# Sentiment Analysis and Experimenting on Domain shifting

Stefanos Konstantinou, Angelos Zinonos

October 27, 2023

**Abstract**

Domain adaptation is widely acknowledged as a challenging endeavor in the field of Natural Language Processing (NLP), particularly due to the scarcity of large datasets and the potential for a domain's distinct linguistic features to detrimentally impact model performance. Given the rapid advancements in NLP, new techniques are continually emerging, some of which have not undergone comprehensive testing. In our study, we conducted experiments using a small dataset to assess the effectiveness of three fine-tuning techniques in the task of sentiment analysis.

## 1 Introduction

This is an era where natural language processing(NLP) is constantly evolving a new techniques are being tested all the time to determine the best performing one. The following study, "Attention is all you need" [5], created a new path in the NLP field, as it revolutionised how embeddings are used in NLP tasks. The transformer architecture has emerged as a fundamental framework for numerous natural language processing (NLP) tasks, encompassing machine translation, text generation, and various others. By employing a "pre-training" approach, which involves training a model through language modeling tasks (like "Masked Language Modeling"), transformer architectures have enabled the acquisition of contextualised feature representation of words. This means that they can capture the changing meanings of the same word in diverse contexts. With these contextualized embeddings, particularly in encoder architectures, we have the flexibility to introduce different heads. By leveraging the pre-trained parameters of the encoder, we can interchange and fine-tune the heads to suit specific tasks as needed.

A drawback to say for this we could say, is that transformer based models require a relatively substantial amount data for the purpose of finetuning, which is something that is not always available. Furthermore, these models typically possess a broad understanding of general language and aren't tailored to specific attributes of domains such as Twitter or Reddit. This can present additional challenges when it comes to finetuning these models for tasks involving these complex domains, particularly when the dataset is limited in size.

In this project we will explore the efficacy of using various finetuning methods of a pretrained large language model. This will be done in the context of finetuning a RoBERTa (base size) model on datasets that contain unique linguistic characteristics that can pose difficulties (such as language ambiguities) for the models to adapt to the task of sentiment analysis. Finetuning methods including: the standard fine-tuning method (or as we will refer to: SFiT), Adapters, and "Head First FineTuning" (HeFiT). More details of the setup will be discussed in the Methodology section.

## 2 Dataset

To conduct our comprehensive exploration experiments, we selected two datasets characterized by distinct linguistic attributes within their respective domains. These datasets are sourced from the Twitter and Reddit platforms [1]. Both Twitter and Reddit are open

platforms where users from diverse backgrounds can readily create accounts and participate in discussions, fostering unstructured dialogues. Notably, these platforms allow users to post and comment without extensive user background checks, facilitating open and free participation.

Indeed, these datasets present significant challenges for models when it comes to learning a task. The data within these platforms is characterized by linguistic peculiarities, including slang, unorthodox syntax, and incomplete phrasing. Consequently, this creates a heightened level of difficulty for natural language processing tasks, as the contextual embeddings coming from pre-trained large language models inherently capture an understanding of generally correct language. As a result, during the fine-tuning phase, the model must also adapt to the specific domain of the dataset, or undergo what is commonly referred to as domain shift. In practical terms, this increased linguistic complexity makes it more challenging for models to achieve high performance metrics, in our case, in the context of a sentiment analysis task, when compared to working with more conventional language data.

To clarify, the one dataset consists of only Tweets from the Twitter platform and the other from the Reddit platform which consists of Comments. The sentiment labels range from -1 to 1 with -1 indicating a "Negative" sentiment, 0 indicating "Neutral" sentiment and 1 indicating a "Positive" sentiment. In short, both datasets consist of 3 classes.

For our experiments, we curated two distinct subsets from each dataset. The first subset partition comprises 400 data records per class for training and an additional 100 records per class for evaluation, resulting in a total of 1,500 data records. In the second dataset subset, each class includes 1,000 records for training, with an additional 250 records for evaluation, totaling 3,750 data records.

Both of these subsets have intentionally limited dataset sizes to assess the effectiveness of the fine-tuning techniques when dealing with domain shift in scenarios where data availability is constrained. The second subset, being deliberately larger than the first, serves to evaluate the techniques as dataset size increases, providing valuable insights in both directions.

Moreover, it is a common practice when working with transformer models not to extensively preprocess the data but to utilize the model's built-in tokenizer and feed the data directly to the model. Transformer-based models typically possess a strong understanding of language and a sufficiently extensive vocabulary, enabling them to handle noise and capture crucial information, even in the presence of idiosyncrasies.

