UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
MATEMATICA
DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA"

LLM Report

Fine Tuning of MiniLM for Lie Detection Task

Authors: Cveevska Marija, Erim Suleyman, Karakus Isikay,Varagnolo Mattia
Supervisor: Prof.Giuseppe Sartori, Dr.Merylin Monaro

20/12/2023

1 Abstract

Humans are not very good at telling when someone is lying. In the realm of deception detection, human accuracy is constrained, barely surpassing chance levels. To address this limitation, a surge of automated verbal lie detection methodologies has emerged, leveraging the power of Machine Learning and Transformer models. This study delves into the fine-tuning of MiniLM Language Models, exploring their efficacy in discerning deceit. By harnessing the capabilities of these advanced models, we aim to push the boundaries of accuracy, seeking to transcend the inherent limitations of human lie detection capabilities. We will compare fine-tuning methods and also do different scenarios concerning the data set.

2 Objectives

The main objective of this project is to delve into LLMs fine-tuning and compare different methods and different scenarios for Lie Detection. LLMs, known as Transformer language models, pack in hundreds of millions of parameters and are trained on vast collections of texts (called corpora) in their initial phase. This pre-training helps LLMs understand the intricate structures and patterns of language, getting a good grip on how words fit together in sentences, what they mean, and how people use them.

Once they have gone through this pre-training phase, these models can be tuned further for specific jobs using smaller sets of data. They are really good at many tasks like translating languages, figuring out sentiments in texts, answering questions, summarizing texts, and even generating code. That is why LLMs are great for various language tasks instead of just one.

In our study, we aimed to assess the ability of one of these LLMs, specifically MiniLM (a model created by Microsoft researchers and available on HuggingFace), to determine the veracity of short stories. We achieved this by fine-tuning MiniLM using a dataset focused on intentions, referred to as the Intention dataset. We employed two distinct fine-tuning methods: Transfer Learning and Parameter Efficient Fine Tuning (PEFT) with LoRA.

3 MiniLM

MiniLM represents a task-agnostic, compact pre-trained transformer model, as detailed in the paper "MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers." This model provides a range of small and efficient pre-trained variants designed for language understanding and generation, encompassing both cased and uncased versions, as well as multilingual and English pre-trained iterations.

In its multilingual variant, the Multilingual-MiniLM-L12-H384 model seamlessly integrates XLM-R's tokenizer compatibility with BERT's Transformer architecture. This model, supporting 16 languages, is equipped with 21 million Transformer parameters and 96 million embedding parameters. It demonstrates robust and competitive performance on cross-lingual tasks, including benchmarks such as XNLI and MLQA. In our project, we have chosen to utilize this specific version of MiniLM for its multilingual capabilities and impressive task performance.

For English tasks, MiniLM offers several pre-trained models, including MiniLMv1-L12-H384, MiniLMv1-L6-H384, and MiniLMv1-L6-H768. Distilled from an in-house pre-trained UniLM v2 model in BERT-Base size, these models use the same WordPiece vocabulary as BERT.

The ensuing table, Table 1 delineates a comparative analysis between BERT and GPT-3.5, highlighting key facets such as Model Architecture, Token Size, Model Size, Functions, and Use Cases.

Table 1: Comparison of BERT and GPT-3.5

Feature	*BERT*	*GPT-3.5*
Model Architecture	Bidirectional encoder representations, ”encoder only” transformer architecture.	Based on unidirectional transformer architecture with autoregressive training, ”decoder -only” design for diverse language generation tasks. Also, additional fine-tuning process RLHF (Reinforcement Learning with Human Feedback) added into the GPT-3 model.
Token Size	WordPiece vocabulary max length of 512 subword tokens.	Max of 4,097 tokens (GPT-3.5 Turbo).
Model Size	Two models, 110M and 340M parameters.	Three models, 1.3B, 6B, and 175B parameters.
Functions	Specifically designed for tasks demanding context-aware representations. BERT addresses the challenge of interpreting ambiguous language in text by leveraging surrounding context. Pre-trained on Wikipedia text, the BERT framework is adaptable and can be fine-tuned using Q&A datasets.	GPT-3.5 model is a refined version of GPT-3, designed for autoregressive language generation and demonstrates proficiency in various language tasks without requiring task-specific pretraining.
Use Cases	Natural language generation.	Natural language or code generation.

