Exploring Robustness of Prefix Tuning in Noisy Data: A Case Study in Financial Sentiment Analysis

Sudhandar Balakrishnan and **Yihao Fang** and **Xiaodan Zhu**
Department of Electrical and Computer Engineering & Ingenuity Labs Research Institute
Queen's University
{sudhandar.balakrishnan, yihao.fang, xiaodan.zhu}@queensu.ca

Abstract

The invention of transformer-based models such as BERT, GPT, and RoBERTa has enabled researchers and financial companies to finetune these powerful models and use them in different downstream tasks to achieve state-of-the-art performance. Recently, a lightweight alternative (approximately 0.1% - 3% of the original model parameters) to fine-tuning, known as prefix tuning has been introduced. This method freezes the model parameters and only updates the prefix to achieve performance comparable to full fine-tuning. Prefix tuning enables researchers and financial practitioners to achieve similar results with much fewer parameters. In this paper, we explore the robustness of prefix tuning when facing noisy data. Our experiments demonstrate that fine-tuning is more robust to noise than prefix tuning—the latter method faces a significant decrease in performance on most corrupted data sets with increasing noise levels. Furthermore, prefix tuning has high variances on the F1 scores compared to fine-tuning in many corruption methods. We strongly advocate that caution should be carefully taken when applying the state-of-the-art prefix tuning method to noisy data.

1 Introduction

The transformer architecture (Vaswani et al., 2017) has given rise to several powerful language models such as BERT (Devlin et al., 2018) and GPT (Radford et al., 2018). These models are trained on large text corpora and the pre-trained models can be used on different downstream tasks by finetuning these models, which refers to the process of updating the weights of the pre-trained model to adapt to the downstream task and the associated dataset. This approach is critical in achieving state-of-the-art results in many downstream tasks. However, these fine-tuned language models are large in size and the deployment of these models in production to solve real-world problems becomes difficult due to the memory requirement, constraining the deployment of models in many real-life financial applications. Given that it is anticipated that model sizes will continue to rise, this will become more serious.

Li and Liang (2021) introduced a lightweight alternative to finetuning known as prefix tuning. The authors freeze the model parameters of GPT-2 (Radford et al., 2019) and use a task-specific vector to tune the model for natural language generation. This method achieves comparable performance with finetuning and uses approximately 0.1% - 3% of the original model parameters. This method will enable the use of pre-trained language models for many industrial applications.

In the financial sector, natural language processing has a wide variety of applications ranging from building a chatbot to interact with customers (Yu et al., 2020), predicting stock movements based on sentiments from financial news headlines and tweets (Sousa et al., 2019), to summarizing financial reports (La Quatra and Cagliero, 2020). Prefix tuning can be applied to many tasks with fewer parameters and much less memory consumption.

However, in the real world, the data might be noisy, especially in the case of chatbots and social media data where misspellings, typographical errors, and out-of-vocabulary words occur frequently. Recent studies have investigated the robustness of finetuning language models such as Rychalska et al. (2019), Jin et al. (2020), Aspillaga et al. (2020), Sun et al. (2020) and Srivastava et al. (2020), and found that finetuning is not robust to noisy texts.

To the best of our knowledge, there have been no studies that explore the robustness of prefix tuning that reflect real-life scenarios and compare it with finetuning to identify the more robust method. Our work corrupts the financial phrasebank dataset (Malo et al., 2014), using various text corruption methods such as keyboard errors (typos), inserting random characters, deleting random words, replacing characters with OCR alternatives and replacing

Proceedings of the Fourth Workshop on Financial Technology and Natural Language Processing (FinNLP), pages 78 - 88

words with antonyms by varying percentages in each sentence. The corrupted dataset is used with two widely used pre-trained models, BERT-base (Devlin et al., 2018) and RoBERTa-large (Liu et al., 2019), under both prefix tuning and fine-tuning, to compare their performance at different noise levels. In addition, we evaluate the performance on a Kaggle Stock Market Tweets dataset (Chaudhary, 2020), which is a real-life noisy dataset. With our experiments, we show that fine-tuning is more robust than prefix tuning in most setups. Fine-tuning updates the weights based on the downstream task and the dataset, and because of this, it can adapt to the noise, whereas prefix tuning uses the pre-trained model without updating the weights which limits the model from learning task-oriented information when facing noisy data. In summary, the contributions of this paper are three-fold.

- To the best of our knowledge, this is among the first efforts in exploring the robustness of prefix tuning when facing noisy data, particularly noisy financial data.
- We use a comprehensive set of corrupted data and show that fine-tuning is more robust to noise compared to prefix tuning. The latter has also shown to have high variances in F1 scores.
- We provide detailed results at different levels of noise. With that, we advocate that caution should be carefully taken when practitioners apply state-of-the-art prefix tuning methods to noisy data. We hope our work will set baselines for further studies along this line.

