Stock Price Volatility Prediction: A Case Study with AutoML

Hilal Pataci
Rensselaer Polytechnic Institute, USA
patach@rpi.edu

Yannis Katsis
IBM Research, USA
yannis.katsis@ibm.com

Yunyao Li *
Apple, USA
yunyaoli@apple.com

Yada Zhu
IBM Research, USA
yzhu@us.ibm.com

Lucian Popa
IBM Research, USA
lpopa@us.ibm.com

Abstract

Accurate prediction of stock price volatility, the rate at which the price of a stock increases or decreases over a particular period, is an important problem in finance. Inaccurate prediction of stock price volatility might lead to investment risk and financial loss, while accurate prediction might generate significant returns for investors. Several studies investigated stock price volatility prediction as a regression task using the transcripts of earnings calls (quarterly conference calls held by public companies) with Natural Language Processing (NLP) techniques. Existing studies use the entire transcript, which can degrade performance due to noise caused by irrelevant information that may not have a significant impact on stock price volatility. In order to overcome these limitations, by considering stock price volatility prediction as a classification task, we explore several denoising approaches, ranging from general-purpose approaches to techniques specific to finance to remove the noise, and leverage AutoML systems that enable auto-exploration of a wide variety of models. Our preliminary findings indicate that domain-specific denoising approaches provide better results than general-purpose approaches, while AutoML systems show promising results.

1 Introduction

Predicting stock price volatility is of great interest to researchers and seems to remain one of the interesting open problems. Volatility is about information disclosure, and how unexpected the information is to the market; therefore, volatility persists in the market until the future values of stock reflect the information provided. According to (Fama, 1998), markets are informationally efficient if prices at each moment incorporate all available information about future values. Such that if there is an information disclosure, not yet incorporated in market prices, the future values will be volatile until the price fully reflects the disclosed information. (Lang and Lundholm, 1993; Baumann et al., 2004) Any information disclosed to the market by competitors, suppliers, customers, and regulators creates volatility, in addition to the internal information the company voluntarily discloses. Every quarter the executive leadership of a public company holds an earnings call meeting with investors and analysts to inform them about the company's status. As executives inform the investors about the company's current status and future outlook, earnings conference calls may result in stock price volatility. In finance and accounting research, the high volatility following an earnings conference call is conceptualized as Post-earnings Announcement Drift (PEAD) (Ball and Brown, 1968; Bernard and Thomas, 1989), which refers to the drift of a company's stock price for an extended period. Stock prices tend to drift upward (or downward) when the announcements are above (or below) expectations following an earnings conference call. Depending on how unexpected the information shared during earnings conference calls is, stock prices and firm valuations change, and markets become more 'informationally efficient' by absorbing this information over the long run (Fama, 1998; Fink, 2021).

In this work, we study the problem of leveraging the textual transcripts of companies' earnings calls and building Natural Language Processing (NLP) models to predict the volatility of their stock prices for a period of time following the earnings calls. While this problem has been studied in the literature, prior works exhibit these limitations:

- First, they model the problem as a regression task trying to predict the exact value of the stock price volatility. While this can be valuable in some settings, financial analysts are often interested in identifying the stocks with abnormally low or high volatility rather than identifying their exact value. This implies

*The work was done when the author was at IBM Research.

Proceedings of the Fourth Workshop on Financial Technology and Natural Language Processing (FinNLP), pages 48 - 57

the need to consider the problem as a classification task rather than a regression task as evaluated in prior work.

- Second, existing works typically leverage the earnings call transcripts as-is. However, transcripts contain a lot of irrelevant information for the purpose of stock price volatility prediction. This raises the question of whether this affects the performance of NLP models and whether there is an opportunity for improving such approaches by appropriately distilling these documents before feeding them into NLP models.

In this work, we address the aforementioned challenges as follows:

- We model the problem as a text classification task, where given the transcript of an earnings call one is asked to predict the stock price volatility as being low, medium, or high (Li and Lin, 2003). Considering this as a classification task also enables us to experiment with AutoML systems and democratization of this task by giving access to a wider user base that includes those without specialized knowledge of AI. To the best of our knowledge, this is the first work that models the stock price volatility prediction problem from earnings call transcripts as a classification problem and leverages associated NLP techniques.