4 Dataset

The Intention dataset (Future Intentions), consisting of 1640 statements, focuses on participants' most significant nonwork-related activities, collected through recruitment on Prolific Academic. Participants were instructed to provide convincing answers to two specific questions:

1. Q1: ”Please describe your activity as specifically as possible.”
2. Q2: ”Which information can you provide to reassure us that you are telling the truth?”

Participants were required to describe activities that were not continuous or daily, taking place beyond the next seven days, with clearly defined start and end times. They were randomly assigned to either the truthful or deceptive condition. In the deceptive condition, participants were given matched activities from the truthful condition. The dataset includes six columns:

1. signevent: Significance of the nonwork-related activity.
2. q1: Explanation for Q1.
3. q2: Answer for Q2.
4. unid: Unidentified column.
5. id: Number assigned to each statement.
6. outcome class: Indicates whether the statement is truthful (T) or deceptive (F).

The dataset comprises 857 deceptive and 783 truthful statements, each with two answers per participant.

5 Scenarios

In the context of our research, we designed two distinct scenarios to rigorously test our hypotheses—leveraging transfer learning and implementing Parameter- Efficient Fine-Tuning (PEFT) with LoRA (Low-Rank Adaptation). These scenarios aim to assess the model's ability to discern and respond effectively to linguistic nuances, contributing valuable insights into intention lie detection. Furthermore, we tested whether model performance may depend on model sizes. Therefore, we first fine-tuned the small-sized version of MiniLM in every scenario, and then we repeated the same experiments in every scenario with the full-sized version.

Finally we tested both scenarios on Q1, Q2 and Q1 + Q2 columns.

5.1 Scenario 1: Transfer Learning

Transfer learning is a methodology that involves training a model initially on one task and subsequently applying the acquired knowledge to a distinct yet related task. Essentially, the model leverages its pre-existing knowledge to enhance efficiency in learning new tasks. In the context of fine-tuning MiniLM, we incorporate transfer learning through a structured process encompassing three key steps: the pre-training phase, the fine-tuning phase, and the inference phase. This approach allows us to analyze expressions in our dataset.

The outcomes derived from employing various hyperparameters and the results after the application of transfer learning are comprehensively documented in Table 2 and Table 3, respectively. These tables serve as valuable resources for understanding the effectiveness of different configurations in the context of fine-tuning MiniLM with transfer learning for the Intention Dataset.

5.2 Scenario 2: Parameter-Efficient Fine-Tuning (PEFT) with LoRA

MiniLM underwent fine-tuning using Parameter-Efficient Fine-Tuning (PEFT) with Low-Rank Adaptation (LoRA). Our strategy involves fine-tuning only a small subset of additional model parameters while keeping the majority of the pre-trained MiniLM parameters frozen. This approach effectively mitigates issues such as catastrophic forgetting observed during full fine-tuning. The PEFT methodology facilitates the generation of compact checkpoints, each worth only a few megabytes for every downstream dataset. These smaller trained weights obtained through PEFT seamlessly integrate into the pre-trained MiniLM, enabling the utilization of the same model for multiple tasks.

LoRA is a reparametrization method that aims to reduce the number of trainable parameters through low-rank representations. The weight matrix is decomposed into low-rank matrices, which are then trained and updated independently. All pre-trained model parameters remain frozen during this process. After training, the low-rank matrices are reintegrated with the original weights. This approach enhances the efficiency of storing and training a LoRA model, as it involves significantly fewer parameters. Table 2 displays the hyperparameters tested, and Table 3 presents the corresponding outcomes obtained through the utilization of Transfer Learning and Table 4 presents the corresponding outcomes obtained through the utilization of LoRA.

