**Project 3: Using Pre-Trained Language Models**

**Language as Data at Göttingen University**

Moujan Mirjalili, Farhan Kayhan, Merle (Joris) Hellwig 27.01.2025

**Deliverable 1: Task Analysis for SST-2 (Stanford Sentiment Treebank)**

**Task Overview**

* **Task Name:** SST-2 (Stanford Sentiment Treebank)
* **Dataset Source:** Sentences extracted from movie reviews on Rotten Tomatoes
* **Total Sentences:** 11,855 single sentences
* **Unique Phrases:** 215,154 phrases in the parse trees of 11,855 sentences
* **Annotation:** Each phrase annotated by 3 human judges
* **Task Type:** Binary sentiment classification.
* **Objective:** Predict whether the sentiment of a given sentence is positive or negative.

**Task Setup with Examples**

* **Input**: A single English sentence.
* **Output**: A binary label (0 for negative, 1 for positive).
* **Examples:**

1. **Input**: A fascinating look at a deeply troubled man. → **Output**: Positive (1).
2. **Input**: A dull and lifeless movie. → **Output**: Negative (0).

**Relevant Statistics**

Using publicly available data from the GLUE benchmark for SST-2:

* **Dataset Size:**

1. **Training Set**: 67,349 examples.
2. **Validation Set**: 872 examples.
3. **Test Set**: 1,821 examples (labels withheld for competition submissions).

* **Class Distribution:**

1. **Positive Sentiment**: ~54% of the dataset.
2. **Negative Sentiment**: ~46% of the dataset.

The class distribution is relatively balanced, minimizing bias in evaluation.

* **Sentence Length:**

1. **Average Sentence Length**: ~19 words.
2. **Longest Sentence**: ~52 words.
3. **Shortest Sentence**: 2 words.

* **Preprocessing:**

The sentences are already tokenized and lowercased.

**Manual Instance Analysis**

We reviewed a subset of sentences to classify them into **easy** and **difficult** based on linguistic and contextual complexity.

* **Easy Instances (10 Examples):**

1. **Input**: A thrilling and suspenseful movie! → **Output**: Positive (1).
2. **Input**: Completely boring. → **Output**: Negative (0).
3. **Input**: A must-watch for anyone.**Output**: Positive (1).
4. **Input**: A waste of time and money.**Output**: Negative (0).
5. **Input**: Heartwarming and delightful. → **Output**: Positive (1).
6. **Input**: Avoid this at all costs. → **Output**: Negative (0).
7. **Input**: An amazing performance. → **Output**: Positive (1).
8. **Input**: Terrible in every way. → **Output**: Negative (0).
9. **Input**: Beautifully directed. → **Output**: Positive (1).
10. **Input**: It lacked any sense of fun. → **Output**: Negative (0).

* **Difficult Instances (10 Examples):**

1. **Input**: Not bad at all. → **Output**: Positive (1).

Double-negative can confuse models.

1. **Input**: It could have been better. → **Output**: Negative (0).

Requires understanding implied sentiment.

1. **Input**: An interesting mix of good and bad moments. → **Output**: Neutral/Positive (1).

Mixed sentiment, challenging classification.

1. **Input**: The film tries hard but ultimately fails. → **Output**: Negative (0).

Contrastive sentiment (positive attempt, negative result).

1. **Input**: Leaves the audience wanting more. → **Output**: Positive (1).

Ambiguity around wanting more.

1. **Input**: Some scenes are great, others fall flat. → **Output**: Neutral/Negative (0).

Mixed polarity within a single sentence.

1. **Input**: The acting was fine, but the plot was dull. → **Output**: Negative (0).

Requires prioritization of the dominant sentiment.

1. **Input**: Not a movie I’d recommend. → **Output**: Negative (0).

Implied negativity without explicit words.

1. **Input**: Better than most, but still not great. → **Output**: Neutral/Negative (0).

Complicated sentiment requiring subtle understanding.

1. **Input**: I have mixed feelings about it. → **Output**: Neutral/Negative (0).

Explicit mention of ambiguity.

**Reflection on the Annotation Setup**

* **Annotator Sample:**

The original SST annotations were performed by crowd workers on Amazon Mechanical Turk.

Workers were likely native or fluent English speakers but may have varied in linguistic or cultural expertise.

* **Annotation Guidelines:**

Annotation guidelines provided by the dataset creators included examples of positive, negative, and neutral sentiment.

While detailed, nuances like sarcasm or idiomatic expressions might not have been well-anchored.

* **Inter-Annotator Agreement:**

Agreement metrics are reported in the original dataset documentation.

CoLA-like tasks typically aim for a high Cohen’s Kappa score (≥0.8), but agreement on difficult sentences may be lower.

* **Conflict Resolution:**

Conflicts were resolved using majority voting among annotators.

Instances with strong disagreement may have been excluded or flagged.

* **Dataset Quality:**

Strengths: Balanced dataset, diverse sentence structures, and real-world applicability.

Weaknesses: Some examples are ambiguous, and mixed sentiment instances might lack consistent annotation.

