.430 ± 0.001	0.467 ± 0.002
pv, 500	0.376 ± 0.026	0.405 ± 0.030	0.419 ± 0.001	0.479 ± 0.003

From these results we can denote several observations. First of all the baseline indeed outperforms other representations dramatically in terms of predictive capacity. None of our representations succeeds to reach a comparable level of MSE.

Second important observation is about our data in general. If we fit a strong non-linear model like Random Forest to our data, it's MSE on test set significantly differs from statistics we see on cross-validation. From one point it is an indicator for us to not over-promise and over-emphasize test set performance. From another point, it tells us that our test year might be anomalous and our model evaluation should consider both CV and test set scores of our models.

Third point is that TFIDF-based models seem to perform similarly for both linear SVR and non-linear RF with consistent behavior on both CV and test. PV features seem to be less stable in terms of consistency. They seem to benefit from higher dimensionality with linear model and become less stable with higher dimensionality with RF.

Although linear regression is considered a baseline, further in our study we will consider the test result of RF our true baseline which we should strive to improve.

C. Single representations with auto-regressive feature

Text representations alone can not independently compete with the auto-regressive baseline feature on our risk prediction problem. Thus in our next experiment we test whether our representations can single-handedly improve the baseline. Results of this experiment are given in Table.II.

TABLE II. ALL FEATURES WITH PREVIOUS VALUE

Representation	SVR MSE_{CV}	RF MSE_{CV}	SVR MSE_{Test}	RF MSE_{Test}
TFIDF, svd 100, Prev. value	0.174 ± 0.020	0.159 ± 0.027	0.204 ± 0.007	0.183 ± 0.001
TFIDF, svd 250, Prev. value	0.164 ± 0.022	0.160 ± 0.027	0.199 ± 0.006	0.183 ± 0.001
TFIDF, svd 500, Prev. value	0.160 ± 0.0204	0.159 ± 0.027	0.194 ± 0.004	0.184 ± 0.001
pv, 100, Prev. value	0.196 ± 0.016	0.194 ± 0.018	0.2048 ± 0.007	0.184 ± 0.001
pv, 250, Prev. value	0.192 ± 0.014	0.196 ± 0.018	0.202 ± 0.002	0.184 ± 0.001
pv, 500, Prev. value	0.193 ± 0.018	0.194 ± 0.018	0.206 ± 0.004	0.186 ± 0.001

We can clearly see significant difference in behavior of both representations with both model.

On TFIDF features RF significantly beats the baseline model in terms of CV performance by nearly 25%. Test set performance however only slightly increases the baseline result of RF by 1.5%. It is also worth noting that RF results are consistent across different amounts of components. SVR performance on TFIDF features benefits from more features, but doesn't reach comparable accuracy unlike previous experiment.

PV features demonstrate less accuracy improvement capacity in neither linear nor non-linear models. When interacting with the auto-regressive feature our models show only slight 4% improvements on CV. Test set performance exhibits similar pattern, however RF provides a slight 1% improvement.

For our further experiments we can consider dropping representation options with 500 components. In fact on both of our representations best RF models could keep up with 100 features of both representations only with visible performance drops.

D. Multi-representation fusion

Finally we examine two representation fusion scenarios, each with a separate experiment.

We start with the simplest fusion approach of direct feature concatenation. In this experiment we limit ourselves to only two combinations of feature counts instead of each possible pair. Based on previous results regression models performed without necessity to use maximal amount of available components. There is no clear indication whether best performance was achieved by representations with 100 or 250. Therefore two combinations of component amount were chosen: TFIDF 100 with PV 100 and TFIDF 250 with PV 250. Results of naive concatenation-based fusion are shown in Table.III.

One can note a slight overall improvement on 100-component data reaching 22% improvement on MSE_{CV} and 3% improvement on MSE_{Test}. However these results are very close to the TFIDF with 100 components from the previous experiment. Results from 250-component data are slightly worse, but still slightly ahead of the best previously achieved single representation result.

TABLE III. FUSION

Representation	SVR MSE_{CV}	RF MSE_{CV}	SVR MSE_{Test}	RF MSE_{Test}
TFIDF 100, pv 100, Prev. value	0.173 + 0.020	0.157 + 0.027	0.198 + 0.007	0.180 + 0.001
TFIDF 250, pv 250, Prev. value	0.163 ± 0.020	0.158 ± 0.027	0.196 ± 0.004	0.181 ± 0.001
TFIDF 100, Prev. value → pv 100	0.170 + 0.022	0.156 + 0.027	0.197 + 0.007	0.177 + 0.001
pv 100, Prev. value → TFIDF 100	0.190 ± 0.020	0.159 ± 0.027	0.200 ± 0.007	0.182 ± 0.001

Previously we considered fusion on the level of features. In our next experiment we consider multi-representation models from the viewpoint of model fusion. The experiment is based on the idea of out-of-fold (OOF) stacking, an ensembling technique where predictions from different cross-validation folds are combined.