Parameter	Value
Train-Val-Test split	0.9 - 0.05 - 0.05
LoRA parameters	$r = 16$, $lora\_alpha = 16$ $lora\_dropout = 0.1$, $bias = "all"$
Learning rate LoRa	0.001
Learning rate TL	0.00002
Weight decay	0.01
Optimizer	"adamw_torch"
All parameters of model	117,950,212
Trainable parameters	347,138
Trainable %	0.2943

Table 2: Experiment Parameters for Scenario 2 and Scenario 1

6 Results

Fine-tuning large language models (LLMs) involves adapting a pre-trained language model to a specific task through additional training on task-specific data, enhancing its ability to generate contextually relevant and coherent text aligned with task objectives. We applied this process to both the small and Full versions of MiniLM, utilizing intention datasets and following the outlined experimental setup.

Our primary objective was to address the lie-detection task as a binary classification challenge, where the dataset comprised raw texts associated with binary labels, categorizing instances as either truthful or deceptive. By fine-tuning MiniLM, we transformed the model into a specialized tool for binary classification tasks.

In this section, we present the accuracy results obtained from Transfer learning and Parameter-Efficient Fine-Tuning (PEFT) with Low-Rank Adaptation (LoRA). The evaluation encompasses both sections Q1 and Q2 independently and along with their unified form is comprehensively documented in Table 3 and Table 4.

6.1 Results: Transfer Learning

The results of the transfer learning experiments, as illustrated in Table 3, showcase the performance of our model across various configurations and hyperparameters. The dataset comprises different question types (q1, q2, q1+q2) within the context of the Intention Dataset. Notably, fine-tuning the MiniLM model using transfer learning for 30 epochs yields promising results. The validation accuracy (Val Acc) and test accuracy (Test Acc) metrics provide insights into the model's ability to generalize to unseen data. The experiments involve freezing different layers during training, ranging from 0 to 9 layers. Interestingly, we observe that freezing 4 layers, particularly in configurations involving q1 and q2, leads to improved accuracy. These results contribute valuable insights into the effectiveness of transfer learning and the impact of hyperparameter choices on the performance of our model in the specific context of the Intention Dataset.

Dataset	Question type	Epochs	Val Acc	Test Acc	Layers frozen
Full	q1+q2	30	77.50%	70.00%	4
Full	q2	30	57%	59.76%	4
Full	q1	30	70%	69.51%	4
Full	q1	30	62%	65.85%	3
Full	q2	30	65.85%	64.63%	3
Full	q1+q2	30	64.63%	62.20%	3
Full	q1	30	62.20%	64.64%	9
Full	q2	30	52.44%	53.66%	9
Full	q1+q2	30	64.63%	67.07%	9
Full	q1	30	66%	67.07%	0
Full	q2	30	62%	59.75%	0
Full	q3	30	59.75%	67.07%	0

Table 3: Transfer Learning Results

6.2 Results: Parameter-Efficient Fine-Tuning (PEFT) with LoRA

The results of Parameter-Efficient Fine-Tuning (PEFT) with LoRA, as detailed in Table 4, provide valuable insights into the impact of various hyperparameters on the fine-tuning process. In this series of experiments, we explored different configurations involving question types (q1, q2, q1+q2) across the Intention dataset, including the entire dataset and a subset of 800 samples. Notably, the table reveals the model's performance under different epoch settings (50, 30, and 10 epochs). The application of LoRA, a reparametrization method, proves to be effective in reducing the number of trainable parameters, resulting in a more efficient model with enhanced generalization capabilities. For instance, when fine-tuning the entire dataset with q1+q2 questions for 30 epochs, the model achieves a test accuracy of 59.76%. Additionally, variations in sample size and question types showcase the adaptability of the PEFT with the LoRA approach. An interesting outcome emerges in the case of fine-tuning 800 samples with q1 questions for 30 epochs. This configuration yields the most impressive results, with a remarkable test accuracy of 65.00% and a validation accuracy of 75.00%. This stands out as the best-performing setting within the Parameter-Efficient Fine-Tuning (PEFT) with LoRA approach. These findings underscore the importance of hyperparameter tuning and the utilization of parameter-efficient techniques in achieving optimal performance for the given task of intention recognition. With enough input in the cases of full dataset or in q1 + q2 the convergence of training happened in 10 epochs. Further training didn't affect the results. But in the cases of 800 samples with only q1 or q2 with increased number of epochs also the accuracy increased until some point. That is why hyperparameter choice is very important in fine tuning.