2 Related Work

2.1 Sentiment Analysis in Financial Text

Sentiment analysis is the process of understanding the sentiments from textual data (Liu, 2012). Sentiment analysis in finance tries to achieve a different objective when compared to general sentiment analysis. Financial sentiment analysis aims to predict the stock movement or impact on stock price based on the sentiments of news headlines and news articles (Li et al., 2014). Loughran and McDonald (2016) provide a survey of the machine learning approaches used to predict the sentiments in financial data. With the introduction of transformer-based language models like BERT (Devlin et al., 2018), several attempts have been made to predict the sentiments using the pre-trained BERT models trained on large text corpora. Araci (2019) introduced FinBERT, where the BERT model was pre-trained on a large financial corpus and it achieved state-of-the-art results in financial sentiment analysis. Zhao et al. (2021) use RoBERTa (Liu et al., 2019), an optimized version of BERT to predict the sentiment of online financial texts generated on social media.

2.2 Robustness of Pretrained Language Models

Several attempts have been made to test the robustness of popular transformer-based language models. Rychalska et al. (2019) test the robustness of ULMFiT (Howard and Ruder, 2018) on various NLP tasks like QA, NLI, NER and Sentiment Analysis. The authors found that the high-performing language models are not robust to various corruption methods like removing articles, removing characters from words, misspellings, etc. Jin et al. (2020) introduced a technique called TEXTFOOLER to generate adversarial texts. The authors successfully attacked BERT and significantly reduced the accuracy of BERT on text classification tasks. Aspillaga et al. (2020) compared the robustness of RoBERTa, BERT and XLNET (Yang et al., 2019) with recurrent neural network models and found that RoBERTa, BERT and XLNET are more robust than recurrent neural networks but they are still not fully immune to the attacks and their robustness can be improved. Sun et al. (2020) performed a detailed study on the robustness of BERT, especially concerning mistyped words (keyboard typos) and found that typos in informative words affect the performance of the BERT to a greater extent than typos in other words. Srivastava et al. (2020) analyzed the robustness of BERT to noise (spelling mistakes and typos) on sentiment analysis and textual similarity. The authors discovered that BERT's performance had significantly declined in the presence of noise in the text.

Prefix tuning freezes the parameters of the language model and updates the prefix vector for downstream tasks. Yang and Liu (2022) used the GPT-2 (Radford et al., 2019) model to evaluate the robustness of prefix tuning to various textual adversarial attacks, but the attacks do not resemble the noise presented in real-world data. The authors did not compare the robustness of prefix tuning and fine-tuning and did not study which training methodology is more robust.

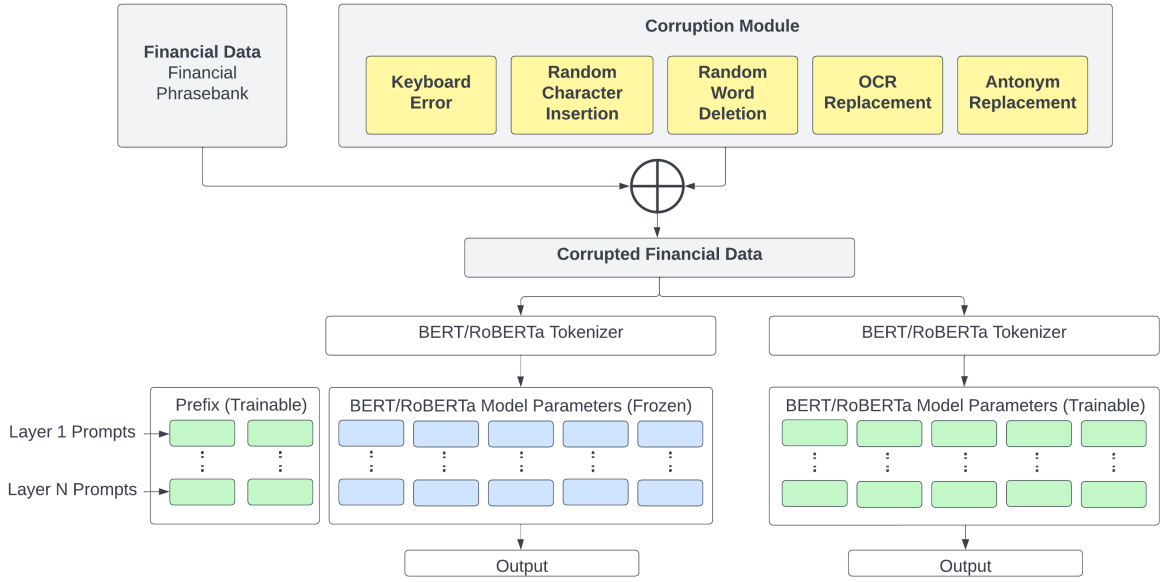


Figure 1: Overview of prefix tuning and fine-tuning methodologies. Green boxes represent trainable parameters and blue boxes represent frozen parameters.