- Earnings call transcripts include information that has almost no impact on the stock price volatility, and we conceptualize such irrelevant information as noise in our analysis. To improve the signal coming from the transcripts, we propose and experiment with an entire spectrum of denoising approaches, ranging from domain-agnostic denoising techniques to domain-specific approaches that utilize domain knowledge to improve the denoising process further. Our experimental evaluation shows that domain-specific denoising approaches outperform domain-agnostic techniques, which points to the importance of incorporating domain knowledge into the denoising process.

The rest of this paper is structured as follows: We start by reviewing related work in Section 2. In Section 3, we describe the problem definition and data preparation. We propose a range of denoising approaches in Section 4 and explain how we discover appropriate NLP models by leveraging an AutoML system in Section 5. Finally, we present the experimental evaluation results and associated insights in Section 6 and conclude the paper in Section 7.

2 Related Work

Information is one of the most valuable and highly sought assets in financial markets (Vlastakis and Markellos, 2012; Grossman and Stiglitz, 1980; French and Roll, 1986; Antweiler and Frank, 2004) and is found to be impacting stock price volatility in several studies. Moreover, as posited by the mixture of distributions hypothesis, the sequential arrival of new information generates trading volume and price movements (Clark, 1973; Tauchen and Pitts, 1983; Bessembinder and Seguin, 1992) (i.e. information shocks). Briefly stated, the impact of information disclosure on the volatility of stock prices has been investigated from several angles in the literature.

Four types of textual data have been mainly used for stock volatility prediction: Annual Statements, News, Social Media data, and Earnings calls transcripts.

Annual statements (10-K reports): Annual statements include historical data about a company's financial performance and a future outlook that can be valuable in predicting the volatility of its stock. For instance, Kogan et al. (2009) formulate a "text regression problem", where information from the 10-K reports is used to predict the volatility of stock returns in the periods following the reports. Loughran and McDonald's (Loughran and McDonald, 2011) financial lexicon generated from fourteen years of historical annual statements (10-K reports) is one of the major and initial attempts that utilize language resources to predict stock price volatility.

News data: News data often provide important information about events related to a company. For instance, Tetlock (2007) uses daily content from the Wall Street Journal to predict volatility. In a similar vein, Ding et al. (2014) adapt Open IE technology for event-based stock price movement prediction by extracting structured events from large-scale public news.

Social media data: Social media data often capture public sentiment about a company that can

be an important indicator for the future price of its stock. Bollen et al. (2011) use behavioral economics to investigate how societal moods affect collective decision-making for Dow Jones Industrial Average (DJIA)'s values.

Earnings calls transcripts: Several works have found that earnings calls (as captured through their transcripts) can be predictive of investor sentiment for stock price volatility prediction tasks (Frankel et al., 1999; Bowen et al., 2002; Cohen and Lou, 2012; Matsumoto et al., 2011). Recent studies also combined the textual transcripts with additional verbal and vocal cues from audio recordings of earnings call events and leveraged multi-modal learning to predict stock price volatility (Qin and Yang, 2019; Li et al., 2020). Most of the existing research models the stock price volatility prediction as a regression task, with the exception of Keith and Stent's work (Keith and Stent, 2019), which - similar to our work - models it as a classification task. However, they use classification to predict the analysts' recommendations to buy/sell/hold a stock. In contrast, we predict the market reaction itself by predicting the actual stock price volatility (classified as low, medium, and high volatility).

Our work makes multiple novel contributions: First, we consider stock price volatility prediction as a text classification task different from existing work (Qin and Yang, 2019; Li et al., 2020). Second, instead of using the earnings call transcripts as-is, we employ denoising techniques designed to separate the signal from the noise caused by irrelevant information in the transcripts and improve the performance of the resulting NLP models. We design and test several denoising approaches and report the results of their effectiveness. Third, AutoML systems have been used in several text classification tasks (Estevez-Velarde et al., 2019; Bisong, 2019; Blohm et al., 2020), however, there is little effort in the literature to use AutoML systems for the stock price volatility prediction task. Using AutoML systems to predict stock price volatility enables non-AI expert users (who may not be proficient in AI/NLP techniques) to create NLP models for the stock volatility prediction task quickly.

3 Problem Definition & Data Preparation

In this paper, we aim to predict the magnitude of volatility from earnings call transcripts and formulate this as a classification problem. In line with that purpose, we combine earnings call transcripts (text data) with their corresponding volatility labels (financial data).

