FINSENTIMENT: PREDICTING FINANCIAL SENTIMENT AND RISK THROUGH TRANSFER LEARNING

A Thesis

by

Zehra Erva Ergün

Submitted to the
Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master's Degree

in the
Department of Computer Science

Özyeğin University
December 2022

FINSENTIMENT: PREDICTING FINANCIAL SENTIMENT AND RISK THROUGH TRANSFER LEARNING

Approved by:

Assistant Prof. Emre Sefer, Advisor
Dept. of Computer Science
Özyeğin University

Prof. Olcay Taner Yıldız
Dept. of Computer Science
Özyeğin University

Asst. Prof. Reyyan Yeniterzi
Dept. of Computer Eng.
Sabancı University

Date Approved: December 2022

To Kübra, Cihangir & Necip Fazıl Ergün, Shevek

ABSTRACT

There is an increasing interest in financial text mining tasks. Significant progress has been made by using deep learning-based models on generic corpus, which also shows reasonable results on financial text mining tasks such as financial sentiment analysis. However, financial sentiment analysis is still a demanding work because of insufficiency of labeled data for the financial domain and its specialized language. General-purpose deep learning methods are not as effective mainly due to specialized language used in the financial context. In this study, we focus on enhancing the performance of financial text mining tasks by improving the existing pretrained language models via NLP transfer learning. Pretrained language models demand a small quantity of labeled samples, and they could be enhanced to a greater extent by training them on domain-specific corpora instead. We propose an enhanced model FinSentiment, which incorporates enhanced versions of a number of recently-proposed pretrained models, such as BERT, XLNet, RoBERTa to better perform across NLP tasks in financial domain by training these models on financial domain corpora. The corresponding finance-specific models in FinSentiment are called Fin-BERT, Fin-XLNet, and Fin-RoBERTa respectively. We also propose variants of these models jointly trained over financial domain and general corpora. Our finance-specific FinSentiment models in general show the best performance across 3 financial sentiment analysis datasets, even when only a subpart of these models are fine-tuned with a smaller training set. Our results exhibit enhancement for each tested performance criteria on the existing results for these datasets. Extensive experimental results demonstrate the effectiveness and robustness of especially RoBERTa pretrained on

financial corpora. Overall, we show that NLP transfer learning techniques are favorable solutions to financial sentiment analysis tasks.

Financial risk is empirically quantified in terms of asset return volatility, which is degree of deviation from the expected return of the asset. Under risk management in finance, predicting asset volatility is one of the most crucial problems because of its important role in making investment decisions. Even though a number of previous studies have investigated the role of natural language knowledge in enhancing the quality of volatility predictions, volatility estimation can still be enhanced via recent deep learning techniques. Specifically, extracting financial knowledge in text through transfer learning approaches such as BERT has not been used in risk prediction. Here, we come up with RiskBERT, the first BERT-based transfer learning method to predict asset volatility by simultaneously considering both a broad set of financial attributes and financial sentiment. In terms of language dataset, we utilize transcripts from the annually occurring 10-K filings of the publicly trading companies to train our model. Our proposed model, RiskBERT uses attention mechanism to model verbal context and remarkably performs better than the state-of-the-art methods and baselines such as historical volatility. We observe such outperformance even when RiskBERT is finetuned with a smaller training set. We found RiskBERT to be more effective in risk prediction after the Sarbanes-Oxley Act of 2002 has passed since such legislation has made the annual reports more effective. Overall, we show that NLP transfer learning techniques are favorable solutions to financial risk prediction task. Our pretrained models, and source code will be publicly available once the review is finished.

ÖZETÇE

Son zamanlarda, finansal metin madenciliğine ilgide artış bulunmaktadır. Derin öğrenmeye dayalı modellerle genel korpuslarda yapılan çalışmalarda hayli yol katedildi ki bu modellerle finansal metin madenciliği işi olan finansal duygu analizinde de mantıklı sonuçlar elde edildi. Ancak, finansal duygu analizi, finans alanında etiketlenmiş veri eksiliğinden ve özelleşmiş bir dili olmasına bağlı olarak hala zahmetli bir iştir. Finans kapsamındaki bu özelleşmiş dilden dolayı genel amaçlı derin öğrenme metotları çok verimli çalışmıyor. Bu çalışmada, halihazırda bulunan dil modellerini transfer öğrenmesi ile geliştirerek finansal metin madenciliği problemlerindeki performansı iyileştirmeye odaklandık. Önceden eğitilmiş dil modelleri daha az miktarda etiketlenmiş veri örneklemi talep eder ve alana özel bir korpora ile eğitildiğinde daha da büyük çapta performansa sahip olur. Biz, son dönemlerde üzerine çalışılmış olan BERT, XLNet ve RoBERTa modellerini içeren, iyileştirilmiş bir model olan FinSentiment 'i sunuyoruz. Bu modeller de doğal dil işleme problemlerinde daha iyi bir performans göstermeleri için finans alanında bir korpora ile eğitildiler. Sırasıyla finans alanı özelinde eğitilmiş FinSentiment modelleri: FinBERT, Fin-XLNet ve Fin-RoBERTa. Bu FinSentiment modelleri genel olarak 3 finansal duygu analizi verisetlerinde en iyi performansı gösterdiler ki bu modellerin bir parçası küçük bir set ile eğitilmişti. Test edilen veri setleri için baz aldığımız performans kriterine göre olan sonuçlarımızda iyileşme gözlemlendi. Daha detaylı sonuçlarda ise özellikle finans korporası ile eğitilmiş RoBERTa modelinin verimliliğini gösterdi. Nihayetinde, transfer öğrenmesi teknikleri duygu analizi çalışmalarında elverişli çözümler sunduğunu göstermiş olduk.

Finansal risk ampirik olarak, varlık getirisi volatilitesi üzerinden ölçülür. Varlık getirisi volatilitesi, varlığın beklenen getirisinden ne kadar sapma olduğunun bir ölçütüdür. Finans'ta risk yönetimi altında, yatırım kararı vermekteki önemine binaen varlık volatilitesi tahmini en mühim problemlerden biridir. Doğal dil işleme yöntemlerini kullanarak volatilite tahminlemesinde verimliliğ arttırmaya yönelik birçok çalışma olmasına rağmen, derin öğrenme teknikleriyle volatilite tahminlemesi daha da iyileştirilebilir. özellikle, risk tahminlemesinde BERT gibi transfer öğrenmesi yaklaşımını kullanarak metinlerden finansal anlam çıkaran yöntemler kullanılmamıştır.

Çalışmamızda öncelikle, BERT'e dayalı transfer öğrenmesi metodu ile varlık volatilitesi tahminlemesi yapan RiskBERT 'i sunuyoruz. Dil veriseti olarak modelimizi eğitirken halka açık şirketlerin yıllık yayınlanan 10-K belgelerini kullandık. RiskBERT modeli, sözcüklere dayalı bağlamı modellerken bir "attention mekanizması" kullanır ve modern yöntemler ile dayanak noktası olan geçmiş volatilite değerine göre dikkate değer bir performans göstermiştir. Bu dikkate değer performans, çok küçük bir veri setiyle eğitilerek göstermiştir. özellikle 2002 yılında geçen Sarbanes-Oxley yasası sonrası raporların daha etkili bir şekilde yazılması zorunlu tutulduğundan, RiskBERT de bu yasa sonrası daha iyi bir performans sergilemiştir. özetle, risk tahminlemesinde de transfer öğrenmesi teknikleri olumlu sonuçlar vermiştir.

ACKNOWLEDGEMENTS

First and foremost I am extremely grateful to my advisor, Asst. Prof. Emre Sefer for his invaluable advice, continuous support, and patience during my thesis and research. I would also like to thank my mom and dad, whom without this would have not been possible. I also appreciate all the support I received from my brother.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Transfer Learning

Asset prices in an open economy mirrors all the existing knowledge about these assets trading in a market [3]. When a novel knowledge is introduced, all performers in the market improve their positions, and accordingly asset prices are updated. Such adjustments make it almost impossible to consistently outperform the market. On the other hand, "novel knowledge" is not clearly defined, and its definition may update while novel information-retrieval technologies appear. As a result, in the shorter term, early adopters of those technologies may have the edge on many financial tasks such as asset price prediction.

The number of Natural Language Processing (NLP) resources and methods has recently been increasing, especially for more specialized domains. They have become much more mature, and have remarkably changed the financial domain's landscape. Especially, portfolio managers have a major interest in utilizing NLP methods to track real-time market sentiment from Twitter posts and online news articles, as they can use the extracted sentiment signal for trading assets. Instinctively, a particular company's stock price is expected to drop if negative knowledge exists about the company, and vice versa. For instance, according to Bloomberg, the historical performance of sentiment portfolios is better than the benchmark portfolios by a significant margin [4]. Additionally, prior research on finance finds that company's performance and market returns can be predicted reasonably well by using the sentiment signal obtained from social media datasets [5].

The volume of financial communication text continues to keep increasing since an

incredible number of such texts are being created on a daily basis. In the finance sector, along with various fields, automatically harnessing financial text data is an inevitable measure that should be taken to predict and analyze future market trends in finance and economics. Here, financial text mining partakes an important role to aggregate official company announcements or analyst reports since manual analysis of such texts and gaining actionable insights from them are quite difficult tasks.

Financial sentiment analysis is the main focus of this paper, which categorizes the text as neutral, negative, or positive in financial domain. We need to handle 2 challenges while working on this financial sentiment analysis task: 1- The up-to-date sentiment categorization techniques mainly use neural networks while requiring significant amount of tagged financial datasets. In this case, due to the required background information in the economics or finance fields, annotators for tagging should be the people who are specialized in such fields which restricts the amount of possible annotators. Labeling these finance-specific textual fragments requisites an experience and expertise which can be extremely expensive. 2- The language of financial statements is task-specific where the statements use a specialized words and they are mainly made up of unclear expressions instead of more easily-comprehended positive or negative words. So, the models trained over more common corpus do not fit well specifically to the financial sentiment analysis tasks. One way to tackle having less annotated data in finance is to construct language models (LM), which is a key step to achieve state-of-the-art results in many different Natural Language Processing (NLP) tasks, for instance, sentiment analysis which is the main focus of this paper. Language models are techniques used to learn the probability of a given sequence of words occurring in a sentence. There are different kinds of LMs using statistical techniques or neural network algorithms to construct a base for the word predictions. Considering statistical LMs with thoroughly and manually crafted

financial sentiment lexicons [6] as an answer to the problem of finance having domain-specific language would not be a good solution since such methods are based on word counting, which fails to analyze the deeper semantic meaning of the provided text.

NLP transfer learning approaches come into the minds as an effective strategy to solve the two aforementioned challenges [7]. In this case, one can use word embeddings and distributed word representations like GloVe [8] and Word2vec [9] which are pretrained with a large unlabelled data to initialize first layer of a neural network model. Then, one can typically train the other layers with a dataset specialized for the task [10]. These two methods may capture the semantic meanings of words, however, they are ineffective at extracting higher-level concepts in context such as polysemous disambiguation, syntactic structures, semantic roles, anaphora [11]. The up-to-date methods utilize pretrained LMs to overcome the unsuccessful behaviour of context understanding for previous methods. The main point of NLP transfer learning approaches can be summarized in two steps: 1- We train these LMs on a major size corpus, 2- Then, we initialize the downstream models by using the parameters learned over LM task. By following these two consecutive steps, we can achieve a significantly better performance. In this case, initialized layers may be the overall model [12], or even the exclusive word embedding layer [13]. In principle, NLP transfer learning shall become a solution to the problem of scarce tagged data. These transfer-learning based LMs do not ask for associated tags or labels as they are rather based on forecasting the following word. These LMs focus on learning the representation of the semantic knowledge. Once such representation is learned, finetuning on the tagged dataset turns into a way to learn the usage of semantic knowledge in predicting the labels.

Transfer learning's main advantage is its capability to additionally pretrain the LMs on domain-specific unlabeled corpora. As a result, the trained model may capture the semantic relationships in the target domain's textual content. This extra

semantic relationship learning is important since the distribution of words in the target corpora is probably unlike general corpora. This proposed pretraining method is particularly important and effective for the niche financial domain as its vocabulary and the language content are remarkably dissimilar than a general vocabulary and language. In this research, we investigate three transfer learning-based deep learning models for financial sentiment analysis tasks. One of them, BERT (Bidirectional Encoder Representation from Transformer) [14] is based on Transformer structure [15] which is developed around the self-attention concept. Additionally, RoBERTa (Robustly optimized BERT) [16] is an optimized version of BERT. Lastly, XLNet (Transformer-XL Network) [17] is another transformer-based model using autoregression. We have also investigated several features of the introduced models, such as the effects of training strategies in preventing catastrophic forgetting, the effects of extra pretraining over financial corpus, and finetuning solely a tiny subset of model layers in order to decrease the training time without a remarkable performance decrease. Among them, catastrophic forgetting [18] is a common problem in transfer learning, which means the pre-trained knowledge can be removed during learning of new knowledge. Therefore, we also investigate whether various transformer-based models (BERT, RoBERTa and XLNet) suffer from the catastrophic forgetting problem.

As a summary, we have the following main contributions in this paper:

- We introduce financial extensions of the recently-introduced transformer-based models (BERT, RoBERTa and XLNet) for financial NLP tasks, in order to transfer knowledge from financial domain corpora. By calling these financial extensions as FinSentiment, we evaluate the performance of FinSentiment over 3 financial sentiment analysis datasets.

- We conduct a number of experiments on several Financial PhraseBank, AnalystTone, and FiQA sentiment scoring benchmark datasets. We achieve the

state-of-the-art results on all these financial datasets.

- We compare the performance of various transformer-based models (BERT, RoBERTa and XLNet) and understand why certain models can perform better on financial text mining tasks. Our models are capable of efficiently capturing the language knowledge and semantic information in large-scale pretraining corpora.

- We have investigated several features of the introduced models, such as the effects of training strategies in preventing catastrophic forgetting, the effects of extra pretraining over financial corpus, and finetuning solely a tiny subset of model layers in order to decrease the training time without a remarkable performance decrease.

- We implement our transformer-based algorithms by using TensorFlow and Hugging Face frameworks [19, 20]. We make the source code and pretrained models publicly available with the minimal number of task-specific architecture modifications.