Table 1: Train-validation-test split for the financial phrasebank 50% agreement level dataset

Label	Train Set	Validation Set	Test Set
Neutral	2011	431	431
Positive	954	204	205
Negative	423	91	90
Total	3388	726	726

Table 2: Train-validation-test split for the financial phrasebank 100% agreement level dataset

Label	Train Set	Validation Set	Test Set
Neutral	973	209	209
Positive	399	85	86
Negative	212	46	45
Total	1584	340	340

Table 3: Train-validation-test split for the Kaggle Stock Market Tweets dataset

Label	Train Set	Validation Set	Test Set
Positive	2577	553	552
Negative	1470	315	315
Total	4047	868	867

3 Approach

Figure 1 shows the overview of the approach used in this paper. The clean financial dataset is corrupted using the corruption module represented by yellow boxes in Figure 1, containing various corruption methods. The corruption module is explained in section 3.1. The corrupted financial dataset is fed into two state-of-the-art pre-trained models, BERT-base and RoBERTa-large (refer to section 3.2) using both prefix tuning and fine-tuning. Figure 1 also shows the difference between the traditional fine-tuning method and prefix tuning respectively, where blue boxes represent the frozen parameters and green boxes represent the trainable parameters.

3.1 Corruption Module

The corruption module consists of 5 text corruption methods which closely replicate the noise found in real-world data. This module is used to corrupt the clean financial dataset and the corrupted dataset is used to evaluate the performance of the models. The following are the various text corruption methods used in the corruption module. Table 4 shows an example for each corruption method. The nlpaug (Ma, 2019) library is used for generating the various corruption methods.

Keyboard Error (QWERTY) Simulates typing mistakes made while using a QWERTY-type keyboard.

Table 4: Corruption methods with an example

Corruption Method	Example
Original Sentence	In Finland's Hobby Hall's sales decreased by 10% , and international sales fell by 19% .
Keyboard Error	In cinland' s Hubby Hall' s sales decreased by 10% , and international saleW fell by 19%.
Random Character Insertion	In FrinDIa*nd' s HZobJb#y Hall's sales decreased by 10% , and international sales fell by 19% .
Random Word Deletion	In Finland' s Hobby Hall' s decreased 10% , and international fell by 19%.
OCR Replacement	In Finland' s H066y Ha11's sa1es decreased by 10% , and national sales decreased by 19%.
Antonym Replacement	In Finland' s Hobby Hall' s sales increase by 10% , and national sales increase by 19%.

Random Character Insertion (ChIns) Inserts random characters into a word in a sentence.

Random Word Deletion (WdDel) Randomly deletes a word from the sentence.

OCR Replacement (OCR) Replaces the characters in the word with their OCR equivalents, e.g., stock can be replaced as st0ck (here an alphabet, o, is replaced with the number zero, 0)

Antonym Replacement (AntRep) Replaces the words with their antonyms (opposite meaning) in the sentence.

3.2 Prefix Tuning and Fine Tuning in Noisy Data

Noisy Data Analysis When the models BERT and RoBERTa encounter a word that is not in their vocabulary, the models try to break down the word to see whether any of its subwords are present in their vocabulary. For example, if BERT has the word 'play' in its vocabulary and when it encounters 'playing' it will tokenize the word as "'play' + '#ing"'. If any word is not present in the vocabulary even after breaking it down, BERT assigns the unknown token (<UNK>) to that word. Table 5 shows how BERT and RoBERTa tokenize the normal and the corrupted word. From Table 5 we can understand how the corrupted word affects the BERT tokenizer and prevents it from learning the word's original meaning resulting in a drop in performance.

The process of prefix tuning and fine-tuning updating the weights is based on the downstream task and the dataset. In prefix tuning, most of the weights are not updated based on the downstream task. Since both the training and the validation sets are corrupted, in fine-tuning, the weights of the model have been updated based on the noisy datasets and contain more dataset-specific information than the prefix-tuned model. This enables the fine-tuned model to adjust to the noisy scenarios better than the prefix-tuned models. Evaluations of our intuition for prefix tuning and fine-tuning in noisy data can be found in Section 4.

4 Experiments

4.1 Financial Tasks

Two financial tasks are used to evaluate the performance of prefix tuning. The first task is the sentiment analysis of the Financial Phrasebank dataset (Malo et al., 2014), which is the main dataset used to compare the performance and evaluate the robustness of both prefix tuning and fine-tuning. The second task is the sentiment analysis of the Twitter Stockmarket dataset from Kaggle, Chaudhary (2020), which is also used to evaluate the performance of prefix tuning and fine-tuning.