3.1 Text Data

After each conference event, recordings of earnings conference calls are shared as audio and text files. In this work we focus on the transcripts of calls and leverage the earnings call transcripts dataset of Qin and Yang (Qin and Yang, 2019). Their dataset was built by collecting all S&P 500 companies' quarterly earnings conference call transcripts in 2017 from Seeking Alpha with written consent. It contains 576 conference calls, totaling 88,829 sentences. In order to avoid interference among different speakers, previous work (Qin and Yang, 2019) only processes the sentences of the most spoken executive (usually the CEO or CFO of the company). In the next stage, we use company names and earnings call dates collected from the dataset to retrieve the associated stock price information and compute the stock price volatility labels.

3.2 Financial Data

We manually collect the ticker symbols (an abbreviation used to uniquely identify publicly traded shares of a particular stock in a specific stock market) of these companies from Yahoo Finance with their corresponding company names obtained from the earnings call dataset (Qin and Yang, 2019). We use the ticker symbols of companies to extract their financial data and calculate stock price volatility labels by leveraging the Yahoo Finance API (Rekabsaz et al., 2017). Although Qin & Yang (Qin and Yang, 2019)'s dataset contains 576 conference call transcripts, due to missing financial information data on Yahoo Finance, we drop 27 transcripts, resulting in 549 transcripts that we use for our subsequent analysis.

We define stock price volatility prediction as a 3-class classification task; high-volatility, medium-volatility, or low-volatility for the respective company stock; similar to (Li and Lin, 2003). If the market reaction is almost neutral, we expect the stock price volatility to be low, so we label it as 'low volatility'. If the market reaction is high because there was too much unexpected news in the call, we expect the stock volatility to be high, so we label it as 'high volatility'. If the market reaction is mixed and in between neutral to high, we expect the stock volatility also to be in between,

and therefore we label it as 'medium volatility'.

$$v_{[0,n]} = \ln\left(\sqrt{\left(\frac{\sum_{i=1}^{n}(r_i - \bar{r})^2}{n}\right)}\right) \quad (1)$$

Consistent with prior work, we compute stock volatility for a period of n days following the earnings call event. We first calculate the absolute value of volatility as shown in Equation 1. In this equation, r_i is the stock return on day i and $\bar{r}$ is the average stock return in a window of n days. The return is defined as $r_i = (P_i - P_{i-1})/P_{i-1}$, where P_i is the adjusted closing price of a stock on day i.

Using this equation, we first calculate stock price volatility for four time periods of n = 3, 7, 15, and 30 days. Once the stock price volatility is calculated for 3 days using Equation 1, we calculate the thresholds for the high volatility, medium volatility, and low volatility labels by considering the distribution of volatility within our corpus for 3-days. Once we identify the range of stock price volatility for each category for 3-days, we apply the same range for 7-days, 15-days, and 30-days and label accordingly. Given that each stock volatility will fade over time, we use 3-days volatility ranges to identify the ranges for each class.

In particular, following an earnings call conference of Company A, if the stock price volatility of Company A is at the lowest 33% of the stock price volatility distribution, the transcript of that call is labeled as low-volatility. If the stock price volatility of Company B is between the 33% to 66% of the stock price volatility distribution, the transcript of that call is labeled as medium-volatility. Finally, if the stock price volatility of Company C is at the highest 33% of the volatility distribution, the transcript of that call is labeled as high-volatility for 3-days. Through this process, we identify the stock price volatility ranges that correspond to high, medium, and low volatility for 3-days and we use the same ranges to generate the volatility labels for the 7-day, 15-day, and 30-day stock price volatility. Figure 1 shows the resulting distribution of volatility labels for 3-days, 7-days, 15-days, and 30-days.

		Volatility over the following n-days			
		3-days	7-days	15-days	30-days
Labels	Low volatility	186	137	129	102
	Medium volatility	182	255	292	335
	High volatility	181	157	128	112

Figure 1: Distribution of volatility labels

4 Denoising Approaches

Earnings call transcripts are typically long documents containing a lot of information. While some of this information is valuable for predicting stock price volatility, another part of it can be irrelevant for the stock price volatility prediction task and can thus introduce unwanted noise. To address this problem, we experiment with several approaches of denoising the transcripts as a pre-processing step. We propose a spectrum of approaches, ranging from generic domain-agnostic approaches that are used in different tasks to more domain and task-specific approaches related to finance.