The remaining parts of this paper is organized as follows: Section 1.3 discusses the related work on both more general sentiment analysis as well as finance-specific sentiment analysis via pretrained models. Next, Section 2 defines the evaluated models, including the existing models and FinSentiment. Section 3 follows this section focusing on our experimental setup to evaluate the performance of FinSentiment over the sentiment analysis tasks. Results section 4 comes after section 3 focusing on experimental results on three finance-specific sentiment analysis tasks as well as discussion and analysis of FinSentiment results across multiple points of view. Lastly, Section 5 summarizes our findings and concludes the paper.

1.2 Volatility Prediction

Financial risk prediction is one of the main components of modern portfolio selection, and it is clearly interesting not only to analysts and traders on the Wall Street, but also almost for anyone who has invested in assets such as publicly listed companies stocks. Financial risk of asset is measured by asset return volatility in almost all scenarios, which is degree of fluctuations of a given asset's return. Financial risk is one of the key determinants and drivers for a company's value. It is also important for adaptation into different market conditions, so its proper and precise estimation is crucial in financial investment decision making. As a result, volatility prediction is one of the most crucial problems across risk management area in finance [21]. Although direct predictability of asset prices is controversial in economics, this is not the case for predicting an asset's volatility level by utilizing publicly available knowledge [22].

Even though volatility prediction has in general many applications in finance such as algorithmic trading, the methods to predict volatility are not solely beneficial to financial domain. Instead, such methods are promising across many settings and domains. For instance, volatility forecasting tools have been quite useful in predicting the presidential approval rates, forecasting weather, and modeling muscular activities in neurons [23].

Predicting financial risk has conventionally focused on quantitative modeling of historical time-series financial data. For instance, GARCH model [24] has been used extensively to predict volatility by using only factual market data such as historical prices. However, growing number of late financial studies have also used textual datasets, where these studies extract and quantify the financial disclosures sentiment [6]. Given the importance of risk prediction problem in finance, applying Natural Language Processing (NLP) techniques to risk prediction has begun to take

more attention in recent years. There are not many studies that perform a text-based volatility prediction by utilizing different types of financial disclosures content [25, 26, 27]. In more detail, asset volatility prediction has been studied by using a number of different textual sources, including company earnings call transcripts [28], social media [29], public news articles [30], and company disclosed reports [31].

Among these textual resources, company's annual disclosures, known as 10-K filing documents, are interesting for risk prediction. These filings are mandated by government which are hypothesized to incorporate a significant amount of knowledge regarding companies and their values. These filings also include detailed knowledge about risk factors for companies and companies' business. For instance, *Item 1A - Risk Factors* section includes knowledge regarding the most noteworthy potential risks for the company. Nevertheless, one main problem with these reports is that they are long, complex, redundant, and cannot be easily comprehended due to their complexity. As noted in [32], over 20 years of formal education is required to process these reports properly. [32] has found a significant increase in the report lengths and the number of covered topics over the years, which makes the manual processing of reports almost impossible. In our case, whether these mandatory disclosures provide any extra knowledge is an important question as these reports are expensive to form and they are supposed to protect investors.

In this case, an emerging question is how we can better attack the problem of combining factual market knowledge and mostly unstructured textual data. Deep learning seems like a good solution to combine them, and efficiently extract the sentiment information hidden in the text.

Here, we integrate the the existing information in financial domain and risk management with the latest enhancements in NLP and transfer learning to come up with RiskBERT, first financial sentiment-based transfer learning approach to jointly consider textual and financial data to predict asset volatility. In our case, we have

gathered a detailed list of historical financial data, and have enriched it with natural language knowledge disclosed in recurring annual filings of publicly traded companies which summarize company fundamentals, discussion of current company status, as well as company management's future expectations. Afterwards, we train RiskBERT to predict risk for a short period following those filings. Most of the studies so far have mainly focused on using the 10-K documents before 2008. There is a only a few studies on how informative and effective the recent reports are in terms of predicting volatility. The time period we have included in our analysis includes both 2008 economic crisis and COVID crisis. Via our analysis, we show that report contents keep frequently changing not only relative to 2008, but instead in a 3-4 year cyclic period.

1.3 Previous Work

1.3.1 Sentiment Analysis

Sentiment analysis, which is a demanding NLP task for domain-specific fields, infers people's opinions or sentiments from their textual language [33]. The recent studies can be examined across two different categories: 1- Traditional machine learning approaches where attributes are inferred from text by word counting [34, 35, 36, 37], and 2- Deep learning-based approaches where a number of embeddings represent a given text [38, 39, 40]. More traditional machine learning-based approaches cannot easily represent the semantic knowledge resulting from a particular word sequence. On the other hand, deep learning-based approaches may require huge amount of data since their parameter space is much larger than the more traditional approaches [41].

Financial sentiment analysis is different than the more general analysis of sentiment in terms of both aim and domain perspectives. Financial sentiment analysis, as a purpose, mainly guesses the reaction of markets to the knowledge introduced in the text [42]. [6] introduces a detailed analysis on machine learning-based financial text analysis that includes lexicon-based and bag-of-words based approaches. For

instance, [6] generates a financial terms dictionary, where they assign negative, positive, uncertain tags to the terms in the dictionary, and assess a document's tone over extracted words appearing in this dictionary [43]. [44] investigates the semantic orientations on economic texts where they have also generated directional words lexicon and financial phrase-bank for financial sentiment analysis. [45] focuses on detecting the sentiment by extracting n-grams over tweets having financial knowledge, and feeding them into a machine learning method.

In terms of deep learning approaches, [46] has applied LSTM (Long Short-term Memory) to analyze financial textual polarity. They have used LSTM over company disclosures in predicting stock price changes, where their approach has outperformed traditional ML-based methods. In comparison to our work, they have pretrained their method over a greater corpora. Their method is more restrictive since they pretrain over a labeled dataset whereas our approach FinSentiment pretrains a LM in an unsupervised way. A number of other studies employing neural models also focus on analyzing the financial sentiment. [47] have applied a number of neural network models over a StockTwits dataset, and have found CNN (Convolutional Neural Network) as the top performer. [48] generate sentence embeddings by Doc2vec over company announcements, and predict stock market by multi-instance learning. [49] focus on classifying a list of financial text sentences by their sentiment via utilizing LSTM and text simplification. Moreover, aspect-based sentiment analysis (ABSA), as a text analysis type, clusters opinions via aspect and captures the sentiment relevant to each aspect [1] where deep learning applications has recently started to appear.

Neural networks do not achieve a higher performance in analyzing financial sentiments since labeled financial datasets are missing. Even though the pretrained values are used on the initial neural network layers, the remaining layers yet learn complex associations by using labeled dataset with fewer samples. In this case, a better approach would be to initialize the whole model with pretrained values, and fine-tune

these values in regards to the following classification task.

1.3.2 Text Classification over Pretrained Models

Language modeling predicts the succeeding word for a given text. Recently, it has been realized that we can favorably apply a NLP model trained for language modeling to the remaining downstream NLP tasks with minor changes by finetuning. Those models are in general trained over a larger corpus, and then are finetuned over the target dataset by integrating extra task-specific layers [50]. One main example use case for such method is text classification, as focused in this paper. ELMo (Embeddings from Language Models) [12] can be considered as the initial application of such finetuning task. ELMo includes a bidirectional deep language model which is pretrained on a larger corpus. It comes up with a contextualized representation for each word by using the hidden states. Using the contextual embeddings for downstream tasks enhances the performance over many tasks when compared to using the static embeddings extracted by Word2vec [9]. Another model ULMFit (Universal Language Model Fine-tuning) [13] achieves transfer learning for NLP tasks by integrating novel approaches such as gradual unfreezing, slanted triangular learning rates, and discriminative finetuning which sustain information during the training period. They have additionally pretrained language model on domain-specific corpora, where they assumed that the distribution of the target tasks dataset is not same as the one which is used to train the initial model.

Transformer-based language models are utilized in several different fields of NLP [51]. As an example, BERT (Bidirectional Encoder Representations from Transformers) [14] has enhanced this principal idea of effectively finetuning a pretrained language model for downstream tasks. BERT has 2 key differences from the previous work: 1- BERT is composed of a huge neural network which is trained on an extremely large corpus, 2- BERT focuses on forecasting randomly-masked tokens over a given token sequence

instead of the subsequent token, as well as on classifying two sentences whether they follow each other or not. Due to those 2 enhancements, BERT has achieved the best results in a number of NLP tasks including question answering. [17] has introduced a generalized autoregressive pretraining approach XLNet which has 2 key differences than BERT: 1- XLNet's autoregressive formula is useful in solving a number of limitations of BERT, 2- XLNet learns bidirectional contexts via optimizing the expected likelihood over the factorization order's all possible permutations. In comparison to XLNet, BERT depends on noising the input sequence with masks. It suffers from a pretrain finetune discrepancy by neglecting the dependencies among the masked parts. XLNet has significantly outperformed BERT on a number of tasks such as sentiment analysis, document ranking, etc. Similarly, [16] has found that BERT has been remarkably undertrained, and as a result, introduced RoBERTa which is an enhanced version of BERT in terms of training size and multiple principal hyperparameters. Their results exhibit the importance of previously-overlooked design criteria in BERT. The details of finetuning for these recently-introduced models BERT, XLNet, and RoBERTa have not been analyzed in detail for text classification. One such study [52] has carried out experiments over a number of BERT configurations for text classification.

There are a number of previous attempts to use BERT model on financial corpora [53, 54, 55]. [54] have proposed a model named FinBERT where they have replaced the masked language modeling and next sentence prediction stages of BERT with various different kinds of financial text mining tasks such as dialogue relation task, sentence distance task, token-passage prediction task, etc. They have outperformed BERT across various domain-specific tasks, where they have also trained FinBERT on both general and financial corpora. Additionally, [53] has also come up with a different FinBERT model to investigate the prediction performance for the same tasks. He pretrained a BERT model once more with a financial dataset to

compare it with the previously suggested ULMFit and ELMo models, as well as to arrange an experimental setup in a way that catastrophic forgetting effects and layer contributions on FinBERT have been observed. Furthermore, [55] have introduced an another FinBERT-named model. They have finetuned BERT over financial datasets including analyst reports in accounting and finance, where their study has indicated another success of domain-specific pretraining over the general BERT model. In addition to all, one more FinBERT model has been introduced by [56] which uses SEC filings for a number of NLP tasks such as next sentence prediction and masked language modeling. They have also investigated the language change for finance, and their models outperformed BERT on these NLP tasks.

Furthermore, solutions to domain-specificity have been introduced on different research areas, such as defining a specialized language on biomedical studies. [57] have implemented BioBERT for biological text mining tasks, which has showed outstanding performance over a number of such tasks including biomedical question answering, biomedical relation extraction, and biomedical named-entity recognition. [58] have presented an unsupervised data-driven approach to represent and extract features for biological sequences, which are in turn used in a number of machine learning applications.

1.3.3 Risk Prediction

Market prediction has been attracting much attention in recent years in the natural language processing community. Kazemian and his study group [59], used sentiment analysis for predicting stock price movements in a simulated security trading system using news data, showing the advantages of the method against simple trading strategies. Ding et al. [60] addressed a similar objective while using deep learning to extract and learn events in the news. Xie et al. [61] introduced a semantic treebased model to represent news data for predicting stock price movement. Luss et al. [62]

also exploited news in combination with return prices to predict intra-day price movements. They use the Multi Kernel Learning (MKL) algorithm for combining the two features. The combination shows improvement in final prediction in comparison to using each of the features alone. Motivated by this study, we investigate the performance of the MKL algorithm as one of the methods to combine the textual with non-textual information. Other data resources, such as stocks' message boards, are used by Nguyen and Shirai [63] to study topic modelling for aspect-based sentiment analysis. Wang and Hua [64]investigated the sentiment of the transcript of earning calls for volatility prediction using the Gaussian Copula regression model.

CHAPTER II

DETAILED PROBLEM DESCRIPTION

Recently, unsupervised pretraining of language models on large corpora has significantly improved the performance of many NLP tasks. The language models are pretained on generic corpora such as Wikipedia. However, sentiment analysis is a strongly domain dependent task. Financial sector has accumulated large scale of text of financial and business communications. Therefore, leveraging the success of unsupervised pretraining and large amount of financial text could potentially benefit wide range of financial applications. Here, we introduce our BERT, XLNet, and RoBERTa implementations for financial domain named as Fin-BERT, Fin-XLNet, Fin-RoBERTa respectively, after providing a background on the related neural network models.

2.1 Preliminaries

2.1.1 LSTM

Long short-term memory (LSTM) [65], as kind of recurrent neural network (RNN), uses update and forget gates to better model the persistence of long-term dependencies in a given sequential data. LSTM is one of the most common techniques to model a sequential data creation process, ranging from natural language to asset prices. As a text is composed of a token sequence, deciding the initial representation of an individual token is the initial target for any LSTM NLP model. Then, one usual practice is to use pretrained weights for representing the token initially. GLoVe (Global Vectors for Word Representation) [8] is one of such pretraining approaches. GLoVe model calculates word embeddings by training a regression model over word co-occurrence distribution on larger corpora. Even though GLoVe represents words effectively, its

embeddings are not contextualized in regards to the sequence they are part of. On the other hand, ELMo [12] represents the words as contextualized embeddings, where embedding of a word is influenced via its neighbouring words. ELMo mainly uses bidirectional LM having several LSTM layers. LM learns probability distribution on a consecutive series of tokens over provided vocabulary. ELMo focuses on modeling the probability of a token, given subsequent and preceding tokens in a token sequence. The model calculates a single contextualized embedding for each token, by learning a scheme to weight various embeddings over multiple LSTM layers. Once the model infers the contextualized embeddings, those embeddings may be utilized in initializing the downstream NLP tasks.

2.1.2 Transformers

Transformer uses an attention-based mechanism to model the knowledge in a sequential data [15]. Transformers are great alternatives to RNNs, and they have encoder-decoder architecture as a sequence-to-sequence model. The encoder in a transformer includes a number of identical layers. Every single layer consists of a fully-connected feed-forward neural network and multi-head self-attention layer. Transformer learns 3 mappings, value, query and key, from the embeddings over a single self-attention layer. It calculates a similarity score by multiplying every token's key vector with query vectors of all tokens. The model arrives at token's newer embedding via weighting the value vectors according to those similarity scores. In a multi-head self-attention mechanism, the model concatenates those layers all together, such that it could evaluate a token sequence according to different perspectives. Following such concatenation, the resulting vectors undergo fully-connected layers that share parameters. Here, we have focused on explaining solely the encoder part in more detail. However, the decoder part is extremely alike. Transformer has a number of advantages over recurrent neural network models [15]. For instance, RNNs cannot be easily parallelized over

GPUs due to their sequential nature. Additionally, information cannot persist due to excessive number of steps among distant items in a sequence.