Financial Phrasebank The Financial Phrasebank dataset (Malo et al., 2014), consists of 4840 sentences from financial news articles and the sentences were manually labelled as positive, negative or neutral by 16 annotators with backgrounds in finance and business. The annotators labelled the sentences depending on whether the information from the sentence had a positive, negative or no impact on the stock prices of the company mentioned in the sentence. It is an imbalanced dataset with 1363 positive sentences, 604 negative sentences and 2873 neutral sentences. In addition to it, depending

Table 5: Tokenization of corruption variants for the word 'stock'

Corruption Method	Corrupted Word	Tokenized Word	
		BERT	RoBERTa
No Corruption	'stock'	['stock']	['stock']
Keyboard error	'srosk'	['s', '##ros', '##k']	['s', 'ros', 'k']
Random character insertion	'sto*rck'	['s', '##to', '*', 'r', '##ck']	['st', 'o', '*', 'r', 'ck']
OCR replacement	'st0ck'	['s', '##t', '##0', '##ck']	['st', '0', 'ck']

on the agreement level among the annotators on the polarity of the sentence, the dataset was classified into 50%, 66%, 75% and 100% agreement levels. For example, 50% annotator agreement means more than 50% of the annotators agreed and selected the same polarity for a particular sentence. This paper uses the financial phrasebank dataset with 50% annotator agreement level (4840 sentences) to run the experiments on estimating the robustness of prefix tuning and the 100% agreement level (2262 sentences) to compare the performance. The dataset was split into the train, validation and test sets for the experiments with a 70-15-15 split (stratified split) giving rise to 3388 training sentences, 726 validation sentences and 726 test sentences in the 50% agreement level and 1582 training sentences, 340 validation sentences and 340 test sentences in the 100% agreement level dataset. Table 1 shows the split up of the 50% agreement level dataset and Table 2 shows the split up of the 100% agreement level dataset.

Kaggle Stock Market Tweets The Stock Market tweets dataset is from Kaggle, Chaudhary (2020). The reason for selecting this dataset is to evaluate the performance of prefix tuning and fine-tuning on a real-world noisy data. This dataset contains tweets from Twitter consisting of information about the stocks of multiple companies and the tweets are labelled as either positive or negative based on the sentiment associated with each tweet. This dataset is from Kaggle and it is not from a renowned journal and the authenticity cannot be validated. The dataset consists of 2106 negative tweets and 3685 positive tweets. The dataset was split into the train, validation and test sets with a 70-15-15 split giving rise to 4047 training sentences, 868 validation sentences and 867 test sentences. Table 3 shows the split up of the Kaggle Stock Market dataset.

4.2 Setup

Corruption Strategy The clean versions of the financial phrasebank dataset, 100% agreement level and 50% agreement level, are used to evaluate the performance of prefix tuning and fine-tuning on both BERT-base and RoBERTa-large models to establish the baseline performance levels. To test the robustness of prefix tuning and find out which one between prefix tuning and fine-tuning is more robust to the noisy text, the train and validation sets of the financial phrasebank dataset (50% agreement level) are corrupted by various text corruption methods. The reason for corrupting the train and validation sets is that it is difficult to find large-scale high-quality training data, especially with respect to chatbots and social media texts in an industrial setting. In general, test data is smaller in size compared to the training data and can be manually cleaned before feeding into the model. Due to this, the training and validation sets have been corrupted. For each corruption method, the sentences are corrupted by 10%, 20%, 30%, 40% and 50% corruption levels. Each corruption level represents the percentage of corrupted words in a sentence. For example, 10% corruption level means 10% of the words in the sentence are corrupted. For antonym replacement, all the words which have antonyms in the nlpaug (Ma, 2019) library are replaced with antonyms and there are no varying corruption levels for this particular corruption method.

Implementation Details After corrupting the dataset using the above-mentioned corruption strategy, we conduct the experiments on two models, BERT base and RoBERTa large. The BERT base fine-tuned model has 108,312,579 trainable parameters while the prefix-tuned model has 370,947 trainable parameters for 30 epochs for the financial phrasebank dataset. Similarly, the RoBERTa large fine-tuned model has 355,362,819 trainable parameters while the prefix-tuned model has 986,115 trainable parameters for 30 epochs for the financial phrasebank dataset. More information about the implementation details can be found in Appendix A.1 for replication.