We start by using raw earnings call transcripts without further processing (which we use as our baseline). In the second approach, we use a general domain-agnostic denoising approach by leveraging the T5 summarization model (Raffel et al., 2019) to create a summary of the earnings call transcript. In the third approach, we experiment with a more domain and task-specific approach by borrowing a finance domain-specific dictionary (Loughran and McDonald, 2011), which we use to identify the sentences with important information. In the fourth approach, we create an intermediate domain-specific NLP model to identify the sentences containing important information that has the potential of affecting the stock price volatility.

4.1 Full document processing

In our first approach, we experiment with the full documents provided in (Qin and Yang, 2019) without any further processing. In this setting, we use the volatility labels calculated above and process the raw documents. The full document processing approach helps us identify how accurate stock price volatility prediction is when we process the earnings call transcripts without denoising or preprocessing. By considering full document processing as our baseline, we can also observe how other denoising approaches improve the model predictions.

4.2 General-purpose summarization of documents through T5

Sentences in a conference call have an order and relationships, leading to high dependency. Such that, a company executive answers a question and then motivates his/her answer with additional information in the following sentences. Drawing on this dependency, distilling the overall information

from the earnings call transcripts by summarizing the transcripts could potentially be an appropriate approach for removing the noise from an input document. Text-to-Text Transfer Transformer (T5) is an encoder-decoder model pre-trained on a multi-task mixture of unsupervised and supervised tasks (Raffel et al., 2019). T5 provides state-of-the-art results for various tasks such as translation and summarization. In this work, we implement summarization with Text-to-Text Transfer Transformer (T5) and consider this a domain-agnostic approach since it does not require domain-specific finance knowledge. We process each input paragraph separately while limiting the number of tokens in each iteration to less than 512. We then concatenate the summarized versions of subsequent paragraphs to create a summary of the entire earnings call transcript.

4.3 Application of the domain-specific dictionary of Loughran - McDonald

Even though there may be relations and dependency among sentences, some may not provide any information relevant to the stock price volatility, such as "good morning" or "thank you for your question." Moreover, some sentences might provide relatively more important information, while previous or following sentences may just be minor clarifications of the previous message. In this denoising technique, we leverage a domain-specific dictionary developed particularly for financial documents to identify sentences with relatively more important information. The dictionary of Loughran and McDonald (Loughran and McDonald, 2011) is one of the most recognized dictionaries in the financial literature and has been used for several different tasks (Keith and Stent, 2019; Rekabsaz et al., 2017). The LM dictionary provides the list of words with positive, negative, uncertainty, litigious, strong modal, weak modal, constraining, and complexity that exist in annual company statements (10-K documents). In this work, we use this dictionary as a benchmark to identify the sentences that contain relatively more important information that may impact stock price volatility. We search for words from the LM dictionary at the sentence level in each earnings call transcript and drop the sentences that do not contain any matching words with the LM dictionary. Finally, we append the filtered sentences and create a new distilled document for each earnings call transcript.

Label	Sentence
0: irrelevant	Thank you and good morning
1: buy	In Q4 we generated worldwide revenue growth of 7%.
2: sell	{our} prices declined 1% in Q4.

Table 1: Examples of labels for intermediate model

4.4 Creating an intermediate model to filter irrelevant data

Given that domain-specific knowledge can often help improve model performance, we also experiment by filtering sentences containing irrelevant information by building an intermediate task-specific filtering model. In this approach, we trained a separate model specifically for filtering out information that we believe may be irrelevant for the stock price volatility prediction task. To this end, we randomly selected 1,000 sentences from our corpus and labeled them to be used for training, validating, and testing purposes [1]. During labeling, sentences were labeled as 'buy', 'sell', or 'irrelevant'. As illustrated in Table 1, similar to previous research (Keith and Stent, 2019), sentences that provide positive information about the company were labeled as 'buy', sentences that provide negative information about the company were labeled as 'sell', and finally sentences that are generic or do not create an impact on the analysts' decision making were labeled as 'irrelevant'. We trained a BERT(base-cased) model by fine-tuning it on 70% of the labeled data (700 sentences) and used the remaining 15 % for test (150 sentences), and 15% for validation (150 sentences). We used the fine-tuned model to get 'buy'/'sell'/'irrelevant' predictions for the remaining sentences in the corpus (87,829 sentences in 549 earnings call transcripts). In a similar vein to the training data, sentences with similar positive information are expected to be labeled as 'buy' (1), sentences with negative information are expected to be labeled as 'sell' (2), and sentences with generic information are expected to be labeled as 'irrelevant' (0). In the final stage, we dropped sentences with generic information ('irrelevant'), and kept only sentences with 'buy' or 'sell' labels in each earnings call transcript with their corre-

[1]The labeling process was performed by one of the authors of this paper with relevant background.

sponding volatility label [2].