2.1.3 BERT

BERT [14], as a language model, is composed of a number of Transformer encoders added over each other. But, its LM tasks are defined different than LSTM and ELMo. BERT does not focus on forecasting the following token given subsequent tokens. Instead, one task it focuses on is to mask a randomly-selected 15% of the whole set of tokens. These masked tokens are forecasted by using a softmax layer over the final encoder layer. Predicting the next sentence is the another task for BERT training. Given 2 sentences, BERT forecasts whether these sentences come after each other. It represents each input token sequence with position and token embeddings. [SEP] and [CLS] tokens are appended to the termination and start of the sequence respectively. It uses [CLS] token during all classification tasks. BERT comes up with 2 versions: 1- BERT-base version has 12 encoder layers with latent size being 768. It has 12 multi-head attention heads, and is defined over 110M trainable parameters; 2- BERT-large version has 24 encoder layers with latent size being 1024. It has 16 multi-head attention heads, and is defined over 340M trainable parameters in total.

2.1.4 XLNet

XLNet is a generalized auto-regressive language model which is developed considering the vulnerabilities of BERT-type models causing from degradation of data based on masking [17]. The aim is to get rid of such flaws by using probabilities of all permutations in the context. XLNet utilizes an permutation-based objective that combines the autoregression and autoencoding advantages. In summary, XLNet has two main advantages and enhancements: 1- It learns bidirectional context via optimizing the likelihood over the factorization sequence's whole set of permutations,

2- Its autoregression-based formula allows it to overcome BERT's limitations. Additionally, XLNet incorporates the Transformer-XL concepts into pretraining, which is an advanced autoregressive model. Their greatest model XLNet-Large uses the same architectural parameters as BERT-large, where their model sizes become the same. XLNet is found to achieve remarkable performance enhancement over BERT on multiple NLP tasks.

2.1.5 RoBERTa

RoBERTa is basically an adjusted version of BERT according to the study of Liu and her colleagues [16]. They found BERT to be remarkably undertrained, and modified the effect of multiple principal hyperparameters. The following levels of BERT model has been optimized: 1- They implement dynamic masking for the masked language modeling stage, 2- They eliminate the next sentence prediction objective and loss, 3- They train with large mini-batches leading to viable speed and accuracy, and 4- They use Byte-Pair Encoding instead of Unicode Encoding. RoBERTa is found to achieve a performance enhancement over BERT on multiple NLP tasks.

2.2 Fin-BERT: BERT for financial domain

Here, we define the enhanced BERT model for financial domain. We will focus on three major components: 1- How will additional pretraining on financial domain corpora be achieved by Fin-BERT, 2- How will Fin-BERT be implemented for text regression and classification tasks, 3- How will Fin-BERT block catastrophic forgetting by integrating additional training strategies as part of finetuning.

2.2.1 Additional Pretraining

According to [13], the final classification results can be enhanced by additional pretraining of a LM on target domain corpora. Similarly, [53, 54, 55] also show an enhanced performance for BERT across financial domain over classification results.

We carry out experiments with 2 techniques for additional pretraining. The first technique is to pretrain the network on a significantly larger corpora from the target task domain. In this technique, we pretrain a BERT LM over a financial corpus, where corpus details are defined in Section 3.1.1. The second technique is to pretrain the network solely over the training classification dataset sentences. Even though the second corpus is remarkably more compact, utilizing the dataset directly from the target may adapt better to target domain.

2.2.2 Fin-BERT for Text Classification and Regression

For sentiment classification, we have added a dense layer just after the final hidden layer's [CLS] token state. This practice is the recommended approach to use BERT in classification tasks [14]. Afterwards, we train the classifier neural network over labeled sentiment dataset. Even though this paper focuses on classification, we have also implemented regression by using almost similar framework over a distinct dataset having continuous target values. The only principal difference between classification and regression frameworks is that classification uses cross-entropy loss whereas regression uses mean squared error.

2.2.3 Preventing Catastrophic Forgetting

This fine-tuning method comes with the major risk of catastrophic forgetting as discussed previously by [13]. This major catastrophic forgetting risk is due to model rapidly forgetting the knowledge from LM task when it is being fine-tuned for the new task. We handle this catastrophic forgetting risk by applying 3 approaches as introduced by [13]: gradual unfreezing, discriminative fine-tuning, and slanted triangular learning rates.

Slanted triangular learning rate approach uses a non-uniform learning rate throughout training. Learning rate schedule in the form of slanted triangle is applied, where learning rate initially climbs up to a certain point and linearly drops afterwards. On

the other hand, discriminative fine-tuning uses smaller learning rates for neural network's lower layers and higher rates for network's shallow layers. Let the learning rate at l'th layer be α. Given θ discrimination rate, we use $\alpha_{l-1} = \theta\alpha_l$ as the learning rate for $l-1$'th layer. In discriminative fine-tuning, fine-tuning different layers non-uniformly makes sense since lower layers model the deeper language knowledge whereas the upper layers incorporate the knowledge for the actual classification task. Lastly, gradual freezing initially trains the network over all layers except the classifier layer which is frozen. Then, as part of the training, we slowly unfreeze all layers beginning from the highest layer, such that least fine-tuned features correspond to lower layer features. As a result, model is blocked to forget lower-level language knowledge throughout initial training stages, which has been learned over pretraining.

2.3 Fin-XLNet: XLNet for financial domain

Similar to the additional pretraining of Fin-BERT in Section 2.2, we carry out experiments with the same 2 techniques for additional pretraining. The first technique is to pretrain the network on a significantly larger corpora from the target task domain, whereas the second technique is to pretrain the network solely over the training classification dataset sentences. Similar to the Fin-BERT, we have added a dense layer just after the final hidden layer's [CLS] token state in terms of sentiment classification. The rest of the process is similar for both classification and regression. In this case, classification again utilizes cross-entropy loss whereas regression utilizes mean squared error. Lastly, Fin-XLNet may also suffer from the major risk of catastrophic forgetting as discussed previously. Similar to Fin-BERT, We handle this catastrophic forgetting risk by applying 3 approaches as introduced by [13]: gradual unfreezing, discriminative fine-tuning, and slanted triangular learning rates.

2.4 Fin-RoBERTa: RoBERTa for financial domain

Additional pretraining, application of Fin-RoBERTa to classification and regression tasks, and incorporating training strategies in Fin-RoBERTa to prevent catastrophic forgetting are similar to Fin-BERT as discussed in Section 2.2.

2.5 RiskBERT

The basis of transfer learning is pretraining, a method for improving results without starting from scratch with a new model. Pretraining involves first training the model using an extensive corpus and then fine-tuning the model layers on a small sample. After pre-training in an unsupervised manner on a significant amount of text data, the model can be quickly fine-tuned on a particular downstream task with a small number of labels because the general linguistic patterns have already been learned during pre-training. The purpose is to maximize performance in fields that use domain-specific languages, such as the economy and finance sectors. It gained popularity in 2018 thanks to ULMFiT, one of the first neural networks (based on LSTMs) to use transfer learning for natural language processing successfully [13]. In this study, our base model is BERT, for which we use the BERT-base version, having 12 layers, 768 hidden dimensions per token, and 12 attention heads with a total of 110 million parameters, to fine-tune with the financial data sample. To put it another way, the weights of the BERT-base model serve as a starting point for fine-tuning the BERT language model in the finance domain.

The first thing to do to fine-tune your base model is to have a corpus, the reports from SEC in 2005, which is a 1.7GB text file for this study. In the second step, the tokenizer belonging to the choice as base model is trained and a vocabulary file is extracted. Then, with the of the transformers library, our language model is trained using masked language modeling, a self-supervised pretraining strategy for learning text representations.

The trademark of BERT is that a different target is employed to train the language model rather than next-word prediction [66]. There are two parts to this goal. The model learns to anticipate randomly masked tokens from their context in the first section, which is called the masked language model objective. The model must determine if a sequence would logically follow the prior sequence in the second component, which is the next-sequence prediction procedure. The model can better capture long-term dependencies thanks to this strategy.

There are 12 layers with latent size 768, and 12 attention heads with a total of 110 million parameters for RiskBERT as well, since the BERT-base model is fine-tuned.

CHAPTER III

EXPERIMENTAL SETUP

3.1 Datasets

3.1.1 Corporate Filings 10-K & 10-Q

Corporate reports and filings are one of the most crucial textual datasets in business communication and finance. In US, all publicly trading companies are required to file annual reports, known as 10-K, and quarterly filings, known as Form 10-Q, mandated by the Securities Exchange Commission (SEC). 10-K and 10-Q filings carry out detailed information on the filing company's financial and business conditions. Companies cannot make misleading or false statements in these filings, as prohibited by regulation and law. Publicly available 10-K and 10-Q filings could be accessed from SEC website. Here, we obtain 60490 10-K and 142622 10-Q filings of Russell 3000 companies between 1994 and 2019 from SEC website. We focus on the subset of sections with textual parts in our analysis; Item 1 (Business) in 10-Ks, Item 1A (Risk Factors) in both 10-Ks and 10-Qs, and Item 7 (Managements Discussion and Analysis) in 10-Ks.

3.1.2 Financial PhraseBank

We mainly focus on Financial PhraseBank dataset for sentiment analysis tasks [44]. Financial PhraseBank dataset is composed of 4845 English sentences chosen randomly from a financial news database called LexisNexis. Each sentence is then labeled by people with finance or economics background. As part of such annotation, the annotating people have been requested to assign labels to sentences with respect to the possible effect of the sentence content in the corresponding company's stock price. In addition, the dataset incorporates knowledge about the agreement degrees over the

sentences between annotators. These sentences are grouped according to agreement degree of the annotators, such as 100%, $75-99\%$, $66-74\%$, $50-65\%$. Table 1 summarizes the dataset in terms of sentiment labels and agreement degrees. In our analysis, 20% of all sentences are kept as test set, and 20% of the remaining sentences are kept as validation set. As a result, train set in our analysis has 3101 examples. We also utilize 10-fold cross-validation for some of the experiments.

Table 1: Agreement degrees and sentiment tags on Financial PhraseBank dataset.

Agreement degree	Count	Neutral	Positive	Negative
100%	2262	61.4%	25.2%	13.4%
$75-99\%$	1191	63.6%	26.6%	9.8%
$66-74\%$	765	50.9%	36.7%	12.3%
$50-65\%$	627	54.5%	31.1%	14.4%
All	4845	59.4%	28.1%	12.4%

3.1.3 FiQA Dataset

FiQA dataset [67] contains a total of 1174 examples of news headlines and tweets, and it has specifically been generated for financial question answering and opinion mining challenges. Each example contains the sentence, the sentence snippet associated with the target financial entity, aspects, and sentiment score. A sample FiQA entry is shown in the table 2. Different than Financial PhraseBank, the sentiment score takes continuous value in the interval of -1 and 1 where 1 represent the most positive case. A Level 1 Aspect label takes on one of 4 possible labels (Corporate, Economy, Market or Stock), and Level 2 Aspect label takes on one of 27 possible labels (Appointment, Risks, Dividend Policy, Financial, Legal, Volatility, Coverage, Price Action, etc.). 10-fold cross-validation is performed for assessment of the models over this dataset.

3.1.4 AnalystTone Dataset

AnalystTone dataset [68] measures judgements and beliefs in analyst reports that is frequently employed in Finance and Accounting literature. The dataset consists of

Table 2: An example entry from FiQA

Sentence	easyJet expects resilient demand to withstand security fears.
Aspect Level 1	Corporate
Aspect Level 2	Risks
Sentiment Score	0.165
Target	easyJet

10000 randomly-selected sentences over analyst reports in Investext database. Each sentence is labeled into one of the 3 groups: neutral, positive, and negative. As a result, the dataset is not balanced as it contains 4590 neutral, 3580 positive, and 1830 negative sentences. 10-fold cross-validation is performed for assessment of the models over this dataset.

3.1.5 Dataset for RiskBERT

Utilizing the Securities Exchange Commission (SEC) annual reports named as "Form 10-K" which can be accessed through the website of SEC, the study is conducted. They are mandatory reports having the data about "the company history, organization, equity and subsidiaries and financial information" that published annually by all the "publicly-traded" institutions [31].

The data used in this research contains reports belong to year interval of 1996-2021 and the documents having Section 7, "management's discussion and analysis of financial conditions and results of operations" (MD&A), and Subsection 7A, "quantitative and qualitative disclosures about market risk" are used as input for the models as same as the study of Kogan and Levin [31] .

From the collection of Kogan and Levin [31], 30472 trimmed and ready-to-be-fed reports are obtained for the period of 1996-2006, while the reports published between 2007 and 2021 which are belong to companies in Russell 3000 index are got downloaded from the SEC website. Hence, for the 2007-2021 interval, 32050 reports are downloaded. After aggregating all reports, the ones who do not have a filing date or Sections 7 and 7A, are eliminated using simple search scripts for the texts.

The stock prices are obtained from the yfinance library of 'Yahoo! Finance' network. Using stock prices, volatility values of the company is calculated for both prior and posterior 12 month periods.

3.2 Preprocessing

Before feeding 10-K and 10-Q corpus for pretraining, each sentence is separated line by line and an empty line is inserted between each different document. In the sentiment analysis part, for the Financial PhraseBank and AnalystTone datasets, label "positive" is converted to 1, label "negative" is converted to -1 and label "neutral" is converted to 0. However, for FiQA datasets, each sentences is labeled according to the corresponding sentiment score. Sentiment scores greater than 0.33 are labeled as 1, those that are less than -0.33 are labeled as -1, and those in between are labeled as 0. We would expect that the model trained with FiQA headline will perform better than other models trained by Financial PhraseBank datasets for predicting FiQA post labels, and vice versa.

3.3 Baseline Methods

To better evaluate the performance of our domain-specific models Fin-BERT, Fin-XLNet, and Fin-RoBERTa, we consider the following 4 baselines:

3.3.1 LSTM Classifier

We implement 2 bidirectional LSTM classifiers: LSTM classifier with GLoVe [8] embeddings and LSTM classifier with ELMo [12] embeddings. Both models are same except one uses ELMo embeddings, whereas the other one uses GLoVe embeddings. In both classifiers, we use a hidden size of 128 for all states except the last one where the last hidden state dimension is 256 mainly because of bidirectionality. Then, the models have a fully-connected feed forward layer mapping the last latent state to a three dimensional vector which represents the three labels probabilities. We use a

learning rate of $3e^{-5}$ and 0.3 dropout probability across both models. Both models are trained till no improvement is observed for 10 epochs in the validation dataset loss. In the results section, we report the best result out of these 2 models under the name LSTM.