Table 6: Results for the uncorrupted version of the datasets for the BERT-base model

	Prefix Tuning		Fine Tuning	
Dataset	**Acc.(%)**	**F1(%)**	**Acc.(%)**	**F1(%)**
Financial Phrasebank - All agree	97.35	97.01	96.17	96.80
Financial Phrasebank - More than 50% agree	86.91	85.55	86.09	85.48
Kaggle Stock Market Tweets	79.60	77.74	80.41	78.96

Table 7: Results for the uncorrupted version of the datasets for the RoBERTa-large model

	Prefix Tuning		Fine Tuning	
Dataset	**Acc.(%)**	**F1(%)**	**Acc.(%)**	**F1(%)**
Financial Phrasebank - All agree	98.24	98.09	98.53	98.35
Financial Phrasebank - More than 50% agree	87.60	87.25	88.15	87.45
Kaggle Stock Market Tweets	81.79	79.61	82.71	80.61

Table 8: Financial Phrasebank results for various text corruption methods for both the BERT-base and the RoBERTa-large model

		BERT-base				RoBERTa-large			
	Cor.	**Prefix Tuning**		**Fine Tuning**		**Prefix Tuning**		**Fine Tuning**	
Method	**(%)**	**Acc.**	**F1**	**Acc.**	**F1**	**Acc.**	**F1**	**Acc.**	**F1**
None	-	86.91	85.55	86.09	85.48	87.60	87.25	88.15	87.45
Qwerty	10	0.47	-0.47	-0.16	-0.5	-0.78	-1.60	-0.94	-1.63
	20	-0.96	-0.44	-0.01	-0.43	-1.40	-1.43	-1.08	-0.82
	30	-0.16	-0.68	0.15	-1.09	-4.37	-5.50	-3.26	-4.64
	40	-1.58	-2.50	-0.81	-1.12	-6.21	-8.85	-3.56	-4.55
	50	-6.34	-8.87	-3.04	-5.11	-6.21	-9.04	-3.57	-4.80
ChIns	10	-1.10	-1.27	-0.65	-1.31	-0.46	-1.58	-0.46	-1.84
	20	-3.01	-3.53	-0.96	-1.37	-2.80	-3.30	-1.86	-3.09
	30	-3.33	-4.57	-0.96	-2.40	-2.78	-3.81	-3.73	-5.15
	40	-3.01	-4.63	-3.68	-4.22	-2.64	-4.41	-4.04	-4.24
	50	-5.38	-6.63	-2.89	-4.73	-5.56	-8.31	-5.13	-7.40
WdDel	10	-0.63	-0.04	0.15	-1.42	-0.94	-1.45	-1.40	-1.26
	20	-0.96	-0.98	-0.82	-1.70	-1.70	-2.48	-1.56	-2.27
	30	-1.90	-2.49	-1.29	-2.06	-4.04	-3.75	-2.80	-3.69
	40	-3.01	-4.63	-0.80	-2.29	-1.70	-2.05	-0.94	-1.74
	50	-5.38	-6.63	-3.21	-3.11	-2.02	-2.86	-2.02	-2.29
OCR	10	-0.63	-0.28	-1.13	-1.04	-0.78	-1.31	-0.78	-1.89
	20	-0.79	-1.30	-0.01	-0.27	-0.94	-1.42	-0.78	-1.67
	30	-2.53	-2.93	-2.73	-2.69	-2.48	-3.01	-3.57	-4.04
	40	-2.53	-2.93	-2.73	-2.69	-4.66	-5.08	-2.80	-4.34
	50	-7.44	-11.30	-10.73	-10.45	-5.13	-8.59	-4.19	-6.62
AntRep	-	-12.36	-25.55	-14.25	-27.41	-14.13	-28.20	-16.78	-30.05

Evaluation Metrics The F1 score and accuracy are selected as the metrics for the evaluation of the experiments. The F1 score is used as the main metric for comparison since the financial phrasebank is an imbalanced dataset with 3 classes, positive, negative and neutral.

4.3 Results

Clean Baselines Table 6 and Table 7 show the performance of both models on the clean versions of the financial phrasebank dataset and the noisy Kaggle stock market tweets dataset (uncorrupted). Both prefix tuning and fine-tuning achieve comparable performance in both clean versions of the financial phrasebank dataset (all agree and more than 50% agree). In the noisy tweets dataset, fine-tuning performs better than prefix tuning in both models. The Bert-base finetuning method achieves an F1 score of 78.96% which is greater than prefix tuning (77.74%) by 1.22 point F1 score. Similarly, RoBERTa fine-tuning method achieves an F1 score of 80.61% which is greater than prefix tuning (79.61%) by 1 point F1 score.