Dataset	Question Type	Epochs	Val Acc	Test Acc
Full	q1+q2	50	56.098%	59.76%
Full	q1+q2	30	56.098%	59.76%
Full	q1+q2	10	56.098%	59.76%
Full	q2	50	56.098%	59.76%
Full	q2	30	56.098%	59.76%
Full	q2	10	56.098%	59.76%
Full	q1	50	56.098%	59.76%
Full	q1	30	56.098%	54.88%
Full	q1	10	56.098%	59.76%
800 samples	q1+q2	50	52.500%	57.50%
800 samples	q1+q2	30	52.500%	57.50%
800 samples	q1+q2	10	52.500%	57.50%
800 samples	q2	50	52.500%	57.50%
800 samples	q2	30	70.000%	65.00%
800 samples	q2	10	52.500%	57.50%
800 samples	q1	50	67.500%	52.50%
800 samples	q1	30	75.000%	65.00%
800 samples	q1	10	52.500%	57.50%

Table 4: PEFT with LoRA Results

7 Conclusions

In conclusion, our investigation into fine-tuning large language models (LLMs), specifically MiniLM, for the lie-detection task has yielded valuable insights. Through the application of transfer learning and Parameter-Efficient Fine-Tuning (PEFT) with Low-Rank Adaptation (LoRA), MiniLM was adapted into a specialized tool for binary classification tasks. The comprehensive results presented in Tables 3 and 4 offer a detailed overview of the model's performance across various configurations, hyperparameters, and datasets.

The transfer learning experiments, outlined in Table 3, underscore the effectiveness of freezing specific layers, particularly evident in configurations involving q1, q2 and q1+q2 questions. Notably, freezing 4 layers during training led to the best accuracy in our task.

Turning our attention to Table 4, which details the results of PEFT with LoRA, we observe a nuanced interplay between hyperparameters and model performance. The adaptability of MiniLM across different question types (q1, q2, q1+q2) and datasets (Full and 800 samples) is evident. Of particular interest is the standout performance in fine-tuning 800 samples with q1 questions for 30 epochs, achieving a remarkable test accuracy of 65.00% and a validation accuracy of 75.00%. This setting emerges as the top-performing configuration within the PEFT with LoRA approach and also the best accuracy for q1 in our task.

Despite our diligent efforts in hyperparameter tuning and model selection, we acknowledge that our achieved maximum accuracy of 70% may not meet the desired threshold for optimal performance. To achieve further improvements in model performance, experiments can be conducted with a larger number of epochs along with the search for superior optimization algorithms and improved learning rates. This can enable additional insights from the data to optimize convergence and strike a better balance between training efficiency and accuracy.

These findings not only emphasize the importance of hyperparameter tuning but also highlight the performance of LoRA in reducing trainable parameters and enhancing model efficiency. The success of our approach in tailoring language models for intention recognition tasks demonstrates the potential for real-world applications in lie detection.

Moving forward, these results will guide our ongoing efforts to further refine language models for heightened performance in real-world lie-detection scenarios. The combination of transfer learning and PEFT with LoRA presents a promising avenue for advancing the capabilities of large language models in nuanced classification tasks.

References

[1] Loconte, Riccardo, et al. ”Verbal Lie Detection using Large Language Models.” arXiv preprint, 2023. https://doi.org/10.21203/rs.3.rs-3126100/v1

[2] Wang, Weizhe, et al. , ”MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers.” arXiv preprint, 2020. https://arxiv.org/abs/2002.10957

[3] Kleinberg, Bennett, and Verschuere, Bruno. , ”How humans impair automated deception detection performance.”, Acta Psychologica, 2023. journalhomepage: www.elsevier.com/locate/actpsy

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,2018. https://arxiv.org/abs/1810.04805,

[5] Hugging Face LoRA Token Classification Guide, https://huggingface.co/docs/peft/task_guides/token-classification-lora#lora-for-token-classification.

[6] Hugging Face Blog on PEFT, https://huggingface.co/blog/peft.

[7] Hugging face fine-tuning guide https://huggingface.co/docs/transformers/training

[8] GPT-3.5 Documentation https://platform.openai.com/docs/models/gpt-3-5

[9] BERT Documentation on Huggingface https://huggingface.co/docs/transformers/model_doc/bert,