3.3.2 BERT

We use the default BERT parameters as discussed in [14]. BERT is trained with 256 batch size over a million steps. Adam optimizer is used with a learning rate of $1e^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay of 0.01 for $L2$. The learning rate decays linearly, and warmup is applied for first 10000 steps for learning rate warmup. 0.1 dropout probability is used for the whole set of layers. GELU activation is used instead of more traditional RELU. BERT training loss is the summation of two likelihoods: the average masked language model likelihood and the average following sentence forecasting likelihood.

3.3.3 XLNet

We use the default XLNet parameters as discussed in [17]. Their largest model XLNet-Large's architectural parameters are same as BERT-large, resulting in the same model dimensions. As part of the pretraining, the length of the used sequences is 512. Adam optimizer is run for half a million steps where learning rate decays linearly and batch size is 8192.

3.3.4 RoBERTa

We use the default RoBERTa parameters as discussed in [16]. RoBERTa is trained similar to BERT-large architecture.

3.4 RiskBERT Features

The research was conducted for the models employing the two base feature representations: word count-based features and bert-based embeddings, following the dataset

collection, relevant section extraction, and volatility calculation for the published dates.

Punctuation marks are removed from texts before being fed into the models using word count representation, and all letters are changed to lower case. All texts are also tokenized, and stop-words are removed. Fortunately, the sklearn feature extraction library can modify all of them. Utilizing sklearn library, tf (term frequency) and tf-idf (term frequency inverse document frequency) values are calculated. Term frequency is the number of times when the word appears in the document, as the name suggests. tf-idf is the multiplication of term frequency (tf) and inverse document frequency (idf). When n is the total number of documents, idf calculation is:

$$idf = log(\frac{(1+n)}{(1+documentfreq)}) + 1$$

By utilizing bidirectional training with *Transformer*, an attention model that essentially considers a text sequence as a combination of left-to-right and right-to-left training, BERT has greatly improved word vector representation. Encoder and decoder are the two major components of the transformer attention model. It applies masking to 15% of the words in each sequence before feeding them in, rather than making predictions based on the input sequence [15]. Based on non-masked input supplied through probability of occurrence with a softmax layer, it attempts to predict the masked words. Additionally, BERT includes a goal called "next sentence prediction" that simply determines whether each pair of sentences is consecutive or not. RiskBERT model used in this study is the bert-base model which is fine-tuned with financial corpus.

3.4.1 Machine Learning models tested

After feature extraction, three machine learning models are utilized to test the models: Support Vector Regression (SVR), xgboost and random forest. **SVR** is the support vector machine model using regression function for data classification problems for

continuous data [69]. A support vector machine technique seeks to locate an n-dimensional space hyperplane that classifies the data points. Support Vectors are the closest-to-the-hyperplane data points on either side of the hyperplane. These affect the hyperplane's position and orientation, influencing how the SVM is constructed. SVR is a supervised learning method for making predictions about discrete values. Finding the optimum fit line is the fundamental tenet of SVR. The hyperplane with the most points is the best-fitting line in SVR. The SVR model establishes a threshold error allowance that ensures all data points within the regression line are not penalized for their errors.

Random forests or random decision forests is an ensemble learning method which works by building a lot of decision trees during training. Decision trees produce a model that predicts the label by analyzing a tree of feature questions such as true/false or if/else and estimating the bare minimum of questions required to assess the probability of choosing the right choice. They might as well be used for regression tasks with continuous value. For classification and regression, another approach is Gradient Boosting Decision Trees (GBDT), which is a decision tree ensemble learning similar to **random forest** [70]. Ensemble learning algorithms mix different machine learning techniques to create a more accurate model. Both GBDT and random forest create a model of several decision trees. The way the trees are constructed and joined makes a difference.

From random bootstrap samples of the data set, Random Forest builds entire decision trees in parallel using a method called 'bagging.' All of the decision tree forecasts are averaged to provide the final prediction. The concept of "boosting" or strengthening a single weak model by combining it is where the term "gradient boosting" originates. As an extension of boosting, gradient boosting formalizes the process of additively creating weak models as a gradient descent method over an objective function. To reduce errors, gradient boosting establishes desired results for the

following model. The gradient of the error concerning the prediction determines the targeted outcomes for each case; therefore, the name "gradient boosting." A group of shallow decision trees is iteratively trained by GBDTs, with each iteration using the error residuals of the prior model to fit the new model. The weighted average of all the tree predictions represents the final projection.

Xgboost,Extreme Gradient Boosting, is a distributed, scalable gradient-boosted decision tree (GBDT) machine learning framework for regression, classification, and ranking issues offers parallel tree boosting [71].

The SVR and random forest models are imported from the sklearn library, whereas the xgboost model is imported from the XGBoost package. Trees are constructed using XGBoost in parallel as opposed to GBDT's sequential method. It employs a level-wise approach, scanning over gradient values and assessing the quality of splits at each potential split in the training set using these partial sums.

3.5 Performance Evaluation

3.5.1 Performance Evaluation for Sentiment Analysis

We evaluate the models as part of classification in terms of cross-entropy loss, macro F1 average, and accuracy. Accuracy is calculated as the number of correct predictions over the number of all predictions. Macro F1 average first estimates F1 scores for each class, and then averages them out, where $F1 = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$. In this case, precision is the ratio of True Positives (TP) to the predicted positive instances ($\frac{TP}{TP+FP}$), whereas recall is the proportion of True Positives to the actual positive instances ($\frac{TP}{TP+FN}$). F1 score is another successful metric to evaluate the classification performance, which is useful for Financial PhraseBank dataset where labels are not quite balanced (approximately 60% of the sentences are neutral). Cross-entropy loss is computed when adjusting model weights, aiming to minimize the loss. Cross-entropy loss is basically a measure of the difference between two probability

distributions for a given random variable or set of events. While evaluating the regression model, we use standard metrics such as R^2 and mean squared error. Among these evaluation metrics, higher F1, higher accuracy, and lower loss scores indicate better performance.

3.5.2 Volatility

Volatility could be described as the variability of stock price changes for a certain duration according to [72]. For the dataset used in this study, it is calculated as the standard deviation of stock return prices divided by the time period. Return values are calculated that stock price of the month divided by stock price of previous month and one subtracted form that value ($\frac{p_t}{p_t-1}$-1) as it is calculated in [31]. While the stock price having stable values during the time period there is low volatility value, fluctuating stock prices results in higher volatility [31] .

3.5.3 Performance Evaluation for Risk Prediction

Models are evaluated using mean squared error (MSE) scores for each year. MSE scores are calculated as the square of the difference between the log of volatility calculated after the document is released and the log of predicted volatility for each year and then divided by the number of documents.

$$MSE_{score} = \frac{1}{N} \times \sum_{n=1}^{N} (log(v_{post}) - log(v_{predicted}))^2$$

MSE score is used for evaluation because a stock's return variance typically changes gradually [31]. Other changes typically precede significant price swings, and stable periods are more likely to last.

3.6 Implementation Details

In Fin-BERT, we use the following metrics: Trained sequence is maximally made up of 64 tokens, warm-up ratio is 0.2, learning rate becomes $2e^{-5}$, and the dropout

probability is 0.1. The models are trained over 15 epochs, and the best models are selected over evaluation results on validation set. The discrimination ratio is used as 0.85 as part of discriminative finetuning. In Fin-XLNet and Fin-RoBERTa, we mainly use the default hyperparameters [17, 16]. Apart from these parameters, both models are again trained over 15 epochs, and the best models are selected over evaluation results on validation set. The discrimination ratio is used as 0.85 as part of discriminative finetuning.

We have used the transformers from Hugging Face library [20] for pretraining and fine-tuning purposes. We have extracted the results for pretrained models uploaded in the Hugging Face library. For BERT, BERT-base cased and BERT-large cased models; for RoBERTa, roberta-base and roberta-large; and for XLNet, XLNet-base cased and XLNet-large cased are used.

CHAPTER IV

RESULTS AND DISCUSSION

4.1 Finance-specific FinSentiment Models Outperform the General Models

Table 3 shows the results of finance-specific FinSentiment, Fin-BERT which is constructed by Araci in 2019 [53] and general baseline methods (LSTM, BERT, XLNet, and RoBERTa) that are state-of-the-art for sentiment classification task in Financial PhraseBank dataset over 10-fold cross-validation. In the last rows of the table 3, FinBert-base and FinBert-large models implemented by Liu and his study group [54]. Since only the accuracy and F1 scores are reported in their study, the related results are added separately. Accuracy and F1 scores that their models have are better than all other models including our FinSentiment. However, their models are pre-trained with huge financial corpora, composed of FinancialWeb (24GB), YahooFinance (19GB) and RedditFinanceQA (5GB) which in total 48GB-wide. Our FinSentiment models are pre-trained with just 1-year data of Corporate Fillings belong to Russell 3000 companies, which is around 5GB data, which is too small in comparison to FinBert models researched by Liu et al. [54].

In this case, our model FinSentiment in the table represents the best performance over Fin-BERT, Fin-XLNet, Fin-RoBERTa finance-specific models. Results are presented on both the all dataset and on its robust subset with 100% agreement. In terms of almost all of the evaluation criteria, FinSentiment clearly outperforms all the existing methods. LSTM with GLoVe approach uses static embeddings, and LSTM with GLoVe classifier cannot integrate model knowledge so it performs the worst. The performance of LSTM with GLoVE is close to the remaining models in

terms of accuracy, but its F1 score is lower. This is because LSTM with GLoVe embeddings performs reasonably better across neutral class. On the other hand, in terms of all evaluation criteria, LSTM with ELMo embeddings enhances the performance of LSTM with static GLoVe embeddings. LSTM with ELMo embeddings again has low mean F1 score mainly because of worse performance across sparsely distributed labels. However, its performance is relatively better than LSTM with GLoVe embeddings compared with BERT, XLNet, and RoBERTa in terms of both accuracy and F1 score. According to these results, the results from contextualized word embeddings are closer to ML-based approaches for dataset of this number of samples than the static embeddings. BERT, XLNet, and RoBERTa enhance the performance remarkably on all evaluation metrics, as well as models do not perform significantly better in certain classes than the remaining classes. According to this result, language model pretraining is effective in prediction performance enhancement for these general models. Among them, RoBERTa slightly outperforms the remaining two ML-based models BERT and XLNet. We would expect all of these larger general models (BERT, XLNet, RoBERTa) to overfit on this smaller Financial PhraseBank dataset. However, these models can solve this smaller data issue by incorporating efficient training strategies and LM pretraining. Lastly, finance domain-specific FinSentiment clearly outperforms all the existing methods across all metrics, where we report the best of Fin-BERT, Fin-XLNet, and Fin-RoBERTa for the corresponding FinSentiment values. Such outperformance of FinSentiment shows the effectiveness of finance-domain specific training on sentiment classification task.

In the table 3, we found FinSentiment to outperform all the remaining approaches across the whole set of considered metrics in Financial PhraseBank dataset. We also evaluate the performance of these approaches in Financial PhraseBank dataset across various number of labeled training set samples, including configurations for

Table 3: The best results are shown in bold for the corresponding evaluation metric in Financial PhraseBank dataset. We report 10-fold cross-validation results.

	All data			Data with 100% agreement		
Model	Loss	Accuracy	F1	Loss	Accuracy	F1
LSTM with GLoVe	0.802	0.714	0.641	0.572	0.817	0.745
LSTM with ELMo	0.741	0.756	0.706	0.504	0.843	0.777
BERT	0.391	0.827	0.816	0.122	0.932	0.943
XLNet	0.392	0.822	0.827	0.152	0.933	0.944
RoBERTa	0.372	0.835	0.834	0.133	0.948	0.966
Fin-BERT[53]	0.370	0.860	0.840	0.130	0.970	0.950
FinSentiment	**0.332**	**0.875**	**0.864**	**0.093**	**0.991**	**0.988**
FinBERT-base [54]		*0.91*	*0.89*			
FinBERT-large [54]		***0.94***	***0.93***			

100, 250, 500, 1000, 3000 samples. We found that training dataset of 100 samples is not enough for all models. Nevertheless, once we start using 250 samples in the training set, FinSentiment begins differentiating between labels efficiently. The performance of all approaches increases by increasing the number of samples in the training dataset. However, FinSentiment performs better with 250 samples than LSTM classifiers perform by using all samples in the training dataset. These results indicate the importance of language model pretraining.

Table 4 presents our results for FiQA financial sentiment dataset, where LSTM column denotes the best performance of LSTM with GLoVe and LSTM with ELMo embeddings and FinSentiment column reports the best of Fin-BERT, Fin-XLNet, and Fin-RoBERTa results as stated previously. FinSentiment clearly outperforms all the existing models for both MSE and R^2. It even outperforms the state-of-the-art models introduced in [1, 2]. All general models without a finance-specific training again outperform LSTM models with different embeddings, similar to Financial PhraseBank dataset. We use the reported results for state-of-the-art papers [1, 2], on which used test set is the official FiQA Task 1 test set. As we cannot access the full test set used in these papers, 10-fold cross-validation results are reported by us. From this perspective, our models are at disadvantage as we should keep a subset

of training set as a test set, whereas the state-of-the-art papers utilize the whole training set. LSTM is even outperformed by non domain-specific BERT, XLNet, and RoBERTa on both metrics, which also shows the importance of language model pre-training. Similar to Table 3 above for classification task, FinSentiment outperforms their non-finance specific counterparts, which shows the domain specific training on regression tasks.

Table 4: The best results are shown in bold for the corresponding evaluation metric. The results reported by [1, 2] are over the official test. Such original dataset is not available to us, so we report MSE and R^2 results with 10-fold cross-validation.