* **Opinion on Quality:**

The quality of the SST-2 dataset is generally high for binary sentiment analysis due to its balanced class distribution, diverse sentence structures, and real-world applicability, which make it a robust resource for training and evaluating models. However, the selection of instances includes some ambiguous examples and sentences with mixed sentiment, which could challenge both annotators and models. The annotations, while effective for most cases, may lack consistency in handling nuanced expressions, such as sarcasm or cultural idioms, and do not always provide clear guidelines for ambiguous or mixed-polarity cases. Enhancements, such as clearer annotation protocols for these edge cases or additional labels for nuanced sentiment, could further improve the dataset’s overall utility and reliability.

* **Explanation of Categorization:**

Easy Instances: Explicit sentiment, simple vocabulary, no ambiguity.

Difficult Instances: Ambiguous sentiment, idiomatic expressions, mixed polarity, or implied meaning.

**Deliverable 2: Finetune**

**Objective**

In this deliverable we fine-tuned GPT-2 on the Stanford Sentiment Treebank (SST-2) dataset from the GLUE benchmark, which involves binary sentiment classification. Additionally, we compared the performance of:

1. A pre-trained GPT-2 model fine-tuned on SST-2. (2 fine-tuning variants)
2. A randomly initialized GPT-2 model fine-tuned on SST-2.

Performance Analysis:

* Comparison of the two models on validation/test metrics.
* Analysis of performance on 20 manually selected instances (10 easy and 10 difficult).
* Reflection on results and their implications.

**Dataset**

* **Dataset**: SST-2 (Stanford Sentiment Treebank, GLUE benchmark task).
* **Task**: Binary classification of sentences into positive or negative sentiment.
* **Dataset Statistics**:
  + Training set: 67,349 samples
  + Validation set: 872 samples
  + Test set: 1,821 samples

**Model Fine-tuning**

1. **Pre-trained GPT-2 Fine-tuning:**

* **Full Fine-Tuning** (Updating all GPT-2 parameters + classifier):

In this approach, we update all trainable parameters of the pre-trained GPT-2 model, including the added classification head. The model contains approximately 124 million trainable parameters, which allows it to leverage the full capacity of GPT-2 for adaptation to the downstream task.

The model starts with an initial train accuracy of 68.75% for the first 10 batches. Over 6 epochs, the training process significantly improves both training and validation accuracy. By the final epoch:

* + - Train Accuracy: 99.34%
    - Validation Accuracy: 94.00%
    - Test Accuracy: 89.79%

The training process takes approximately 2741.62 seconds, reflecting the computational expense of updating the entire model. Loss consistently decreases over epochs, indicating effective convergence. This strategy achieves high performance at the cost of longer training time and computational resources due to the large number of parameters being updated.

* **LoRA** (Introducing Low-Rank Adaptation layers to reduce parameter count):

The LoRA (Low-Rank Adaptation) technique modifies the pre-trained GPT-2 model by introducing lightweight, low-rank layers into specific parts of the model, such as attention or feedforward layers. This drastically reduces the number of trainable parameters to just 1.77 million (about 1.4% of the total parameters because only LoRA layers and classifier parameters are updated, with most GPT-2 layers frozen).

The model starts with moderate training accuracy and validation accuracy in the first epoch:

* Train Accuracy (Epoch 1): 73.84%
* Validation Accuracy (Epoch 1): 81.07%

Over 6 epochs, the performance steadily improves:

* Train Accuracy: 84.83%
* Validation Accuracy: 84.83%
* Test Accuracy: 84.82%

The training process is significantly faster, taking approximately 905.94 seconds just about 33% of the time needed for full fine-tuning. The losses stabilize after several epochs, indicating convergence. While the LoRA approach achieves slightly lower test accuracy compared to full fine-tuning, it provides a computationally efficient alternative with substantially reduced training time and memory requirements, making it suitable for resource-constrained environments.

1. **Randomly Initialized GPT-2 Fine-tuning**

In this approach, the GPT-2 model is not pre-trained and starts with randomly initialized weights. The sequence classification head is added to the model, and all parameters are updated during training. Unlike pre-trained GPT-2, this method requires the model to learn patterns and representations entirely from scratch, relying solely on the downstream task dataset.

The randomly initialized GPT-2 has approximately 124 million trainable parameters, the same as the fully fine-tuned pre-trained model. However, the lack of pre-trained weights means the model’s initial performance is significantly lower. Before any training, its baseline accuracies are:

* Train Accuracy: 44.38%
* Validation Accuracy: 51.88%
* Test Accuracy: 0.00%

This performance is close to random guessing for binary classification tasks, as the model has not yet learned any meaningful patterns. Over the course of training, the model is expected to improve, but the process requires considerably more epochs and computational resources to achieve comparable performance to pre-trained alternatives.