Corruption Results Table 8 shows the change in the baseline scores of prefix tuning and fine-tuning on different corruption methods for BERT-base and RoBERTa-large respectively. The performance of both fine-tuning and prefix tuning drops as the noise level increases. Overall, finetuning performs better than prefix tuning in all the corruption methods except for antonym replacement. Even though the difference in F1 scores is very minimal for the lower percentage of noise like 10% and 20%, the difference becomes more predominant when the noise percentage in each sentence increases. This trend can be observed in both BERT-base and RoBERTa-large models.

To further evaluate the validity of the results, the variance (how the F1 scores vary from mean F1 scores across various iterations) for 50% noise level for all the corruption methods is measured. The experiments were repeated 5 times with reshuffled data for all the corruption methods to measure the mean and variance of F1 scores. Table 9 shows the mean and variance of F1 scores for the BERT-base model. It can be observed that the variance for prefix tuning is very high in two corruption methods, keyboard (qwerty) error and OCR replacement error.

There is a significant drop in performance (more than 25%) for antonym replacement. Fine tuning achieves an F1 score of 62.05% whereas prefix tuning achieves an F1 score of 63.69%. When compared to prefix tuning, the fine-tuned model achieves lower performance and it could be due to the following reason. The weights of the fine-tuned model are updated with the corrupted dataset containing antonyms instead of the original words. Since the model is trained to predict the opposite sentiment (sentences with antonyms), the performance drops significantly when evaluated on the test dataset. This results in the fine-tuned model being more adapted to the corrupted dataset and achieving lower performance when exposed to a clean test dataset whereas prefix tuning performs comparatively better.

Table 10 shows the predicted labels for BERT-base OCR replacement 50% corruption level where fine-tuning predicted the correct labels and prefix tuning predicted the wrong labels. In most of the cases, the positive labels were incorrectly predicted as neutral, the neutral labels were incorrectly predicted as positive and the negative labels were incorrectly predicted as neutral.

Another interesting observation is the minimal performance drop seen in the random word deletion corruption method even when 50% of the words are deleted from the sentences. The performance drop in the F1 score for the BERT base model was 6.63% for prefix tuning and 3.11% for fine-tuning. Similarly, the performance drop in the F1 score for the RoBERTa large model was 2.86% for prefix tuning and 2.29% for fine-tuning. The main reason behind this could be the way BERT is trained. BERT uses masked language modelling where it masks the words at random by varying percentages and tries to predict the masked word based on the context. This might be the reason why there is no significant drop in performance even when deleting 50% of the words since both BERT and RoBERTa are trained to handle the missing words in a sentence.

5 Conclusion

With the sizes of pre-trained models continuing to be significantly larger, lightweight models have become more important for many financial applications. However, the robustness of such models has not been well understood yet. In this paper, we explored the robustness of prefix tuning by corrupting the financial phrasebank dataset with various corruption methods, including keyboard (qwerty) er-

Table 9: Mean and Variance of F1 scores for the BERT-base model for 50% noise level

	Prefix Tuning		Fine Tuning	
Corruption Method	**Mean (F1%)**	**Variance**	**Mean (F1%)**	**Variance**
No Corruption	85.48	0.16	85.48	0.13
Keyboard error	80.57	5.66	82.00	1.15
Random character insertion	80.84	0.72	81.97	0.86
Random word deletion	81.98	0.10	82.50	0.13
OCR replacement	75.77	3.64	77.49	0.86
Antonym replacement	64.05	1.94	62.23	0.06

Table 10: Predicted labels for BERT-base OCR replacement 50% corruption level in cases where fine-tuning predicted the correct labels and prefix tuning predicted the wrong labels

Sentence	**True Label**	**Predicted Label**	
		Prefix Tuning	**Fine Tuning**
The amending of the proposal simplifies the proposed plan and increases the incentive for key employees to stay in the Company	Positive	Neutral	Positive
The company 's net sales in 2009 totalled MEUR 307.8 with an operating margin of 13.5 per cent	Neutral	Positive	Neutral
The move was triggered by weak demand for forestry equipment and the uncertain market situation	Negative	Neutral	Negative

ror, random character insertion, OCR replacement, random word deletion and antonym replacement under varying noise levels at 10%, 20%, 30%, 40% and 50%, as well as on the Kaggle stock market tweets, which is a real-world noisy dataset. We show that fine-tuning is more robust to noise than prefix tuning in most of the corruption methods. As the impact of noise is more significant along with increasing noise levels, prefix tuning shows a more significant decrease in performance compared to full fine-tuning. The variance of performance of prefix tuning is higher than that of fine-tuning for most corruption setups. Our study suggests that caution should be taken by practitioners when applying prefix tuning to noisy data. A solution to improving the robustness to reduce the impact of noise is desired and is our immediate future work.