Metric	LSTM	[1]	[2]	BERT	XLNet	RoBERTa	FinSentiment
MSE	0.13	0.08	0.09	0.07	0.07	0.07	0.06
R^2	0.35	0.40	0.41	0.52	0.51	0.55	0.59

Table 5 analyzes the performance of the contributing finance-specific models in FinSentiment (Fin-BERT, Fin-XLNet, Fin-RoBERTa) with respect to their general counterparts in detail. Additionally, FinBERT-FinVocab uncased model from the study of Yang and his colleagues [55] is added to the table 5. It has the best performance value in their paper [55]. They construct a vocabulary set called FinVocab on financial corpora collected from 10K& 10Q Corporate reports and BERT model is trained from scratch with this vocabulary for 1M iterations [55]. Although, it outperformed our Fin-BERT model, yet it is not better than our Fin-RoBERTa. In general, finance-specific models, especially Fin-RoBERTa, substantially outperform the general counterparts of the corresponding finance-specific models in terms of accuracy on all 3 financial sentiment analysis tasks; PhraseBank, FiQA, AnalystTone. The accuracies of all approaches appear to be higher in AnalystTone dataset. Overall, the difference between finance-specific models and their counterparts appear to be quite remarkable for FiQA dataset. We observe similar results in terms of F1 score as well. These results again show the importance of domain-specific training across all 3 different datasets and tasks.

Table 5: The performance of general (BERT, XLNet, RoBERTa) and finance-specific models (Fin-BERT, Fin-XLNet, Fin-RoBERTa) in terms of accuracy on 3 financial sentiment analysis tasks. The best result for each dataset is shown in bold.

Dataset	BERT	Fin-BERT	XLNet	Fin-XLNet	RoBERTa	Fin-RoBERTa	FinBERT-FinVocab[55]
PhraseBank	0.827	0.862	0.822	0.869	0.835	**0.875**	0.872
FiQA	0.653	0.844	0.662	0.846	0.692	**0.882**	0.844
AnalystTone	0.840	0.887	0.845	**0.901**	0.843	0.897	0.887

Table 6 analyzes the performance of the contributing finance-specific models in FinSentiment (Fin-BERT, Fin-XLNet, Fin-RoBERTa) with respect to their general counterparts in detail in terms of accuracy where the train and tests are possibly from different cross datasets. In the table, *General* represents the already available general models and *Financial* represents the finance-specific models that are fine-tuned by using 10-K corpus. Among the datasets in the table, *allAgreeFPD* denotes the portion of Financial PhraseBank dataset where annotators fully agree on their labeling, whereas *50AgreeFPD* denotes the portion of the same dataset with between $50-65\%$ agreement levels. Similary, *FiQAHeadline* dataset contains only news headlines in FiQA datasets, whereas *FiQAPost* dataset includes the subset of FiQA dataset just with tweets. Even when we train the models by a dataset and test its performance on a different dataset, finance-specific models outperform the general counterparts. The improvement percentage differs depending on the model type and the datasets. For example, when the training dataset is *allAgreeFPD* from Financial Phrase Databank (FPD), unsurprisingly, *allAgreeFPD* perform better when they are tested with dataset again from FPD. Likewise, we have found *FiQAHeadline* and *FiQAPost* to outperform when they are tested with a test set from FiQA in comparison to the test set from FPD. On the contrary, several cases with training set as *FiQAHeadline* or *FiQAPost* show better performance when testing with a set from FPD, even when the test set is extremely different than the training dataset. According to these accuracy results, subject to the dataset, RoBERTa typically outperforms the remaining XLNet and BERT models. General outperformance of RoBERTa is also observed in terms

of F1 score across different training and test dataset combinations.

Table 6: Accuracy results of financially fine-tuned and general models. The dataset is splitted with 0.3 rate across the scenarios with $*$.

Train Set	Test Set	Pre-training	Fin-BERT / BERT	Fin-RoBERTa / RoBERTa	Fin-XLNet / XLNet
FiQAPost	FiQAPost$*$	Financial	0.502	0.684	0.635
		General	0.502	0.644	0.610
FiQAPost	allAgreeFPD	Financial	0.542	0.720	0.637
		General	0.392	0.723	0.621
FiQAHeadline	FiQAHeadline$*$	Financial	0.613	0.702	0.557
		General	0.664	0.552	0.536
FiQAHeadline	allAgreeFPD	Financial	0.650	0.808	0.639
		General	0.539	0.777	0.614
allAgreeFPD	FiQAPost	Financial	0.432	0.545	0.484
		General	0.400	0.543	0.460
allAgreeFPD	allAgreeFPD$*$	Financial	0.985	0.991	0.978
		General	0.972	0.988	0.972
allAgreeFPD	50AgreeFPD	Financial	0.867	0.875	0.872
		General	0.863	0.878	0.873

4.2 The Impact of Additional Pretraining Type

We evaluate the impact of additional pretraining type on the classifier's performance as seen in Table 7. Table 7 shows the results only for BERT variants. However, results are similar for XLNet and RoBERTa as well. We evaluate 3 models: 1- Vanilla BERT: Plain BERT model without any additional pretraining; 2-Fin-BERT-task: Fin-BERT model with additional pretraining over the classification train set; 3- Fin-BERT-domain: Fin-BERT model with additional pretraining over domain corpus, 10Ks and 10Qs. Table 7 shows the results, where we have evaluated the models with respect to macro average F1 score, accuracy, and loss over the test set. The best result for each metric is shown in bold in the table. The classifier which has been additionally

pretrained on the financial domain corpus outperforms the remaining models, even though their relative performance difference is not significant. This result can be mainly explained by the fact that the task set's distribution may be different than the distribution of the corpus.

Table 7: The best result for each metric is shown in bold. Reported results are obtained with 10-fold cross-validation.

Model	Loss	Accuracy	F1
Vanilla BERT	0.39	0.83	0.82
Fin-BERT-task	0.35	0.86	**0.87**
Fin-BERT-domain	0.34	**0.87**	0.86

4.3 Catastrophic Forgetting Prevention

Table 8: Catastrophic forgetting results on BERT and RoBERTa models. The best result for each column is shown in bold. The reported results are obtained with 10-fold cross-validation. STLR: Slanted Triangular Learning Rates, GU: Gradual Unfreezing, DFT: Discriminative Fine-tuning.

	Fin-BERT			Fin-RoBERTa		
Strategy	Loss	Accuracy	F1	Loss	Accuracy	F1
None	0.474	0.831	0.833	0.455	0.822	0.842
STLR	0.392	0.812	0.813	0.387	0.819	0.831
STLR + GU	0.377	0.858	**0.875**	0.356	**0.891**	**0.875**
STLR + DFT	0.425	0.802	0.799	0.396	0.844	0.807
All three	**0.341**	**0.862**	0.856	**0.332**	0.875	0.864

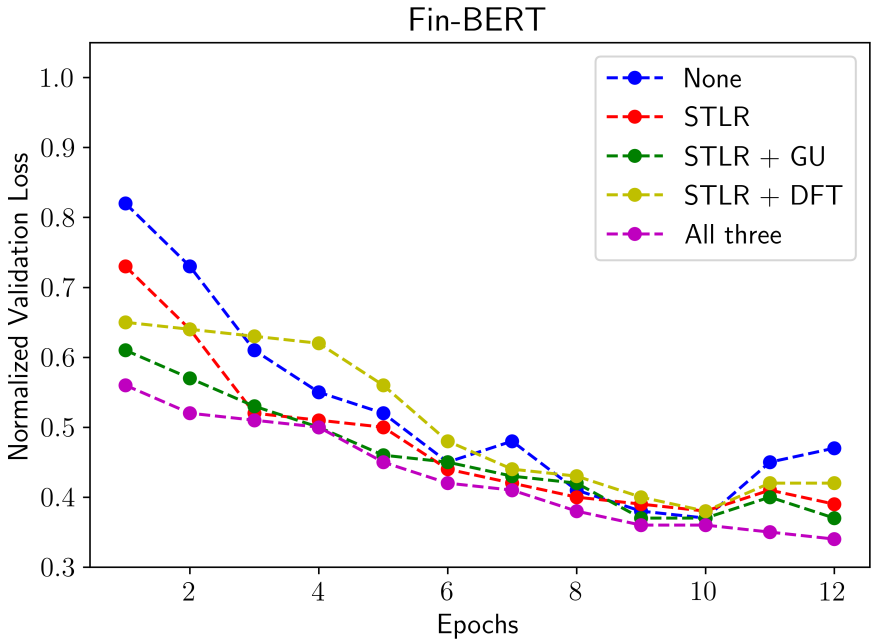
We assess the performance of our approaches against catastrophic forgetting under 5 different settings: 1- No adjustment (None) where all layers have the same stable learning rate, 2- Only with slanted triangular learning rate (STLR), 3- Slanted triangular learning rate and gradual unfreezing (STLR+GU), 4- Slanted triangular learning rate and discriminative fine-tuning (STLR + DFT), 5- All three techniques (All three at the same time). Among these strategies, gradual unfreezing is the strategy in which model layers are unfrozen gradually, and for the discriminative fine-tuning,

the applied learning rate is increased from first to last layers. Hence, the logic behind these two strategies are alike, i.e. for those two techniques we would expect resembling behaviour in the performance of the model. The performance of those 5 settings are reported in terms of accuracy, loss on test set, and F1 score for both Fin-BERT, and Fin-RoBERTa as seen in Table 8. We do not report the results for Fin-XLNet, but its results are similar to BERT. For Fin-BERT, we observe the best performance with regard to accuracy and test loss when all 3 strategies are applied. The main motivation behind discriminative fine-tuning and gradual unfreezing is the same: Lower-level attributes must be fine-tuned less than higher-level attributes, as the knowledge extracted by language modeling exists largely in the lower levels. As seen in Table 8, using solely STLR performs better than using STLR + DFT, so gradual unfreezing is the most principal strategy in our case. For Fin-RoBERTa, using slanted triangular learning rate and gradual unfreezing to prevent catastrophic forgetting performs better than using all three strategies at the same time, even though we have reported the performance for the case we use all three strategies at the same time in previous sections. Therefore, RoBERTa results in previous sections can be further enhanced by adapting better strategies to prevent catastrophic forgetting.

Unanticipated jump in validation loss after a number of epochs can be one of the observations for existence of catastrophic forgetting. If no strategy for catastrophic forgetting is applied, as we train the model, it rapidly begins overfitting. According to Figure 1, we observe such overfitting when none of the proposed strategies have been applied. In this case, after achieving the lowest loss on the initial epoch, the model begins and overfitting and the performance deteriorates. In other extreme case, when all the 3 techniques are carried out, the model will have much greater stability. The remaining strategy mixtures will take place between these 2 extreme scenarios. Figure 1 shows the validation loss with respect to epochs only for Fin-BERT, but plots are similar for Fin-RoBERTa and Fin-XLNet as well. For the overall process, as

expected, STLR + GU, STLR + DFT, and All three strategies applied models behave similarly, i.e. their loss values are extremely closer and their validation loss change plots are nearly parallel. None and only STLR strategies reach the best performance in the initial epochs, and after that, they overfit. However, the remaining strategies perform better and better by increasing epochs.

Figure 1: Normalized validation loss graph for Fin-BERT model across various strategies to prevent catastrophic forgetting.



4.4 Observing the Impact of Layers

Fin-BERT and Fin-RoBERTa base models have 12 encoder layers, while the Fin-XLNet base model has 12 transformer layers. The last layers in these models do not always capture the most appropriate knowledge on the classification task as part of language model training. So, we have applied sentiment analysis to each layer independently in order to see the contribution of each layer in the classification process and results. In our analysis, the classification layer is placed after the respective embedding's [CLS] tokens in BERT.

In Table 9, we show classification results from different layers over Fin-BERT and Fin-XLNet. Results for Fin-RoBERTa are similar to Fin-BERT. There is a sinus wave-like pattern for the results by increasing layers for Fin-BERT. The best performance is obtained in the Layer 7 with 0.921 accuracy which is 0.111 greater than the lowest accuracy score among the layers, which belongs to Layer 1. However, the performance discrepancy of the last layers are not huge, as it can be observed in the Table 9. For Fin-XLNet, there is also a sinus wave-like pattern for the results by increasing layers with zig-zags on the latest layers. Among the transformer layers in Fin-XLNet, Layer 6 has the best performance results, with the performance variation being in a small interval for the layers between 4 and 8. For all these finance-specific models, the highest scores are observed between the layers 4 and 8. Additionally, in this interval of layers, the performance does not make a significant difference. According to the table, almost middle layers, 6th and 7th layers, exhibit the best model performance with regard to all considered metrics for Fin-BERT and Fin-XLNet respectively. Such result may be due to 2 factors: 1- The trained model becomes larger, so remarkably capable when the higher layers are utilized for classification task, 2- The low layers in the model represent deeper semantic knowledge, so those layers cannot easily fine-tune such knowledge for classification task.

4.5 The Impact of Training the Model Over a Subset of Layers

Fine-tuning the whole set of models demands remarkable computational power and running time even for smaller datasets since all general and finance-specific models are enormous. Across several scenarios, a bit lower performance can be preferred if we can achieve such lower performance by fine-tuning smaller subset of the whole set of parameters. Such modification in fine-tuning may make our FinSentiment models more appropriate to use, principally for larger training datasets. We fine-tune solely the final k many encoder layers as part of our experiments. Fine-tuning solely

Table 9: Accuracy, F1 score, and loss results for each encoder/transformer layer of Fin-BERT and Fin-XLNet models respectively when used for classification.

	Fin-BERT			Fin-XLNet		
Layer	Accuracy	F1	Loss	Accuracy	F1	Loss
Layer 1	0.810	0.667	0.525	0.809	0.735	0.488
Layer 2	0.848	0.786	0.397	0.862	0.837	0.374
Layer 3	0.880	0.849	0.334	0.867	0.847	0.374
Layer 4	0.876	0.842	0.341	0.901	0.887	0.281
Layer 5	0.897	0.867	0.303	0.913	0.903	0.273
Layer 6	0.918	0.896	0.256	**0.925**	**0.912**	**0.250**
Layer 7	**0.921**	**0.900**	**0.250**	0.924	0.909	0.262
Layer 8	0.919	0.896	0.286	0.912	0.894	0.259
Layer 9	0.901	0.903	0.307	0.921	0.909	0.283
Layer 10	0.885	0.895	0.315	0.872	0.841	0.336
Layer 11	0.868	0.883	0.324	0.787	0.578	0.470
Layer 12	0.862	0.864	0.334	0.869	0.838	0.374

the classification layer does not perform as good as fine-tuning the remaining layers. Though, clever fine-tuning of only the final layer performs better than most of the state-of-the-art machine learning methods. We observe the almost same performance after Layer 9, which performance is surpassed via fine-tuning the complete model. We find that a longer training of the complete model is not necessary in order to fully use FinSentiment models. In this case, a trade-off exists between shorter running time for training and tiny performance drop in the model.