5 Building Classification Models through AutoAI for Text

Building models for NLP tasks, such as the stock price volatility classification task considered in this work, requires significant technical expertise, effort, and resources. To lower the barrier of entry and accelerate the model development process, the research and industrial community have developed AutoML/AutoAI techniques to automate parts of this process (Hutter et al., 2019; He et al., 2021; Wang et al., 2020). Multiple AutoML techniques suggested in the literature target different parts of the model development process. These include neural architecture search (He et al., 2021), hyperparameter optimization (Weidele et al., 2020), and others.

While previous works on stock volatility prediction using textual data leverage a small set of hand-picked NLP models, we explore AutoML techniques to select the best NLP model for the stock price volatility prediction task in this work. The goal behind this choice is twofold: First, we want to investigate how domain experts (in our case, financial analysts) can create NLP models for their tasks. Second, we want to explore multiple NLP model architectures and gain insights into which model architectures work best for the stock volatility prediction task.

We feed the denoised earnings calls transcripts and their corresponding labels into AutoAI for Text (Chaudhary et al., 2021). AutoAI for Text is a comprehensive end-to-end AutoML system for text classification tasks, which given a labeled text classification dataset explores a large search space of models for the provided dataset. During this search, AutoAI for Text explores multiple *featurizers* (such as GloVe, TFIDF, etc.), *estimators/transformers* (such as SVC, CNN, LSTM, etc.), and *hyperparameters*. The result of this optimization process is a set of NLP models for the given dataset (referred to as *pipelines*), ranked based on a chosen optimization metric (such as accuracy, precision, recall, F1, etc.). As we explain when describing the experimental evaluation in Section 6, AutoAI for Text also allows for various configuration options, including a specification of the set of models to explore, time budget that can be used for optimization purposes, the maximum number of candidate models to be trained, and others [3].

6 Experimental Evaluation

Experimental setting. For each earnings call transcript, we compute its volatility label for four different time periods, corresponding to 3, 7, 15, and 30 days following the earnings call. Through this process we obtain four different sets of labels for our earnings call transcripts (one per time period). In parallel, we run each transcript through the four denoising approaches outlined in Section 4. This leads to four sets of documents (one per denoising approach). For each (time period, denoising approach) pair, we combine the denoised transcript with the corresponding volatility label to generate a labeled dataset corresponding to the given time period and denoising approach. This labeled dataset is then fed into AutoAI for Text, which is tasked with discovering the best NLP model for the given pair.

Each labeled dataset is split into a 90% combined train and validation split and 10% test split. This split is done sequentially based on the timestamp of the earnings calls to ensure that we do not include in the train split any information about the time periods included in the test split (i.e., we want to avoid giving the model at training time information about the future). The combined train and test split is then further split by AutoAI for Text into train and validation utilizing another 90/10 split. In addition to the train/validation split ratio, we use the following configuration for AutoAI for Text: We assign an optimization time budget of 2 hours (i.e., instructing it to use up to 2 hours for optimization purposes) and ask it to explore and train at most 81 candidate models. We also select accuracy as the metric used both internally for optimization purposes and externally to report model performance. Finally, we instruct AutoAI for Text to explore a variety of estimators/transformers, which include SVC, CNN, LSTM, and BERT, and a variety of featurizers, which include GLoVe and TFIDF [4]. For

[2]Note that we modeled filtering as a ternary classification task, distinguishing between 'buy', 'sell', and 'irrelevant' sentences, to ensure that each class is homogeneous. However, in an alternative formulation, one could also model filtering as binary classification (with sentences labeled simply as 'relevant' or 'irrelevant').

[3]The focus of this work is not a comprehensive review of AutoAI for Text, but an investigation of how it can be used to solve the stock price volatility prediction task.

[4]It should be noted that all experiments were ran utilizing CPU (i.e., without GPU support), which may have affected the choice of BERT (which as we will see did not appear in

each (time period, denoising approach) pair, we select the model that AutoAI for Text has identified as having the highest accuracy on the validation set (which we refer to as the *best model*). The best model is then evaluated on the test set, computing its accuracy, which is the metric that we present in our evaluation results.