First we build a model on one representation and use OOF predictions as the current target variable estimate. Then another model is built on the second representation to predict residuals between OOF predictions and real target variable. Technically it means that a model on one representation is trying to capture dependencies which couldn't be captured by a model on another. We limit ourselves only to 100-component representations. Results of this experiment are given in Table.III.

This approach achieves a new level of accuracy compared to the previously discussed ones. In particular the model that first builds on top of TFIDF and then is refitted on PV achieves an overall 23% improvement on MSE_{CV} and 5% improvement on MSE_{Test}. If we consider original linear baseline then it would be nearly 13% overall improvement.

It is interesting that a model that first build on PV and is later refitted on TFIDF is relatively weak compared even to naive feature fusion. However stacking-based approach proved to be the most efficient for our risk prediction tasks.

E. Results

Textual features provide a substantial basis for financial risk prediction accuracy improvements. Features derived from text representations can not compete with standard financial

baselines on their own, however they can be efficiently used together with these baselines to improve their performance.

We reached a total of 13% MSE improvement over official linear baseline and 5% improvement over our Random Forest baseline on test set. This result was achieved by fusing bag-of-words and paragraph embedding text representations of financial reports, on top of which Random Forest regression models were built.

VII. Discussion

We have shown that financial risk prediction models benefit from text derived features. One possible explanation is that our representations provide basis for different types of similarities between texts, with bag-of-words modeling the simplest word occurrences whereas paragraph and word embedding capture more sophisticated semantic features. One could investigate additional text representations like other parameterizations of embeddings (possible future semantic representations are proposed in [39]) or syntax trees [8] for scrapping even more accuracy in volatility prediction task.

Our approach to simultaneous exploitation of different text representations can be adopted in other NLP-fueled financial applications, especially ones that rely on text report data. In particular M&A prediction problems will greatly benefit from such analysis due to the same reasons it works for risks: basic word occurrence for merge consideration indicators have the potential for enrichment with semantics about non-organic growth plans [32]. Other major business events like bankruptcy and fraud can be considered as the basis for potential NLP applications as well. One can note that our representation fusion is naive and follows the stacking and blending techniques which are currently popular in the Data Mining Competitions [40]. It is feasible to justify this whole fusion process with layer-wise methodology similarly to the recent developments in multi-modal representation learning [29].

Despite accuracy improvement potential, we denote several multi-representation fusion drawbacks that should be considered. First of all handling multiple representations simultaneously requires an increased amount of computational resources and memory. It makes these models less desirable for production needs if model inference speed is important. Another problem is that the result of handling multiple representations within one regression model makes it complicated to evaluate resulting feature importances in the final fused model. And finally stacking-based fusion should be done carefully with maximum awareness to overfitting. These problems are not unique to machine learning and there are some developments like graceful degradation of large predictive models that help tackling them. Because even a 1% increase in accuracy may push an investment fund from making a loss, into making a little less loss.

Acknowledgment

The authors thank anonymous reviewers for their valuable input and comments on this paper. All computational work discussed in this paper was performed with tools under open license: scikit-learn - machine learning library for Python, gensim (https://radimrehurek.com/gensim/). The data was acquired from open resources and APIs: EDGAR Database (http://www.sec.gov/edgar.shtml) and Yahoo Finance API (http://finance.yahoo.com/).

References

[1] Z. Harris. “Distributional structure“, *Word*, 1954.

[2] E. F. Fama, “Efficient Capital Markets: A Review of theory and Empirical Work“, *The Journal of Finance*, Volume 25, Issue 2, pp. 383-417, May 1970.

[3] J. Nelder, R. Wedderburn, “Generalized Linear Models“, *Journal of the Royal Statistical Society*, 1972.

[4] Y. Yang and C. G. Chute. “A linear least squares fit mapping method for information retrieval from natural language texts“, *In Proc. of COLING*, 1992.

[5] Y. Yang and C. G. Chute. “An application of least squares fit mapping to text information retrieval“, *In Proc. of SIGIR*, 1993.

[6] C. Cortes, V. Vapnik, “Support-Vector Networks“, *Machine Learning*, Volume 20, Issue 3, pp 273-297, 1995.

[7] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik, “Support vector regression machines“ *In Advances in NIPS 9*, 1997.

[8] I. D. Baxter, A. Yahin, L. Moura, M. SantAnna and L. Bier, “Clone Detection Using Abstract Syntax Trees“, *Proceedings of ICSM98*, 1998

[9] A. Liaw and M. Wiener, “Classification and Regression by randomForest“, *R News*, 2002

[10] D. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. Journal of Machine Learning Research, 2003.

[11] B. Baesens, T. Van Gestel, S. Viaene, M. Stepanova, J. Suykens and J. Vanthienen, “Benchmarking state-of-the-art classification algorithms for credit scoring“, *Journal of the Operational Research Society*, 2003.