4.6 Analysis of Predictions

We examine the forecasts of both general and finance-specific models by using Financial PhraseBank dataset's 100% annotator agreement part as the test dataset. Such dataset consists of 2264 sentences for which 303, 570, 1391 of them are labeled as negative, positive, and neutral respectively. Both general and finance-specific model exhibit a very high accuracy over Financial PhraseBank dataset's 100% annotator agreement subpart. So, examining the scenarios where models fail to forecast the

true label will be quite interesting. In this section, we introduce a number of sentences where models make the incorrect predictions.

For Fin-BERT model, half of the wrongly predicted sentences are originally neutral and among the rest of the sentences, there is an approximately equal distribution between positive and negative sentences. While negative sentences are mostly predicted as positive (80%), positive sentences are predicted neutral with 88%, and neutral sentences are predicted positive with 91%. In other words, Fin-BERT model is inclined to entangle positive with neutral when it fails to predict correctly. Moreover, for Fin-RoBERTA model, 42% of the incorrect predictions have the true label positive. The second place for the model unsuccessful prediction belongs to negative sentences with 33%. In addition, negative sentences are perceived mostly as neutral (96%), as well as positive sentences are mistaken as neutral with 92%. Almost all neutral sentences among the incorrect predictions are predicted as positive (98%). Our Fin-RoBERTa model might confound negative and positive sentences as neutral, plus, it may predict neutral sentences with positive labels. Lastly, for our Fin-XLNet model, incorrect predictions are in general originally negative sentences, i.e. 47% of the incorrect predictions are originally negative. Neutral and positive sentences are predicted as incorrect almost equal likely, 27% and 24% respectively. Also, negative sentences are predicted positive with 78%. Furthermore, nearly the whole set of the neutral sentences is predicted positive and vice versa. As a conclusion, Fin-XLNet tends to predict positive.

Tables 10-12 show a number of examples that are incorrect predictions of the sentences whose true label is negative, neutral, and positive respectively as annotated by the experts in finance. As mentioned above in terms of sentences which true label is negative, Fin-XLNet tend to predict negative sentences incorrectly as positive. When Fin-RoBERTa fails to predict negative sentences, it is inclined to predict them as neutral in comparison to the general RoBERTa model which is not financially

fine-tuned and might predict negative sentences as positive. In other words, Fin-RoBERTa model stays closer to negative in the sentiment range, when testing with the negative sentence prediction. In terms of sentences which true label is neutral, financially fine-tuned Fin-XLNet, Fin-RoBERTa, and Fin-BERT models mostly predict neutral sentences as positive in their unsuccessful predictions. Lastly, in terms of sentences which true label is positive, all financially-tuned models mostly predict neutral sentences as positive in their unsuccessful predictions. We would expect this behaviour for these models, since on the sentiment interval, neutral sentiment is closer to positive sentiment in comparison to negative.

Table 10: The sample of sentences with wrong predictions whose true label is negative.

Text	Fine-tuning	Fin-BERT / BERT	Fin-RoBERTa / RoBERTa	Fin-XLNet / XLNet
Profit before taxes amounted to EUR 56.5 mn, down from EUR 232.9 mn a year ago.	Financial	Positive	Neutral	Positive
	General	Neutral	Neutral	Positive
Salonen added that data shows producers' pulp inventories in North America are declining.	Financial	Neutral	Neutral	Neutral
	General	Neutral	Neutral	Neutral

4.7 Bert-based Embeddings vs Word Count Features for Risk Prediction

Our main aim in these experiments to observe the efficiency of financially-tuned BERT model in risk-prediction.

In the experiments, we tested BERT model embeddings, RiskBERT embeddings and as a baseline tf and tf-idf features of documents in the dataset with or without using previous volatility values of each year.

Table 11: The sample of sentences with wrong predictions whose true label is neutral.

Text	Fine-tuning	Fin-BERT / BERT	Fin-RoBERTa / RoBERTa	Fin-XLNet / XLNet
Ahlstrom's share is quoted on the NASDAQ OMX Helsinki.	Financial	Positive	Positive	Positive
	General	Positive	Positive	Positive
Kemira shares closed at (x20ac) 16.66 ($22US.71).	Financial	Positive	Positive	Positive
	General	Negative	Negative	Positive

Table 12: The sample of sentences with wrong predictions whose true label is positive.

Text	Fine-tuning	Fin-BERT / BERT	Fin-RoBERTa / RoBERTa	Fin-XLNet / XLNet
The agreement strengthens our long-term partnership with Nokia Siemens Networks.	Financial	Neutral	Neutral	Neutral
	General	Neutral	Neutral	Neutral
Production capacity will increase from 36000 to 85000 tonnes per year and the raw material will continue to be recycled paper and board.	Financial	Neutral	Neutral	Neutral
	General	Neutral	Neutral	Neutral

4.7.1 Features

Word Count Representation

For the baseline case, tf and tf-idf values of each word are calculated and fed into the SVR model like in [31]. They have tested the effect of the word feature representations for the same dataset.

BERT-based Embeddings

The features of our elemental model, RiskBERT, were extracted from a financially

fine-tuned BERT-base model. Two embeddings-vector from extracted embeddings from all documents for each year: one is the mean of embedding vectors of words in texts, and the other one is the standard deviation vector of embeddings. These two vectors are appended. Hence, in the end, we got a vector with a 1x1536 dimension. To compare the effect of volatility values of the previous years, another feature set, the historical data, is added to both word feature representations and bert-based model embeddings. Features used in this work can be seen in the tables 13, 14, 15 and , 16.

For computational reasons, word count features are fed into the SVR model using the 2000 best features. BERT-base and RiskBERT embeddings are tested with xgboost, random forest, and SVR models with and without using previous years' volatility values.

4.7.1.1 Comparison of Features

In tables 13, 14, 15 and , 16 the predictions are computed using the data belong to previous 5 years for each year. The same procedure is applied by using one-year and 2-year periods. For the time before 2007, it can be seen in the tables 13 and 17. For the cases in which volatility values of previous years are not included, word count representations performed better, especially those with tf-idf values. The closest model to them is the model using SVR with BERT or RiskBERT embeddings. Using previous year volatility as a feature improves the performance of all cases. However, adding the previous year's volatility values changes the course of action. Appending the historical data to the feature set increases the performance of the models above the tf and tf-idf-based models.

Investigating the years before 2007 in detail, for the results of bert-based embeddings, there is an increase in mean squared error from 2001 to 2003, and a decrease

Table 13: MSE results of features for 5 year data without using previous year volatility

features	2001	2002	2003	2004	2005	2006	micro-average
tf-idf (without prev)	0.2154	0.2300	0.1979	0.1513	0.1596	0.1616	**0.1834**
tf (without prev)	0.2157	0.2299	0.2012	0.1523	0.1609	0.1622	0.1845
xgboost-RiskBERT	0.2684	0.2978	0.3265	0.2712	0.2418	0.2244	0.2718
random forest-RiskBERT	0.2733	0.3069	0.3373	0.2698	0.2404	0.2199	0.2745
svr-RiskBERT	0.2300	0.2530	0.2496	0.1916	0.1682	0.1823	0.2108
xgboost-bert	0.2713	0.3057	0.3394	0.2753	0.2394	0.2285	0.2768
random forest-bert	0.2715	0.3126	0.3424	0.2759	0.2439	0.2277	0.2791
svr-bert	0.2321	0.2525	0.2535	0.1921	0.1702	0.1850	0.2126

Table 14: MSE results of features for 5 year data without using previous year volatility after 2006

features	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	micro-average
tf-idf (without prev)	1.7315	0.8580	0.5381	0.2982	0.3003	0.4413	0.2958	0.3080	0.3573	0.3177	0.3297	0.3471	0.3472	0.3940	0.3372	0.4440
tf (without prev)	1.8683	0.8570	0.5295	0.2878	0.2974	0.4439	0.2958	0.3091	0.3576	0.3182	0.3297	0.3460	0.3433	0.3935	0.3358	0.4490
xgboost-RiskBERT	2.4336	0.4964	0.3273	0.2634	0.2298	0.3312	0.2403	0.2092	0.2393	0.2197	0.2135	0.2209	0.3297	0.3329	0.2551	0.3781
random forest-RiskBERT	3.3808	0.4812	0.3156	0.2425	0.2118	0.3320	0.2399	0.2011	0.2255	0.2050	0.2030	0.2053	0.3118	0.3273	0.2350	0.4113
svr-RiskBERT	2.3048	0.4813	0.3468	0.1976	0.1876	0.2672	0.1884	0.1790	0.2108	0.1955	0.1817	0.2157	0.2433	0.2716	0.2396	0.3374
xgboost-bert	2.2450	0.5602	0.3625	0.2924	0.2763	0.3881	0.2916	0.2877	0.3357	0.3001	0.3074	0.3137	0.3807	0.3901	0.3039	0.4299
random forest-bert	2.1048	0.5091	0.3589	0.2943	0.2804	0.4070	0.3008	0.2866	0.3301	0.3018	0.3095	0.2988	0.3708	0.3980	0.3061	0.4211
svr-bert	2.1759	0.6129	0.4294	0.2691	0.2907	0.4041	0.2836	0.2915	0.3508	0.3090	0.3095	0.3477	0.3406	0.3802	0.3261	0.4356

in the period between 2003 and 2006, as the recent years approached, apart from the support vector regression (SVR) used model. The same pattern appeared for the SVR model, but there is an increase in error from the year 2005 to 2006, although the error is close. In the tables 15 and 13, we can see that with and without using previous volatility data for the tf and tf-idf models, the error goes up for the period between the years 2001 to 2002. Moreover, while the error declined from 2002 to 2004, there was a tiny increase in the error from 2004 to 2006. In the table 15, for the results of bert-based embeddings using the historical data, the models using xgboost and random forest have similar patterns of error difference inclination. There is a descent in error from 2001 to 2002 and from 2003 to 2004. However, the error is ascending for

the years from 2002 to 2003 and 2004 to 2006. The SVR model has increasing error for the periods from 2001 to 2002 and 2005 to 2006. The error is decreasing from the year 2002 to 2005.

In general, there is a visible increase in the performance of models from 2003 until 2007 with or without historical data, which might be a result of the Sarbanes-Oxley Act passed in 2002. The Sarbanes-Oxley Act is federal legislation in the United States that requires businesses to follow particular procedures for financial record keeping, and reporting [73]. According to the act, the annual report must state that "management is responsible for developing and maintaining an effective internal control structure and procedures for financial reporting." Additionally, the company filing must "contain an assessment of the effectiveness of the issuer's internal control structure and processes for financial reporting as of the end of the Company's most recent fiscal year." In fact, this is an implementation that affects the report sections used in this research and is a probable cause for the extension of the reports published.

Table 15: MSE results of features for 5 year data with using previous year volatility

features	2001	2002	2003	2004	2005	2006	micro-average
tf-idf (with prev)	0.2051	0.2129	0.1794	0.1398	0.1524	0.1548	0.1715
tf (with prev)	0.2049	0.2140	0.1860	0.1428	0.1557	0.1562	0.1742
xgboost(with prev)-RiskBERT	0.1908	0.1611	0.1658	0.1183	0.1277	0.1419	**0.1488**
random forest(with prev)-RiskBERT	0.1914	0.1662	0.1712	0.1241	0.1291	0.1428	0.1522
svr(with prev)-RiskBERT	0.1988	0.2060	0.2023	0.1351	0.1301	0.1434	0.1671
xgboost(with prev)-bert	0.1949	0.1668	0.1578	0.1199	0.1309	0.1439	0.1490
random forest(with prev)-bert	0.1941	0.1658	0.1627	0.1262	0.1303	0.1469	0.1521
svr(with prev)-bert	0.2152	0.2125	0.1990	0.1343	0.1304	0.1483	0.1705

Tables 14 and 16 show that there is a singularity in the year 2007, in which all MSE scores are way higher than in the rest of the years. For the results of 2007, using tf and tf-idf values and discarding historical data contributed to the performance.

Table 16: MSE results of features for 5 year data with using previous year volatility after 2006

features	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	micro-average
tf-idf (with prev)	1.7361	0.8564	0.5377	0.2969	0.2993	0.4416	0.2953	0.3073	0.3564	0.2333	0.2449	0.2858	0.2640	0.3308	0.3631	0.4171
tf (with prev)	1.8747	0.8555	0.5292	0.2867	0.2964	0.4443	0.2952	0.3083	0.3566	0.2326	0.2442	0.2824	0.2626	0.3304	0.3589	0.4218
xgboost-RiskBERT	2.8846	0.3543	0.4097	0.2317	0.1555	0.2429	0.1436	0.1460	0.1789	0.1525	0.1596	0.1878	0.2266	0.2565	0.3387	0.3541
random forest-RiskBERT	1.5592	0.3776	0.4305	0.2381	0.1602	0.2546	0.1463	0.1441	0.1801	0.1504	0.1638	0.1851	0.2375	0.2624	0.3228	0.2945
svr-RiskBERT	2.2814	0.4226	0.3886	0.1633	0.1590	0.2470	0.1704	0.1549	0.1862	0.1692	0.1653	0.2065	0.2027	0.2463	0.2765	0.3210
xgboost-bert	0.5303	0.3665	0.4314	0.2226	0.1659	0.2496	0.1513	0.1542	0.1909	0.1601	0.1764	0.2040	0.2140	0.2628	0.3654	0.2511
random forest-bert	0.5160	0.3630	0.4517	0.2372	0.1845	0.2703	0.1524	0.1544	0.1965	0.1605	0.1799	0.1999	0.2179	0.2722	0.3661	0.2560
svr-bert	1.7498	0.4236	0.5028	0.1790	0.2136	0.3068	0.2020	0.1929	0.2226	0.1961	0.2053	0.2687	0.2207	0.2920	0.3418	0.3376

Nevertheless, for these models, the insertion of volatility values belonging to the preceding year did not have the same effect in the other years except 2021. In other words, after 2007, there was outperformance for the results of models using tf and tf-idf features with the addition of previous volatility data. Furthermore, even if adding data from previous years enhanced the results for the models using RiskBERT embeddings, they underperformed the word count-based models in 2007 except for the random forest model, for which the MSE score is closer yet. Unexpectedly, the results of 2007 belong to the models using BERT embeddings with the addition of historical data, performed by far the best amongst others. As a matter of fact, in 2007, xgboost and random forest models using BERT embeddings had the closest MSE scores to other years in which the results acted more expectedly. The irregularity of the results in 2007 instantly suggests the Global Financial Crisis or Great Recession exploded starting in 2007. All models obtained worse results in comparison to other years, meaning the reports on the financial performance of the companies might not have foreseen the incoming financial crisis. Moreover, the only year that RiskBERT embeddings failed to have obviously better results than BERT embeddings was 2007. This could mean that non-financial language modeling interpreted risk analysis better in 2007.