Baseline. Existing works on stock price volatility prediction from earnings call transcripts model the problem as a regression problem, however, as a baseline we use a simple approach that assigns to all transcripts the same label. For each (time period, denoising approach) pair, we report three versions of this baseline, depending on which is the common label assigned to all transcripts: L (low), M (medium), or H (high). For instance, in the L-baseline, all transcripts are predicted as being low volatility. While this is an admittedly simple baseline, it still allows us to understand whether the discovered models have identified a signal in the data or have simply learned to predict the most common label.

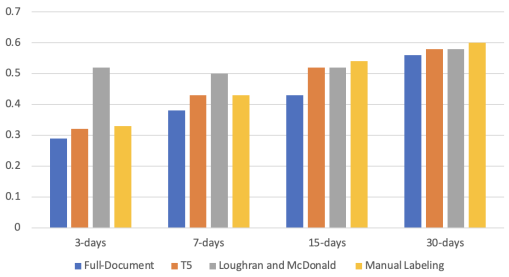

Figure 2: Accuracy of best discovered model for different denoising approaches and time periods

Results. The accuracy of the best model discovered by AutoAI for Text for each (time period, denoising approach) pair is shown in Figures 2 and 3. We next present and discuss the results by focusing on a few main questions that we hope to answer from this experimental evaluation.

Are the models able to identify a signal in the data? The first question we hope to answer is whether the discovered models have identified a signal in the input transcripts that allows them to predict the stock price volatility or whether they have simply learned to predict the most common label. This is a non-trivial question, as transcripts are long documents that in addition to information that may affect the stock price often contain a lot of irrelevant information. To answer this question, we compare in Figure 3 the accuracy of the best model discovered for each time period (shown in bold) with the accuracy of the best baseline (shown in italics).

As we can see, for all time periods, the former is always higher than the latter. Thus the best models seem to have successfully identified a signal in the transcripts that allows them to perform better than the baseline. For instance, the best model for the 3-day time period has an accuracy of 0.52, which is higher than the best baseline accuracy of 0.43 (which corresponds to predicting for every input transcript the most common label, which in this case is the high volatility). The gap between the best model and the baseline closes as the time period increases. While this is an interesting phenomenon that needs to be investigated further, a potential explanation is that transcripts may be more useful in predicting volatility for time periods immediately following the earnings calls, rather than for longer time periods [5].

Which denoising approaches perform best? The next question is identifying the best denoising approach for the studied problem. Which of the proposed denoising approaches should one choose for predicting stock price volatility and does the choice of the approach make a difference? Comparing the performance of the denoising approaches provides some interesting insights:

First, utilizing the full document (without any denoising) always yields the lowest performance. This shows that denoising approaches are important for distilling the long transcripts and making them more amenable for being used as training data for an NLP model.

Second, domain-agnostic denoising (such as the one provided by the T5 summarization approach) consistently underperforms domain-specific denoising approaches (such as the use of the domain-specific dictionary or the intermediate model). This shows that further applying domain knowledge to distill input documents can improve model performance.

Finally, while the best denoising approaches are the two domain-specific approaches, we observe that using the domain-specific dictionary of Loughran - McDonald is better for shorter time periods (i.e., time periods of 3 and 7 days), while using the intermediate denoising model based on

the results).

[5]The stock price may fluctuate due to other causes beyond what has been reported at the earnings call.

Denoising Approach	3-days volatility	7-days volatility	15-days volatility	30-days volatility
Baseline	L:0.25/M:0.30/*H:0.43*	L:0.12/*M:0.43*/*H:0.43*	L:0.10/*M:0.50*/H:0.38	L:0.12/*M:0.58*/H:0.29
Full Document	TFIDF+SVC (0.29)	TFIDF+SVC (0.38)	TFIDF+SVC (0.50)	GloVe+CNN (0.56)
T5	TFIDF+SVC (0.32)	GloVe+CNN (0.43)	TFIDF+SVC (0.52)	GloVe+CNN (0.58)
Loughran McDonald	**GloVe+CNN (0.52)**	**TFIDF+SVC(0.50)**	GloVe+CNN (0.52)	TFIDF+SVC (0.58)
Manual Labeling	TFIDF+SVC (0.33)	TFIDF+SVC(0.43)	**TFIDF+SVC(0.54)**	**GloVe+CNN (0.60)**

Figure 3: Accuracy of best model discovered by AutoAI for Text (together with the architecture of the model) for different denoising approaches and time periods. The baseline shows the accuracy that would be obtained if we assigned to all transcripts the same label of L: low, M: medium, or H: high volatility.

manually provided labels performs better for longer time periods (i.e., time periods of 15 and 30 days). This is an interesting result that we plan to explore and analyze further as part of our future work.