Here's a small look at a few samples of the datasets.

| Text | Class |
|---|---|
| when modi promised "minimum government maximum governance" expected him begin the difficult job refo | -1 |
| talk all the nonsense and continue all the drama will vote for modi | 0 |
| what did just say vote for modi welcome bjp told you rahul the main campaigner for modi think modi | 1 |

Table 1: Twitter Data sample from each class

| Text | Class |
|---|---|
| does evil include the lady pai chunked | -1 |
| jesus was zen meets jew | 0 |
| unto others you would have them unto you would good start | 1 |

Table 2: Reddit Data sample from each class

## 3 Methodology

The model we have chosen is RoBERTa (base size) for several reasons[3]. First and foremost, it is pre-trained on a massive corpus of text data, which enables it to capture a deep understanding of language, syntax, and semantics. Additionally, Roberta employs

a bidirectional architecture, which means it considers context from both directions in a sentence, enhancing its grasp of intricate linguistic relationships. Moreover Roberta consistently yielded impressive results for various applications specifically sentiment analysis. As for the finetuning techniques, here are the techniques tested:

- **Standard finetuning (SFiT):** This is a fundamental and widely employed technique in NLP that involves harnessing pre-trained language models, which have acquired a broad spectrum of linguistic features and general knowledge, and customizing them for particular NLP tasks. The process entails adding a task-specific head layer on top of the pre-trained model (with initialized parameters) and subsequently training the model as a whole, end to end.
- **Adapters:** Adapters are particularly effective for tasks involving small datasets, primarily because of their limited number of trainable parameters. They offer an efficient means of tailoring pre-trained models to specific tasks, as mentioned in [2]. When dealing with a scarcity of training data, conventional fine-tuning methods may run the risk of overfitting and delivering degraded performance on novel, unseen data. Adapters are essentially small slots composed of linear layers with a minimal number of trainable parameters, seamlessly integrated within the model's layers.
- **Head First FineTuning (HeFiT):** It is a fine-tuning method that was tested in the context of a regression task using a limited amount of Twitter data [4]. Unlike the standard fine-tuning method that involves updating the entire model end-to-end right from the beginning, HeFiT adopts a different approach. It prioritizes the adaptation of the model's task-specific head layer initially. The rationale behind this approach is to preserve the contextualized representation of the encoder's parameters until the head layers begin to learn the task. Only after this initial adaptation does the model undergo training end-to-end. This strategy mitigates the risk of strong erroneous update signals propagating during training (due to the linguistic peculiarities of the domain), which could potentially degrade the representation within the encoder's parameters.

Having chosen our methods, we construct a hyper parameter search to find the optimal parameters. For SFiT we experimented with epochs from 3 to 6, with adapters we experimented with 45, 50, 55, 60. And for HeFiT we kept the same hyperparameters as they were outlined in the original research study; 3 training epochs with frozen encoder layers (only the head is trainable) and then 6 epochs end to end training with half the learning rate as before. Also, the maximum input sequence length for the Twitter data was: 128; and for Reddit: 256 due to the platforms' text limit rules.

It very is important to note that **each configuration is run 5 times** and the end scores will be averaged and illustrated in the Results and Discussion section.

## 4 Results and Discussion

In this section, the results of the experiments are to be illustrated and discussed. The results include the averaged (over 5 runs) results of Accuracy, F1, Precision and Recall (0 being the worst and 1.0 being the best possible result) and the standard deviation of the F1 results. Important to note, for the standard deviation of F1 results, unlike the rest, the lower the value, the better the result. Tables 3 & 4 show the results for the small datasets of Twitter and Reddit (1500 data overall) respectively and Tables 5 & 6 show the results for the bigger datasets of Twitter and Reddit (3750 data overall) respectively.

In all datasets and dataset sizes, SFiT consistently demonstrated superior results across all metrics. Notably, for the smaller dataset, the optimal performance was achieved by employing 4 or 5 training epochs, while for the larger dataset, the most effective approach was to use 6 training epochs.

Conversely, when assessing the outcomes with smaller datasets, SFiT exhibits a greater standard deviation in the F1 scores. A higher deviation in F1 results indicates that, when dealing with smaller datasets, the training process can become less stable, leading to significant performance fluctuations across different runs.