After 2007, the models using word count features improved with the insertion of historical data except for 2021. The annual change in MSE score behavior is similar

with or without adding the volatility value of the preceding year. Their performance improved during the 2007-2010 interval. Despite an insignificant increase in the MSE score from 2010 to 2011, the increase in the MSE score is vast from 2011 to 2012, as well as the decrease from 2012 to 2013. Once again, there was a tiny increase in MSE scores from 2013 to 2014 and a significant increase in error from 2014 to 2015. Although there is a decline in the error from 2015 to 2016, the decrease is more effective when using historical data. From 2016 to 2020, the performance of the word count-based models gradually worsened. Then, from 2020 to 2021, the performance worked up without adding the volatility value of previous years, whereas adding the historical data caused an underperformance.

For the models fed with bert-based embeddings, considering data of previous years made the results better except for the year 2021. The alternation in MSE scores is similar whether using the past data or not for the period 2007-2020. There was a rise in the performance of the models between 2007 and 2011. Then, there was a dramatic deterioration in the performance from transition to 2012 and a similar change in the opposite way from 2012 to 2014. The error increases for the duration from 2014 to 2015 and drops back to a similar value in 2016. There are differences in the pattern of MSE scores between using past data and not, after 2016.

For the models without the addition of past records, between 2016 and 2018, there is an inconsiderable change for the ones using xgboost or random forest. However, for the models using SVR, the model with RiskBERT embeddings performed better for the year 2017 than in 2016, then worsened from 2017 to 2018. For the SVR model with BERT embedding, there is no change between 2016 to 2017, while it underperformed in 2018. The results have worsened from 2018 to 2020, and the performances have improved from 2020 to 2021.

The models using bert-based embeddings with the addition of the volatility values of previous years the performance decreased from 2016 to 2021, especially from 2020

to 2021 with a dramatic change. However, there is a tiny exception for the SVR using bert embeddings. There is a little drop in the error from 2018 to 2019, and it goes back up from 2019 to 2020.

Although the financial crisis exploded in 2007, until 2010, or for bert-based models until 2011, there was an upturn in the performance of models in general. Following the Great Recession, the Dodd–Frank Wall Street Reform and Consumer Protection Act was passed in 2010, which modernized financial regulation and made reforms that were felt throughout practically all federal financial regulatory agencies and the country's financial services sector. The section of the act mentioning reporting is the section 954, which addresses "clawback of compensation practices," which are designed to prevent CEOs from profiting from false financial reporting. According to these regulations, in the event of an accounting restatement brought on by a failure to comply with reporting standards, executives must repay any improperly awarded compensation as described in section 953 regarding pay for performance.Any pay paid to current or past executives connected to the company during the three years prior to the restatement must be recouped by the company if an accounting restatement is made [74]. The enactment of the Dodd-Frank might affect the sharp decline in MSE scores after 2012.

The only year in which the data belonging to previous year have not improved the performance of all models is 2021. Without the insertion of historical data to the models they had fair performances, for which the MSE score values are not divergent as in the 2007 with or without addition of previous years' data. This is the explicit impact of COVID crisis that plucked the connection of historical data.

4.7.1.2 Comparison of BERT and RiskBERT

Transfer learning techniques are included in the methods to solve the problem of context understanding for pretrained language models is to fine-tuning them with a

Table 17: Micro-averaged MSE results before 2007

embedding	1 year	2 year	5 year
tf-idf (with prev)	0.1739	0.1647	0.1715
tf (with prev)	0.1703	0.1666	0.1742
xgboost(with prev)-RiskBERT	**0.1529**	**0.1486**	**0.1488**
random forest(with prev)-RiskBERT	0.1565	0.1515	0.1522
svr(with prev)-RiskBERT	0.1693	0.1670	0.1671
xgboost(with prev)-bert	0.1553	0.1500	0.1490
random forest(with prev)-bert	0.1570	0.1522	0.1521
svr(with prev)-bert	0.1746	0.1703	0.1705
tf-idf (without prev)	**0.1785**	**0.1736**	**0.1834**
tf (without prev)	0.1791	0.1746	0.1845
xgboost-RiskBERT	0.2653	0.2573	0.2718
random forest-RiskBERT	0.2568	0.2621	0.2745
svr-RiskBERT	0.2109	0.2092	0.2108
xgboost-bert	0.2636	0.2617	0.2768
random forest-bert	0.2612	0.2670	0.2791
svr-bert	0.2137	0.2108	0.2126

small dataset in the area of the task, in this case, finance. In short, transfer learning is "domain adaptation" for the implementation of context-specific natural language processing tasks [7].

Here, we tested the model only pretrained with large corpora, the BERT-base model, and the model fine-tuned version of the BERT model to see the effect of transfer learning in volatility prediction.

From the tables 13, 14, 15, 16 and 17, it is observed that for nearly all cases, the RiskBERT model performed better than the BERT-base model. In the table 18,

Table 18: Micro-averaged MSE results after 2006

embedding	1 year	2 year	5 year
tf-idf (with prev)	0.4312	0.4212	0.4171
tf (with prev)	0.4327	0.4222	0.4218
xgboost(with prev)-RiskBERT	0.2969	0.3470	0.3541
random forest(with prev)-RiskBERT	0.2878	0.2793	0.2945
svr(with prev)-RiskBERT	0.3515	0.3437	0.3210
xgboost(with prev)-bert	0.2808	0.2689	0.2511
random forest(with prev)-bert	0.2754	0.2682	0.2560
svr(with prev)-bert	0.3759	0.3640	0.3376
tf-idf (without prev)	0.4611	0.4565	0.4440
tf (without prev)	0.4629	0.4577	0.4490
xgboost-RiskBERT	0.3546	0.4743	0.3781
random forest-RiskBERT	0.3830	0.3796	0.4113
svr-RiskBERT	0.3669	0.3582	0.3374
xgboost-bert	0.4500	0.4415	0.4299
random forest-bert	0.4443	0.4409	0.4211
svr-bert	0.4522	0.4526	0.4356

it might seem that when previous data was used, the BERT-base model performed better than the RiskBERT model; however, if 2007 error values are not considered, which is an enormous value, the tide turns.

4.7.2 Effect of the dataset size

The expectation is that the greater the dataset size, the better the metric scores. However, this is not the case for volatility prediction for 10-K reports. As represented in table 17, extending the year does not always end up with better results.

This may stem from the reason that the train set is from the data belonging to previous years and the test set is not. In other words, train and test sets may be considered as from the different datasets [31].

For the word count feature-based models before 2007, training with 2-year-long periods generally gives the best result among other year periods. The same is true for the models using BERT-based embeddings as a feature. Although there are variations for the models using different machine learning methods, the models with a 2-year-period data fed during training performed mostly better than models using the recent 1-year and 5-year data.

The word count features in the models preserved the dataset size effect for the dataset that was collected after 2006. When they were provided with data from a five-year period, they produced the best outcomes, as it is presented in the table 18. The bert-based embeddings had distinct results. For the xgboost and random forest models using RiskBERT features with the addition of previous data, the one-year-long train set outperformed the other duration of train sets. Moreover, while in the ranking of the xgboost and random forest, 5-year-long periods got the second place, in the last place there are 2-year-long periods. SVR using RiskBERT embeddings with historical data employed the dataset size effect. Unexpectedly, all models using bert-base embeddings with the preceding year volatility values showed results in parallel with dataset size effect, which we anticipated a similar pattern with RiskBERT. Excluding previous year's volatility values, mostly, made all the models with each kind of features behaved concurrently with the dataset size effect.

4.7.3 Qualitative Evaluation

As mentioned before, there are two bert-based embeddings we used as features in these experiments. One of them is calculated as the mean value of each word embedding and the other is the standard deviation of the embeddings. For these two

Table 19: The closest words list to mean embedding vector

1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006
increased	increased	increased	increased	increased	increased	increased	reduced	increased	including	increased
reduced	reduced	reduced	reduced	reduced	reduced	reduced	including	total	increased	including
decreased	including	including	including	including	including	including	increased	reduced	reduced	reduced
including	decreased	decreased	decreased	decreased	decreased	decreased	total	including	total	decreased
reduction	reduction	reduction	reduction	reduction	reduction	total	decreased	income	decreased	losses
resulted	company	losses	includes	company	total	reduction	income	decreased	income	reduction
company	received	resulted	reflects	received	received	loss	sales	sales	sales	income
declined	resulted	became	several	sales	losses	income	loss	operating	operating	received
several	sales	expanses	became	became	sales	sales	gross	gross	losses	change
total	decline	sales	sales	rate	several	received	reduction	loss	change	total

Table 20: The closest words list to std embedding vector

1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006
company	company	company	company	company	company	company	company	several	several	company
believes	several	company	several	business	several	several	several	interest	company	market
including	business	operation	including	including	including	including	prominent	company	including	million
two	two	new	continued	interest	business	business	rate	including	interest	rate
operations	believes	implemented	operations	monitor	rates	rates	including	rate	financial	management
several	including	many	many	several	currently	currently	market	us	market	interest
began	operation	numerous	two	rate	market	market	interest	business	rate	including
intends	significant	investigations	market	million	interest	interest	significant	continued	business	several
management	management	subject	year	two	certain	certain	business	sales	risk	revolver
able	certain	effective	business	true	risk	risk	million	market	using	significant

Table 21: The closest words list to mean embedding vector after 2006

2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
increased	increased	increased	increased	increased	increased	increased	increased	increased	increased	increased	increased	increased	increased	increased
reduced	reduced	reduced	reduced	reduced	reduced	reduced	reduced	reduced	reduced	reduced	reduced	reduced	reduced	reduced
including	including	including	including	including	including	including	including	including	including	including	including	including	including	including
decreased	decreased	decreased	decreased	decreased	decreased	decreased	decreased	decreased	decreased	decreased	decreased	decreased	decreased	decreased
changes	using	using	strenghtening	changes	reported	products	reported	differences	changes	provided	using	values	using	estimated
received	reported	become	sold	using	make	make	provided	make	costs	largely	adjustment	adopted	reported	identified
using	reduction	reduction	recognized	loss	rate	decline	loss	loss	loss	loss	loss	loss	company	management

Table 22: The closest words list to std embedding vector after 2006

2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
item	also	also	market	also	also	ceded	several	also	according	departments	became	also	also	used
company	december	business	also	several	several	consists	company	annual	amount	disrupt	investment	average	business	also
doubtful	discussion	company	tax	besides	affect	several	services	approximately	exposed	several	also	became	certain	contemplated
interest	including	compensate	including	rates	credits	also	management	average	also	stock	credit	business	rate	sold
several	item	financial	several	discussion	write	loss	services	define	loss	tax	recognized	company	utilize	restricted
fair	operations	including	item	utilize	fair	included	rate	operations	using	select	net	risk	several	many
following	risk	several	financial	sale	operating	using	managed	measure	utilizing	including	one	following	testing	including

embedding values, the closest words embeddings to mean and standard deviation embedding vectors are decoded to see the most impactful words on the features among the years.

As it can be observed from the table 19, for the closest word list of the mean embedding vector, there are more words common during each year. For the 1996-2000 period, most of the determinator words are similar. In 2001, the word *loss* enters to the closest list, while in 2002, the word *income* enters. During the 2002-2006 period, few words such as *gross, operating* and *change* enter and exit.

For the nearest word list for the std embedding vector in the table 20, there is more variety in the words during years in comparison to list calculated for the mean embedding vector. Between 1996 and 2000 years, there are more difference in the words apart from the fixed 4 or 5 words. For the 2001-2006 period, diversity of the nearest words are stabler. Whereas in the 2001, *market* and *interest* is made to the list, in the 2000, *rate* enters list.

In the tables 22 and 21 the closest word list to mean and standard deviation embedding vector lists for the years after 2006. For the mean embedding vector, the closest list is formed by words defines change and the direction of change. Besides the word *change* itself, *increase, reduced, reduction, decline, loss* occurs in the list. Moreover, the words used in reporting like *reported, including, using* or *make* also added to the list. The std list is more diverse in comparison to mean embedding vector list after 2006. Most of the years has *also* and *several*. The fair amount of years has *including, company,business* and *using* in their list.

CHAPTER V

CONCLUSIONS

In this research, we have come up with FinSentiment model for financial tasks which incorporates enhanced versions of a number of recently-proposed neural language models, such as BERT, XLNet, RoBERTa. The corresponding finance-specific models in FinSentiment are called Fin-BERT, Fin-XLNet, and Fin-RoBERTa respectively. In general, FinSentiment models better perform across sentiment analysis tasks in financial domain by training these models on financial domain corpora. To our best knowledge, this is the first study to investigate the performance of fine-tuning commonly-used pretrained neural language models in financial domain and tasks. Additionally, it is one of the limited comprehensive studies that focuses on additional pretraining over domain-specific corpora. We have evaluated the performance of FinSentiment models over three different sentiment datasets. On all of the used datasets, we have significantly outperformed the state-of-the-art methods. For instance, for sentiment classification tasks, FinSentiment has outperformed the existing generally-trained versions and baselines by more than 10% in terms of accuracy and F1 scores.

When compared with the baseline and non-finance specific models, our results indicate the success of pretrained language models for a subsequent task such as financial sentiment analysis by using only a tiny labeled dataset. For instance, one of our datasets Financial PhraseBank, has almost 3000 training samples. However, our FinSentiment models has surpassed the baseline approaches even when we trained with as few as 500 samples. This result is important since deep learning approaches for natural language processing tasks has in general considered to be

extremely data-eager. According to the detailed experiments we have carried out with FinSentiment, we conclude that additional pretraining of language models on a finance-specific corpora has remarkably enhanced the performance. We have found RoBERTa to mostly outperform the remaining ones and fine-tuning it with the unsupervised financial text data generally improves its performance. We have also investigated the impact of additional pretraining and a number of different training styles. Catastrophic forgetting can be prevented more effectively by using learning rate styles which focus on finetuning the initial layers more strongly than the next layers for all of the considered models.