6 Limitations

The words were randomly corrupted in a sentence with no emphasis on the word's context and no experiments were carried out to find out the importance of the corrupted word in the context of predicting the sentiment. Corrupting an important word may result in an increased drop in performance than corrupting a word which has minimal impact on the sentiment of a sentence. Sun et al. (2020) have found that typos on informative words affect the performance of the BERT to a greater extent than typos in other words. Furthermore, the robustness was evaluated on the sentiment analysis task and it was not evaluated on other natural language processing tasks like question answering, named entity recognition and text summarization.

References

Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.

Carlos Aspillaga, Andrés Carvallo, and Vladimir Araujo. 2020. Stress test evaluation of transformer-based models in natural language understanding tasks. *arXiv preprint arXiv:2002.06261*.

Yash Chaudhary. 2020. Stock-market sentiment dataset. https://www.kaggle.com/datasets/yash612/stockmarket-sentiment-dataset.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jeremy Howard and Sebastian Ruder. 2018. Universal

language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.

Moreno La Quatra and Luca Cagliero. 2020. End-to-end training for financial report summarization. In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pages 118–123.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Xiaodong Li, Haoran Xie, Li Chen, Jianping Wang, and Xiaotie Deng. 2014. News impact on stock price return via sentiment analysis. *Knowledge-Based Systems*, 69:14–23.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Tim Loughran and Bill McDonald. 2016. Textual analysis in accounting and finance: A survey. *Journal of Accounting Research*, 54(4):1187–1230.

Edward Ma. 2019. Nlp augmentation. https://github.com/makcedward/nlpaug.

Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65(4):782–796.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Barbara Rychalska, Dominika Basaj, Alicja Gosiewska, and Przemysław Biecek. 2019. Models in the wild: On corruption robustness of neural nlp systems. In *International Conference on Neural Information Processing*, pages 235–247. Springer.

Matheus Gomes Sousa, Kenzo Sakiyama, Lucas de Souza Rodrigues, Pedro Henrique Moraes, Eraldo Rezende Fernandes, and Edson Takashi Matsubara. 2019. Bert for stock market sentiment analysis. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1597–1601. IEEE.

Ankit Srivastava, Piyush Makhija, and Anuj Gupta. 2020. Noisy text data: Achilles' heel of bert. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 16–21.

Lichao Sun, Kazuma Hashimoto, Wenpeng Yin, Akari Asai, Jia Li, Philip Yu, and Caiming Xiong. 2020. Adv-bert: Bert is not robust on misspellings! generating nature adversarial samples on bert. *arXiv preprint arXiv:2003.04985*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Zonghan Yang and Yang Liu. 2022. On robust prefix-tuning for text classification. *arXiv preprint arXiv:2203.10378*.

Shi Yu, Yuxin Chen, and Hussain Zaidi. 2020. A financial service chatbot based on deep bidirectional transformers. *arXiv preprint arXiv:2003.04987*.

Lingyun Zhao, Lin Li, Xinhao Zheng, and Jianwei Zhang. 2021. A bert based sentiment analysis and key entity detection approach for online financial texts. In *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 1233–1238. IEEE.

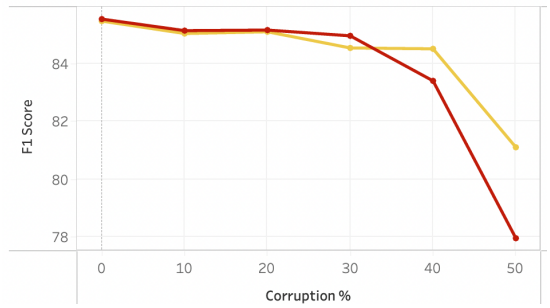
A Appendix

A.1 Implementation Details

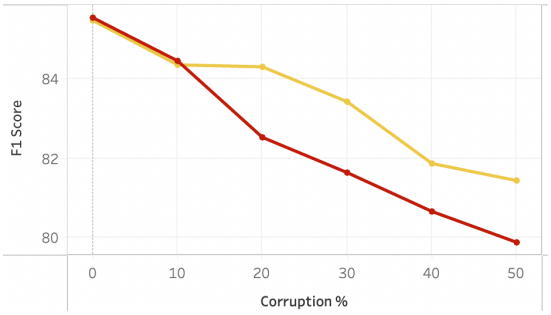
The experiments were carried out on four Nvidia GeForce RTX 2080 GPU's for 30 epochs. The length of the prefix plays a significant role in prefix tuning. In (Liu et al., 2021), the authors have suggested that Natural Language Understanding (NLU) tasks prefer shorter prefix lengths and they have used a prefix length of 20 for sentiment classification to obtain the best performance. We have also used a prefix length of 20 to evaluate the performance of the models. The learning rate differs for each model and method. For prefix tuning, both BERT-base and RoBERTa-large models use a learning rate of 1e-2. For fine-tuning, BERT-base uses

a learning rate of 2e-5 and RoBERTa-large used a learning rate of 2e-6. Furthermore, the 50% noise level is selected for all the corruption methods and the variance is measured for both prefix tuning and fine-tuning for the BERT base model. The Kaggle Stock Market tweets dataset is also used to evaluate the performance of prefix tuning and fine-tuning on real-world noisy data (tweets) with the same set of hyperparameters as the financial phrasebank dataset.