Dataset size: 1500 Twitter Data

| | ACC | F1 | F1 std | Precision | Recall |
|---|---|---|---|---|---|
| SFiT 3ep | 0.6973 | 0.6974 | 0.0431 | 0.6973 | 0.7003 |
| SFiT 4ep | 0.7586 | 0.7589 | 0.0294 | 0.7587 | 0.7640 |
| SFiT 5ep | 0.7473 | 0.7485 | 0.0398 | 0.7473 | 0.7559 |
| SFiT 6ep | 0.6587 | 0.6521 | 0.1167 | 0.6587 | 0.6738 |
| Adapters 45ep | 0.6807 | 0.6804 | 0.0131 | 0.6807 | 0.6852 |
| Adapters 50ep | 0.6887 | 0.6887 | 0.0316 | 0.6887 | 0.6920 |
| Adapters 55ep | 0.6693 | 0.6698 | 0.0365 | 0.6693 | 0.6762 |
| Adapters 60ep | 0.6613 | 0.6608 | 0.0276 | 0.6613 | 0.6645 |
| HeFiT | 0.6787 | 0.6790 | 0.0273 | 0.6787 | 0.6864 |

Table 3: Averaged metric scores on the evaluation dataset of the Twitter data. Datsaset size: 1500. 'F1 std' is the standard deviation over all F1 results. Underlined score indicates the best score of the column.

Table 4: Averaged metric scores on the evaluation dataset of the Reddit data. Datsaset size: 1500. 'F1 std' is the standard deviation over all F1 results. Underlined score indicates the best score of the column.

Dataset size: 1500 Reddit Data

| | ACC | F1 | F1 std | Precision | Recall |
|---|---|---|---|---|---|
| SFiT 3ep | 0.6293 | 0.6293 | 0.0456 | 0.6293 | 0.6444 |
| SFiT 4ep | 0.6787 | 0.6782 | 0.0215 | 0.6787 | 0.6987 |
| SFiT 5ep | 0.6920 | 0.6916 | 0.0168 | 0.6920 | 0.7068 |
| SFiT 6ep | 0.6727 | 0.6746 | 0.0359 | 0.6727 | 0.6984 |
| Adapters 45ep | 0.6180 | 0.6170 | 0.0181 | 0.6180 | 0.6299 |
| Adapters 50ep | 0.6060 | 0.6051 | 0.0155 | 0.6060 | 0.6160 |
| Adapters 55ep | 0.6193 | 0.6184 | 0.0125 | 0.6193 | 0.6253 |
| Adapters 60ep | 0.6200 | 0.6195 | 0.0010 | 0.6200 | 0.6987 |
| HeFiT | 0.6627 | 0.6631 | 0.0236 | 0.6627 | 0.6742 |

Moreover, the training process using adapters appeared to exhibit greater stability, with consistently low standard deviation. However, when examining the other performance metrics, it became evident that adapters were unable to match the effectiveness of the other methods. This discrepancy could be attributed to the fact that the encoder layers were frozen, which meant that the model's language understanding capabilities were not adjusted to the peculiar linguistic attributes of either of the two domains (Twitter or Reddit). It appears that the linear layer adapter slots (which were trainable) within the encoder's layers were insufficient for this purpose.

Dataset size: 3750 Twitter Data

| | ACC | F1 | F1 std | Precision | Recall |
|---|---|---|---|---|---|
| SFiT 3ep | 0.8056 | 0.8059 | 0.0054 | 0.8056 | 0.8121 |
| SFiT 4ep | 0.7800 | 0.7796 | 0.0277 | 0.7800 | 0.7880 |
| SFiT 5ep | 0.8085 | 0.8090 | 0.0092 | 0.8085 | 0.8177 |
| SFiT 6ep | 0.8173 | 0.8181 | 0.0056 | 0.8181 | 0.8173 |
| Adapters 45ep | 0.7685 | 0.7683 | 0.0140 | 0.7685 | 0.7787 |
| Adapters 50ep | 0.7709 | 0.7711 | 0.0154 | 0.7709 | 0.7821 |
| Adapters 55ep | 0.7637 | 0.7636 | 0.0074 | 0.7637 | 0.7740 |
| Adapters 60ep | 0.7752 | 0.7753 | 0.0074 | 0.7652 | 0.7880 |
| HeFiT | 0.7915 | 0.7915 | 0.0164 | 0.7915 | 0.8002 |

Table 5: Averaged metric scores on the evaluation dataset of the Twitter data. Datsaset size: 3750. 'F1 std' is the standard deviation over all F1 results. Underlined score indicates the best score of the column.