As a future work, analysis of financial sentiment cannot be an aim alone, and its analysis can be practical once it supports the decisions on financial tasks. As a further study, the proposed finance-specific models and mechanisms can be improved for classification purposes in terms of aspect levels, and regression purposes. Over and above, the model catalogue can be expanded. One possible extension will be to use the trained finance-specific models with stock market price datasets over financial tweets and news to predict both directionality of the returns as well as predicting stock return volatility. Finance-specific FinSentiment models are capable of inferring the explicit sentiments. However, modeling the implicit knowledge in the text which is hidden even to domain experts could be a difficult problem. Another future work might to be use the trained finance-specific model over financial text mining tasks different than financial sentiment analysis tasks, such as question answering and named entity recognition in financial domain.

Moreover, we have investigated the text regression using transfer learning models. Text regression is an NLP problem that uses text to forecast events that can be measured the actual world, such as volatility values of the companies, which are used as indicator of risk prediction.

We tested several methods to predict financial volatility from 10-K reports of corporations and discovered that predictions from text regression models using RiskBERT features with historical data had better results in general. When the preceding year's volatility value is excluded, for the time until 2006, the models using word count features outperformed other models. After 2006, bert-based embeddings outperformed the models using tf and tf-idf features independent of historical data usage. However, there is a discongruity in 2007 in terms of feature performance pattern and the MSE score values. In 2007, unforeseeing the Global Financial Crisis can be observed from the performance of the models having higher MSE scores than the usual. Also, the word count feature based models performed better than the rest of the models when data belonging to previous years is discarded. Furthermore, the mere year the models with BERT embeddings dominated the models with RiskBERT is 2007. After 2007, the models improved their performance until 2011. The Dodd-Frank Wall Street Reform and Consumer Protection Act, which is enacted in 2010 aiming 'sweeping overhaul of the United States financial regulatory system' after Great Recession, had presumptive effect on the improvements in the models after 2012. From 2012 to 2018, models have more stable performance while there is an increase after 2018. In 2021, there is a contrary behaviour for all the models having worse performance while adding the data belonging to previous year, which is the straight effect of COVID crisis.

REFERENCES

[1] S. Yang, J. Rosenfeld, and J. Makutonin, "Financial aspect-based sentiment analysis using deep representations," *arXiv preprint arXiv:1808.07931*, 2018.

[2] G. Piao and J. G. Breslin, "Financial aspect and sentiment predictions with deep neural networks: an ensemble approach," in *Companion Proceedings of the The Web Conference*, pp. 1973–1977, 2018.

[3] B. G. Malkiel, "The efficient market hypothesis and its critics," *Journal of Economic Perspectives*, vol. 17, pp. 59–82, March 2003.

[4] M. González-Sánchez and M. E. Morales de Vega, "Influence of bloomberg's investor sentiment index: Evidence from european union financial sector," *Mathematics*, vol. 9, no. 4, 2021.

[5] P. C. Tetlock, M. Saar-Tsechansky, and S. Mackassy, "More than words: Quantifying language to measure firms' fundamentals," *The Journal of Finance*, vol. 63, no. 3, pp. 1437–1467, 2008.

[6] T. Loughran and B. McDonald, "Textual analysis in accounting and finance: A survey," *Journal of Accounting Research*, vol. 54, no. 4, pp. 1187–1230, 2016.

[7] Q. Yang, Y. Zhang, W. Dai, and S. J. Pan, *Transfer learning*. Cambridge University Press, 2020.

[8] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

[9] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, (Red Hook, NY, USA), p. 3111–3119, Curran Associates Inc., 2013.

[10] M. Zaib, Q. Z. Sheng, and W. Emma Zhang, "A short survey of pre-trained language models for conversational ai-a new age in nlp," in *Proceedings of the Australasian Computer Science Week Multiconference*, pp. 1–4, 2020.

[11] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, "Pre-trained models for natural language processing: A survey," *Science China Technological Sciences*, pp. 1–26, 2020.

[12] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of*

the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), (New Orleans, Louisiana), pp. 2227–2237, Association for Computational Linguistics, June 2018.

[13] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Melbourne, Australia), pp. 328–339, Association for Computational Linguistics, July 2018.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, June 2019.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

[16] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[17] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, pp. 5753–5763, Curran Associates, Inc., 2019.

[18] M. Mccloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," *The Psychology of Learning and Motivation*, vol. 24, pp. 104–169, 1989.

[19] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems." `https://www.tensorflow.org/`. Accessed: 2022-07-17.

[20] T. Wolf *et al.*, "Huggingface's transformers: State-of-the-art natural language processing," *ArXiv*, vol. abs/1910.03771, 2019.

[21] A. J. Patton and K. Sheppard, *Evaluating Volatility and Correlation Forecasts*, pp. 801–838. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[22] B. Dumas, A. Kurshev, and R. Uppal, "Equilibrium portfolio strategies in the presence of sentiment risk and excess volatility," *The Journal of Finance*, vol. 64, no. 2, pp. 579–629, 2009.

[23] T. Andersen, T. Bollerslev, P. Christoffersen, and F. Diebold, "Chapter 15 volatility and correlation forecasting," *Handbook of Economic Forecasting*, vol. 1, pp. 777–878, 12 2006.

[24] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, vol. 31, no. 3, pp. 307–327, 1986.

[25] C. K. Theil, S. Broscheit, and H. Stuckenschmidt, "Profet: Predicting the risk of firms from event transcripts," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 5211–5217, International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[26] C.-J. Wang, M.-F. Tsai, T. Liu, and C.-T. Chang, "Financial sentiment analysis for risk prediction," in *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, (Nagoya, Japan), pp. 802–808, Asian Federation of Natural Language Processing, Oct. 2013.

[27] M. Tsai and C. Wang, "Financial keyword expansion via continuous word vector representations," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL* (A. Moschitti, B. Pang, and W. Daelemans, eds.), pp. 1453–1458, ACL, 2014.

[28] W. Y. Wang and Z. Hua, "A semiparametric Gaussian copula regression model for predicting financial risks from earnings calls," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Baltimore, Maryland), pp. 1155–1165, Association for Computational Linguistics, June 2014.

[29] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction," in *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, p. 2327–2333, AAAI Press, 2015.

[30] P. C. Tetlock, "Giving content to investor sentiment: The role of media in the stock market," *The Journal of Finance*, vol. 62, no. 3, pp. 1139–1168, 2007.

[31] S. Kogan, D. Levin, B. R. Routledge, J. S. Sagi, and N. A. Smith, "Predicting risk from financial reports with regression," in *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics*, (Boulder, Colorado), pp. 272–280, Association for Computational Linguistics, June 2009.

[32] T. Dyer, M. Lang, and L. Stice-Lawrence, "The ever-expanding 10-k: Why are 10-ks getting so much longer (and does it matter)?," *SSRN Electronic Journal*, 01 2016.

[33] B. Liu, "Sentiment analysis and opinion mining," *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1, pp. 1–167, 2012.

[34] B. Agarwal and N. Mittal, *Machine Learning Approach for Sentiment Analysis*, pp. 21–45. Cham: Springer International Publishing, 2016.

[35] J. Martineau and T. Finin, "Delta tfidf: An improved feature space for sentiment analysis," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 3, pp. 258–261, Mar. 2009.

[36] A. Tripathy, A. Agrawal, and S. K. Rath, "Classification of sentiment reviews using n-gram machine learning approach," *Expert Systems with Applications*, vol. 57, pp. 117–126, 2016.

[37] C. Whitelaw, N. Garg, and S. Argamon, "Using appraisal groups for sentiment analysis," in *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM '05, (New York, NY, USA), p. 625–631, Association for Computing Machinery, 2005.

[38] O. Araque, I. Corcuera-Platas, J. F. Sánchez-Rada, and C. A. Iglesias, "Enhancing deep learning sentiment analysis with ensemble techniques in social applications," *Expert Systems with Applications*, vol. 77, pp. 236–246, 2017.

[39] A. Severyn and A. Moschitti, "Twitter sentiment analysis with deep convolutional neural networks," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, (New York, NY, USA), p. 959–962, Association for Computing Machinery, 2015.

[40] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1253, 2018.

[41] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[42] X. Li, H. Xie, L. Chen, J. Wang, and X. Deng, "News impact on stock price return via sentiment analysis," *Knowledge-Based Systems*, vol. 69, pp. 14–23, 2014.

[43] T. Loughran and B. McDonald, "When is a liability not a liability? textual analysis, dictionaries, and 10-ks," *The Journal of Finance*, vol. 66, no. 1, pp. 35–65, 2011.

[44] P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala, "Good debt or bad debt: Detecting semantic orientations in economic texts," *J. Assoc. Inf. Sci. Technol.*, vol. 65, p. 782–796, Apr. 2014.

[45] V. S. Pagolu, K. N. Reddy, G. Panda, and B. Majhi, "Sentiment analysis of twitter data for predicting stock market movements," in *International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, pp. 1345–1350, IEEE, 2016.

[46] M. Kraus and S. Feuerriegel, "Decision support from financial disclosures with deep neural networks and transfer learning," *Decision Support Systems*, vol. 104, Oct 2017.

[47] S. Sohangir, D. Wang, A. Pomeranets, and T. M. Khoshgoftaar, "Big data: Deep learning for financial sentiment analysis," *Journal of Big Data*, vol. 5, no. 1, pp. 1–25, 2018.

[48] B. Lutz, N. Pröllochs, and D. Neumann, "Sentence-level sentiment analysis of financial news using distributed text representations and multi-instance learning," *ArXiv*, vol. abs/1901.00400, 2019.

[49] M. Maia, A. Freitas, and S. Handschuh, "Finsslx: A sentiment analysis model for the financial domain using text simplification," in *IEEE 12th International Conference on Semantic Computing (ICSC)*, (Los Alamitos, CA, USA), pp. 318–319, IEEE Computer Society, Feb 2018.

[50] N. Kant, R. Puri, N. Yakovenko, and B. Catanzaro, "Practical text classification with large pre-trained language models," *ArXiv*, vol. abs/1812.01207, 2018.

[51] B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heinz, and D. Roth, "Recent advances in natural language processing via large pre-trained language models: A survey," *arXiv preprint arXiv:2111.01243*, 2021.

[52] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune bert for text classification?," in *Chinese Computational Linguistics* (M. Sun, X. Huang, H. Ji, Z. Liu, and Y. Liu, eds.), (Cham), pp. 194–206, Springer International Publishing, 2019.

[53] D. Araci, "Finbert: Financial sentiment analysis with pre-trained language models," *CoRR*, vol. abs/1908.10063, 2019.

[54] Z. Liu, D. Huang, K. Huang, Z. Li, and J. Zhao, "Finbert: A pre-trained financial language representation model for financial text mining," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20* (C. Bessiere, ed.), pp. 4513–4519, International Joint Conferences on Artificial Intelligence Organization, 7 2020. Special Track on AI in FinTech.

[55] P. K. Y Yang, K.Z. Zhang, M. C. S. Uy, and A. Huang, "Finbert: A pretrained language model for financial communications," *ArXiv*, vol. abs/2006.08097, 2020.

[56] V. Desola, K. Hanna, and P. Nonis, "Finbert: pre-trained model on sec filings for financial natural language tasks," tech. rep., 08 2019.

[57] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "Biobert: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[58] E. Asgari and M. R. Mofrad, "Continuous distributed representation of biological sequences for deep proteomics and genomics," *PloS one*, vol. 10, no. 11, p. e0141287, 2015.

[59] S. Kazemian, S. Zhao, and G. Penn, "Evaluating sentiment analysis evaluation: A case study in securities trading," in *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 119–127, 2014.

[60] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction," in *Twenty-fourth International Joint Conference on Artificial Intelligence*, 2015.

[61] B. Xie, R. Passonneau, L. Wu, and G. G. Creamer, "Semantic frames to predict stock price movement," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pp. 873–883, 2013.

[62] R. Luss and A. d'Aspremont, "Predicting abnormal returns from news using text classification," *Quantitative Finance*, vol. 15, no. 6, pp. 999–1012, 2015.

[63] T. H. Nguyen and K. Shirai, "Phrasernn: Phrase recursive neural network for aspect-based sentiment analysis," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 2509–2514, 2015.

[64] W. Y. Wang and Z. Hua, "A semiparametric gaussian copula regression model for predicting financial risks from earnings calls," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1155–1165, 2014.

[65] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, p. 1735–1780, nov 1997.

[66] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, pp. 4171–4186, 2019.

[67] M. Maia, S. Handschuh, A. Freitas, B. Davis, R. McDermott, M. Zarrouk, and A. Balahur, "Www'18 open challenge: Financial opinion mining and question answering," in *Companion Proceedings of the The Web Conference*, WWW '18, (Republic and Canton of Geneva, CHE), p. 1941–1942, International World Wide Web Conferences Steering Committee, 2018.

[68] A. H. Huang, A. Y. Zang, and R. Zheng, "Evidence on the information content of text in analyst reports," *The Accounting Review*, vol. 89, no. 6, pp. 2151–2180, 2014.

[69] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," *Advances in Neural Information Processing Systems*, vol. 9, 1996.

[70] S. J. Rigatti, "Random forest," *Journal of Insurance Medicine*, vol. 47, no. 1, pp. 31–39, 2017.

[71] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, *et al.*, "Xgboost: extreme gradient boosting," *R package version 0.4-2*, vol. 1, no. 4, pp. 1–4, 2015.

[72] G. A. Karolyi, "Why stock return volatility really matters," *Institutional Investor Journals Series*, 2001.

[73] P. Sarbanes, "Sarbanes-oxley act of 2002," in *The Public Company Accounting Reform and Investor Protection Act. Washington DC: US Congress*, vol. 55, 2002.

[74] U. S. Congress, *Dodd-Frank Wall Street Reform and Consumer Protection Act: Conference Report (to Accompany HR 4173).*, vol. 111. US Government Printing Office, 2010.

VITA

Zehra Erva Ergün had her bachelor's degree in Electrical and Electronics Engineering from Boğaziçi University, where she also had a double major in Maths, in 2017. She has been working as a Graduate Teaching and Research Assistant at Özyeğin University under the supervision of Asst. Prof. Dr. Emre Sefer. Her research focuses on natural language processing applications in finance domain.