[12] J. Gimenez and L. Martinez, “SVMTool: A general POS tagger generator based on Support Vector Machines“, *In Proceedings of the 4th International Conference on Language Resources and Evaluation*, 2004

[13] I. Xiying Zhang, “Economic consequences of the SarbanesOxley Act of 2002“, *Journal of Accounting and Economics*, Volume 44, Issues 12, pp. 74-115, 2007.

[14] D. S. Palmer, N. M. O’Boyle, R. C. Glen, and J. B. O. Mitchell, “Random Forest Models To Predict Aqueous Solubility“, *Journal of Chemical Information and Modeling*, 2007

[15] R. Collobert and J. Weston, “A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning“, *Proceedings of the 25th international conference on Machine learning*, 2008

[16] S. Kogan, D. Levin, B.R. Routledge, J.S. Sagi, and N.A. Smith, “Predicting risk from financial reports with regression“, *NAACL 09*, pp. 272-280, 2009

[17] D. Lazer, A. Pentland, L. Adamic, S. Aral, A.-L. Barabsi, D. Brewer, N. Christakis, N. Contractor, J. Fowler, M. Gutmann, T. Jebara, G. King, M. Macy, D. Roy, M. Van Alstyne. “Computational social science“, *Science*, 323(5915):721-723, 2009.

[18] M. Mohler and R. Mihalcea, “Text-to-text Semantic Similarity for Automatic Short Answer Grading“ *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, 2009

[19] K. M. Quinn, B. L. Monroe, M. Colaresi, M. H. Crespin, D. R. Radev, “How to analyze political attention with minimal assumptions and costs“, *American Journal of Political Science*, 2010.

[20] P. D. Turney and P. Pantel, “From Frequency to Meaning: Vector Space Models of Semantics“, *Journal of Artificial Intelligence Research 37*, 2010

[21] M. Joshi, D. Das, K. Gimpel and N. A. Smith, “Movie Reviews and Revenues: An Experiment in Text Regression“, *HLT ’10 Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2010.

[22] D. Yogatama, M. Heilman, B. OConnor, C. Dyer, “Predicting a Scientific Communitys Response to an Article“, *Proceeding EMNLP ’11 Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2011.

[23] B. OConnor, D. Bamman, N. A. Smith, "Computational Text Analysis for Social Science: Model Assumptions and Complexity", *Second Workshop on Computational Social Science and Wisdom of the Crowds, NIPS*, 2011.

[24] S. M. Gerrish, D. M. Blei, "Predicting Legislative Roll Calls from Text", *Proceedings of the 28th International Conference on Machine Learning*, 2011.

[25] W. Zhang, T. Yoshida and X. Tang, "A comparative study of TFIDF, LSI and multi-words for text classification", *Expert Systems with Applications*, 2011

[26] Pedregosa et al., "Scikit-learn: Machine Learning in Python", *JMLR 12*, pp. 2825-2830, 2011

[27] Tim Loughran, Bill McDonald, "When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks", *The Journal of Finance*, 2011.

[28] S. L. Humpherys, K. C. Moffitt, M. B. Burns, J. K. Burgoon, W. F. Felix, "Identification of fraudulent financial statements using linguistic credibility analysis", *Decision Support Systems 50*, pp. 585-594, 2011.

[29] B. McFee and G. Lanckriet, "Learning Multi-modal Similarity", *JLMR*, 2011

[30] A. Oghina, M. Breuss, M. Tsagkias, and M. de Rijke, "Predicting IMDB Movie Ratings Using Social Media", *Proceeding ECIR'12 Proceedings of the 34th European conference on Advances in Information Retrieval*, 2012.

[31] E. H. Huang, R. Socher, C. D. Manning and A. Y. Ng, "Improving Word Representations via Global Context and Multiple Word Prototypes", *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, 2012

[32] G. Xiang, Z. Zheng, M. Wen, J. Hong, C. Rose and C. Liu, "A Supervised Approach to Predict Company Acquisition with Factual and Topic Features Using Profiles and News Articles on TechCrunch", *AAAI*, 2012

[33] M. Ghiassi, J. Skinner, D. Zimbra, "Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network", *Expert Systems with Applications*, pp. 6266-6282, 2013.

[34] A. K. Nassirtoussi, S. Aghabozorgi, T. Y. Wah, D. Chek Ling Ngo, "Text mining for market prediction: A systematic review", *Expert Systems with Applications 41 (2014) 7653-7670*, 2014.

[35] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu and Bing Qin, "Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification", *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 2014

[36] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. In International Conference on Machine Learning, 2014.

[37] A. M. Dai, C. Olah, Q. V. Le, and G. S. Corrado. Document embedding with paragraph vectors. In NIPS Deep Learning Workshop. 2014.

[38] O. Levy and Y. Goldberg, "Dependency-Based Word Embeddings", *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 2014

[39] Y. Liu, C. Sun, L. Lin, and X. Wang, "yiGou: A Semantic Text Similarity Computing System Based on SVM", *Proceedings of the 9th International Workshop on Semantic Evaluation*, 2015

[40] Kaggle: The Home of Data Science, Web: https://www.kaggle.com/