As anticipated, having a larger dataset generally resulted in improved model performance. In the case of the larger dataset, SFiT not only demonstrated superior performance but also showcased a more stable training process. This implies that the standard fine-tuning technique is better suited for training large language models when substantial data is available. However, it's important to note that even though HeFiT didn't consistently achieve the top performance in terms of overall results, it consistently performed closely to the best-performing approach while maintaining a lower standard deviation (of F1). This

highlights the adaptability of the HeFiT fine-tuning technique in seamlessly transitioning to a new domain. It accomplishes this by initially preparing the task-specific head layers without propagating a strong erroneous update signal, which could potentially lead to representation degradation within the encoder's parameters.

Table 6: Averaged metric scores on the evaluation dataset of the Reddit data. Datsaset size: 3750. 'F1 std' is the standard deviation over all F1 results. Underlined score indicates the best score of the column.

| Dataset size: 3750 Reddit Data | | | | | |
|---|---|---|---|---|---|
| | ACC | F1 | F1 std | Precision | Recall |
| SFiT 3ep | 0.8189 | 0.8189 | 0.0324 | 0.8189 | 0.8202 |
| SFiT 4ep | 0.8496 | 0.8497 | 0.0137 | 0.8496 | 0.8512 |
| SFiT 5ep | 0.8442 | 0.8446 | 0.0122 | 0.8442 | 0.8476 |
| SFiT 6ep | 0.8507 | 0.8510 | 0.0079 | 0.8506 | 0.8531 |
| Adapters 45ep | 0.8114 | 0.8118 | 0.0080 | 0.8115 | 0.8141 |
| Adapters 50ep | 0.8191 | 0.8197 | 0.0115 | 0.8192 | 0.8224 |
| Adapters 55ep | 0.8094 | 0.8098 | 0.0159 | 0.8094 | 0.8115 |
| Adapters 60ep | 0.8144 | 0.8150 | 0.0093 | 0.8144 | 0.8169 |
| HeFiT | 0.8325 | 0.8331 | 0.0069 | 0.8325 | 0.8350 |

An important point to note is that in the case of SFiT, a hyperparameter search was conducted, specifically focusing on the number of training epochs. This search highlighted the significance of the epoch count, particularly when dealing with smaller datasets, as the performance varied vastly. On the contrary, for HeFiT, no hyperparameter search was performed; instead, the hyperparameters were kept consistent with those outlined in its original research article.

## 5 Conclusion

In conclusion, our study involved testing various fine-tuning techniques on two datasets, Twitter and Reddit, both of which exhibit their unique linguistic characteristics. These peculiarities can pose challenges for models when adapting to tasks due to the inherent ambiguity in language. Our analysis incorporated a range of metrics, revealing that standard fine-tuning (SFiT), especially with careful consideration of hyperparameters like the number of training epochs, can yield superior performance. However, it is most effective when applied to larger datasets, as it may introduce training instability in smaller datasets. Additionally, while adapters have fewer trainable parameters, they still fall short in terms of adapting the language representation of encoder layers to the specific linguistic peculiarities of the data domain, rendering them less capable for domain adaptation. Finally, the HeFiT fine-tuning technique emerged as a more stable alternative, well-suited for scenarios with a smaller dataset, although it does not achieve the same level of performance.

## 6 Future work

Regarding potential directions for future research, it would be intriguing to explore whether training the adapter's parameters on a language modeling task could yield improved results on the target task, especially when dealing with a limited dataset, as it is anticipated to enhance adaptability. Furthermore, when it comes to the HeFiT technique, it would be valuable to conduct an extensive hyperparameter search in the context of a classification task. This is especially relevant as HeFiT was initially introduced and studied within the context of a regression task.

## 7 Github repository

Here you will find the code, the datasets used and the results. https://github.com/acd17sk/NLP-Domain-Adaptation-Exploration

## References


[1] Charan Gowda, Anirudh, Akshay Pai, and Chaithanya kumar A. Twitter and reddit sentimental analysis dataset, https://www.kaggle.com/datasets/cosmos98/twitter-and-reddit-sentimental-analysis-dataset.

[2] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp, 2019.

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

[4] Andrianos Michail, Stefanos Konstantinou, and Simon Clematide. UZH_CLyp at SemEval-2023 task 9: Head-first fine-tuning and ChatGPT data generation for cross-lingual learning in tweet intimacy prediction. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1021–1029, Toronto, Canada, July 2023. Association for Computational Linguistics.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.