Which model architectures perform best? Finally, leveraging AutoAI for Text, we want to identify which model architectures perform best for the stock volatility prediction task. To aid in answering this question, Figure 3 includes the description of the best model discovered by AutoAI for Text. Each model is shown as $F + E$, where F is the featurizer and E is the estimator/transformer. For instance, GloVe+CNN is a model combining a GloVe featurizer with a convolutional neural network. As described above, in our experiments AutoAI for Text explored the GloVe and TFIDF featurizers. Similarly it searched among the following estimators/transformers: SVC, CNN, LSTM, and BERT.

By comparing the models reported in Figure 3, we can make the following observations: In all cases the models that perform best are based either on SVC and CNN combined with either GloVe or TFIDF featurizers. We cannot observe any systematic difference between SVC and CNN, leading us to believe that both work equally well for the studied problem. However, an important observation is that LSTM and BERT never appear among the best models.[6] However, both this as well as the performance of LSTM in this case our important results that we think are worth investigating further.

[6]All the experiments were done on a CPU-only machine. As such, we instructed AutoAI for Text to explore only CPU-friendly types of models. These are types of models that can be trained fast with CPU-only resources and include classical ones like SVC as well as faster deep-learning based models (CNN, LSTM). We left out BERT from the exploration space, since BERT works better when given GPU resources.

7 Conclusion and Future Work

Compared to existing work in the area, our work makes three main contributions: First, it models the problem as a text classification task (in contrast to the regression task considered before) and explores how one can leverage text classification models. Second, instead of just utilizing the long earnings call transcripts as-is, it explores the use of denoising approaches to distill the information found in the input documents and improve the performance of the learned models. We propose and explore an entire spectrum of denoising approaches, ranging from domain-agnostic techniques (such as general-purpose summarization models) to domain-specific techniques and compare their performance. Third, we leverage AutoML approaches to explore a range of NLP models and understand which model architectures perform best for the stock price volatility prediction task.

Our preliminary findings lead to several important insights. Denoising is shown to improve model performance with domain-specific denoising leading to bigger gains than domain-agnostic denoising approaches. Moreover, the use of AutoML leads to interesting insights on which model architectures perform best for the stock volatility task. We believe that these insights point to new interesting research directions both in developing better domain-specific denoising approaches, as well as further investigating which model architectures work best for long financial documents, which are some of the directions we plan to further explore in our future work.

References

Werner Antweiler and Murray Z Frank. 2004. Is all that talk just noise? the information content of internet stock message boards. *The Journal of finance*, 59(3):1259–1294.

Ray Ball and Philip Brown. 1968. An empirical evaluation of accounting income numbers. *Journal of accounting research*, pages 159–178.

Ursel Baumann, Erlend Nier, et al. 2004. Disclosure, volatility, and transparency: an empirical investigation into the value of bank disclosure. *Economic Policy Review*, 10(2):31–45.

Victor L Bernard and Jacob K Thomas. 1989. Post-earnings-announcement drift: delayed price response or risk premium? *Journal of Accounting research*, 27:1–36.

Hendrik Bessembinder and Paul J Seguin. 1992. Futures-trading activity and stock price volatility. *the Journal of Finance*, 47(5):2015–2034.

Ekaba Bisong. 2019. Google automl: Cloud natural language processing. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, pages 599–612. Springer.

Matthias Blohm, Marc Hanussek, and Maximilien Kintz. 2020. Leveraging automated machine learning for text classification: Evaluation of automl tools and comparison with human performance. *arXiv preprint arXiv:2012.03575*.

Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. *Journal of computational science*, 2(1):1–8.

Robert M Bowen, Angela K Davis, and Dawn A Matsumoto. 2002. Do conference calls affect analysts' forecasts? *The Accounting Review*, 77(2):285–316.

Arunima Chaudhary, Alayt Issak, Kiran Kate, Yannis Katsis, Abel Valente, Dakuo Wang, Alexandre Evfimievski, Sairam Gurajada, Ban Kawas, Cristiano Malossi, Lucian Popa, Tejaswini Pedapati, Horst Samulowitz, Martin Wistuba, and Yunyao Li. 2021. Autotext: An end-to-end autoai framework for text. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(18):16001–16003.