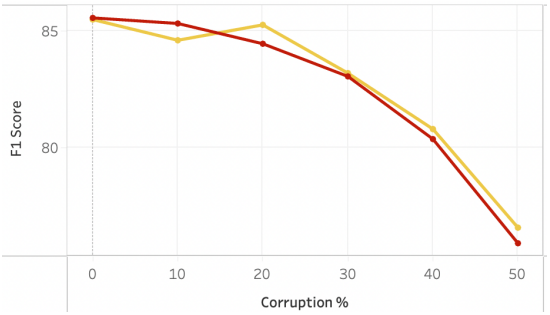
A.2 Experimental Results - Visualizations



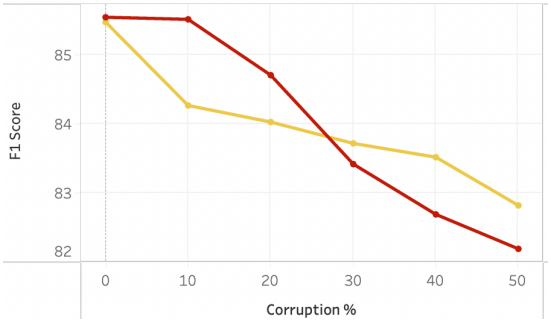
(a) Keyboard Error



(b) Random Char. Insertion

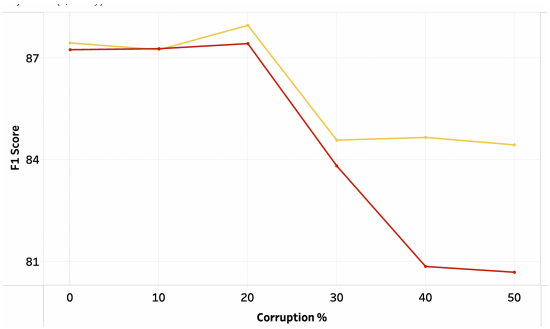


(c) OCR Replacement

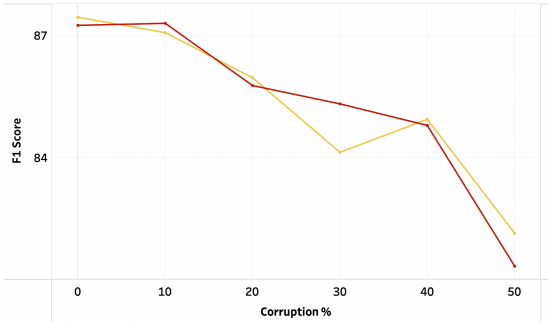


(d) Random word Deletion

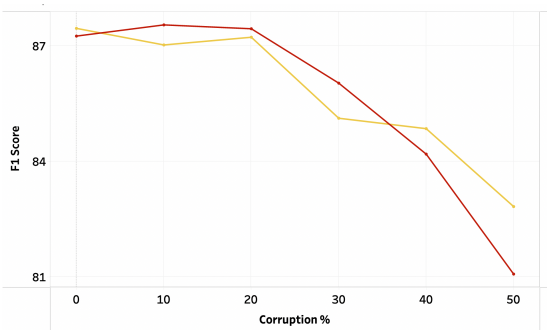
Figure 2: Plot of F1 scores of BERT-base model for various corruption methods. Red line represents prefix tuning and yellow line represents fine-tuning.



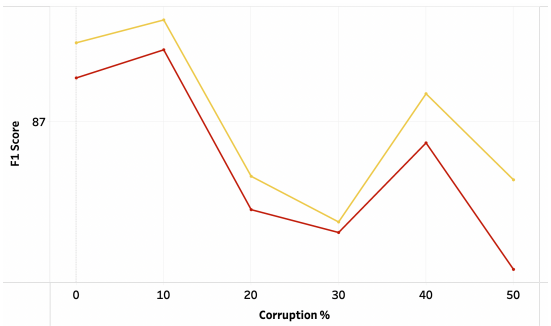
(a) Keyboard Error



(b) Random Char. Insertion



(c) OCR Replacement



(d) Random word Deletion

Figure 3: Plot of F1 scores of RoBERTa-large model for various corruption methods. Red line represents prefix tuning and yellow line represents fine-tuning.