Peter K Clark. 1973. A subordinated stochastic process model with finite variance for speculative prices. *Econometrica: journal of the Econometric Society*, pages 135–155.

Lauren Cohen and Dong Lou. 2012. Complicated firms. *Journal of financial economics*, 104(2):383–400.

Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2014. Using structured events to predict stock price movement: An empirical investigation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1415–1425, Doha, Qatar. Association for Computational Linguistics.

Suilan Estevez-Velarde, Yoan Gutiérrez, Andrés Montoyo, and Yudivián Almeida Cruz. 2019. Automl strategy based on grammatical evolution: A case study about knowledge discovery from text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4356–4365.

Eugene F Fama. 1998. Market efficiency, long-term returns, and behavioral finance. *Journal of financial economics*, 49(3):283–306.

Josef Fink. 2021. A review of the post-earnings-announcement drift. *Journal of Behavioral and Experimental Finance*, 29:100446.

Richard Frankel, Marilyn Johnson, and Douglas J Skinner. 1999. An empirical examination of conference calls as a voluntary disclosure medium. *Journal of Accounting Research*, 37(1):133–150.

Kenneth R French and Richard Roll. 1986. Stock return variances: The arrival of information and the reaction of traders. *Journal of financial economics*, 17(1):5–26.

Sanford J Grossman and Joseph E Stiglitz. 1980. On the impossibility of informationally efficient markets. *The American economic review*, 70(3):393–408.

Xin He, Kaiyong Zhao, and Xiaowen Chu. 2021. Automl: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212:106622.

Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. 2019. *Automated machine learning: methods, systems, challenges*. Springer Nature.

Katherine A Keith and Amanda Stent. 2019. Modeling financial analysts' decision making via the pragmatics and semantics of earnings calls. *arXiv preprint arXiv:1906.02868*.

Shimon Kogan, Dimitry Levin, Bryan R. Routledge, Jacob S. Sagi, and Noah A. Smith. 2009. 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*, pages 272–280, Boulder, Colorado. Association for Computational Linguistics.

Mark Lang and Russell Lundholm. 1993. Cross-sectional determinants of analyst ratings of corporate disclosures. *Journal of accounting research*, 31(2):246–271.

Jiazheng Li, Linyi Yang, Barry Smyth, and Ruihai Dong. 2020. MAEC: A multimodal aligned earnings conference call dataset for financial risk prediction. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, CIKM '20, page 3063–3070, New York, NY, USA. Association for Computing Machinery.

Ming-Yuan Leon Li and Hsiou-Wei William Lin. 2003. Examining the volatility of taiwan stock index returns via a three-volatility-regime markov-switching arch model. *Review of Quantitative Finance and Accounting*, 21(2):123–139.

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

Dawn Matsumoto, Maarten Pronk, and Erik Roelofsen. 2011. What makes conference calls useful? the information content of managers' presentations and analysts' discussion sessions. *The Accounting Review*, 86(4):1383–1414.

Yu Qin and Yi Yang. 2019. What you say and how you say it matters: Predicting stock volatility using verbal and vocal cues. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 390–401, Florence, Italy. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Navid Rekabsaz, Mihai Lupu, Artem Baklanov, Allan Hanbury, Alexander Dür, and Linda Anderson. 2017. Volatility prediction using financial disclosures sentiments with word embedding-based ir models. *arXiv preprint arXiv:1702.01978*.

George E Tauchen and Mark Pitts. 1983. The price variability-volume relationship on speculative markets. *Econometrica: Journal of the Econometric Society*, pages 485–505.

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

Nikolaos Vlastakis and Raphael N Markellos. 2012. Information demand and stock market volatility. *Journal of Banking & Finance*, 36(6):1808–1821.

Dakuo Wang, Parikshit Ram, Daniel Karl I Weidele, Sijia Liu, Michael Muller, Justin D Weisz, Abel Valente, Arunima Chaudhary, Dustin Torres, Horst Samulowitz, et al. 2020. Autoai: Automating the end-to-end ai lifecycle with humans-in-the-loop. In *Proceedings of the 25th International Conference on Intelligent User Interfaces Companion*, pages 77–78.

Daniel Karl I Weidele, Justin D Weisz, Erick Oduor, Michael Muller, Josh Andres, Alexander Gray, and Dakuo Wang. 2020. Autoaiviz: opening the blackbox of automated artificial intelligence with conditional parallel coordinates. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, pages 308–312.