1. **conclusion**

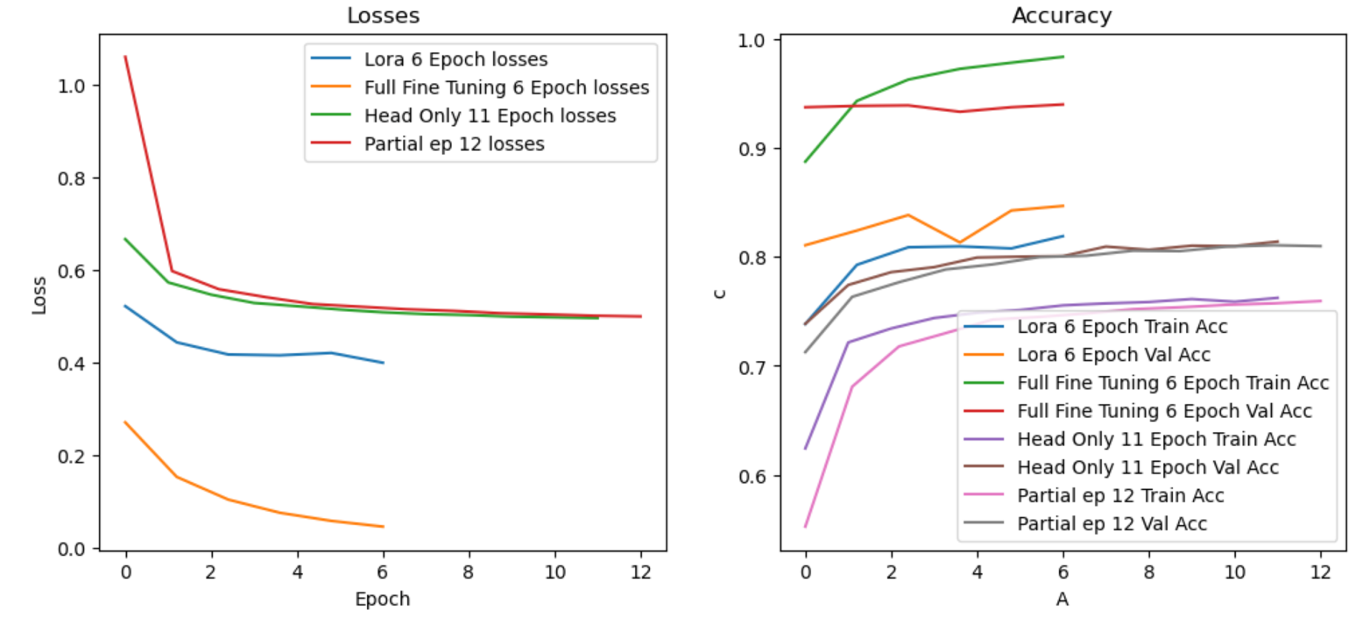
Both Full Fine-Tuning and LoRA Fine-Tuning deliver strong results, but they differ in how much they demand from your hardware and time. Full Fine-Tuning squeezes the most out of GPT-2’s entire parameter set, offering slightly better accuracy, which makes it a great choice if you have plenty of computational resources to spare. On the other hand, LoRA Fine-Tuning is a lot more efficient, giving up a small amount of performance in exchange for drastically lower resource usage and faster training times perfect for situations where hardware is limited or you need quick iterations. Both methods benefit from transfer learning, starting with strong initial performance and converging faster, like LoRA’s impressive 84.83% validation accuracy. Meanwhile, randomly initialized models are at a disadvantage since they start from scratch, need more data, and take much longer to train. Choosing between these approaches really depends on your priorities whether it’s maximizing accuracy or staying efficient with your resources.

**Performance Analysis**

* Comparison of the models based on validation/test metrics. (with plots)

LoRA achieved a balanced performance, with high validation and test accuracies (84.83% and 84.82%) while keeping the training time relatively low compared to full fine-tuning. Full Fine-Tuning reached the highest validation accuracy (94.00%) and test accuracy (89.79%), but at a much higher computational cost (2741.62 seconds). Head-Only Tuning and Partial Fine-Tuning showed comparable results in terms of accuracy, though they were significantly faster to train.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Epochs** | **Train Accuracy (%)** | **Validation Accuracy (%)** | **Test Accuracy (%)** | **Training Time (s)** |
| **LoRA** | 6 | 84.83 | 84.83 | 84.82 | 905.94 |
| **Full Fine-Tuning** | 6 | 99.34 | 94.00 | 89.79 | 2741.62 |
| **Head-Only Tuning** | 11 | 81.07 | 81.41 | 82.34 | 39.54 |
| **Partial Fine-Tuning** | 12 | 81.00 | 80.99 | 81.65 | 31.82 |



* Analysis of performance on 20 manually selected instances (10 easy and 10 difficult).

Full Fine-Tuning consistently performed the best, correctly classifying all easy and difficult instances. LoRA struggled with difficult instances, suggesting that its parameter-efficient nature may limit its ability to capture nuances. Both Head-Only Tuning and Partial Fine-Tuning achieved comparable results, correctly classifying 9/10 difficult instances.

|  |  |  |
| --- | --- | --- |
| **Model** | **Accuracy (Easy)** | **Accuracy (Difficult)** |
| **LoRA** | 90% (9/10) | 70% (7/10) |
| **Full Fine-Tuning** | 100% (10/10) | 100% (10/10) |
| **Head-Only Tuning** | 100% (10/10) | 90% (9/10) |
| **Partial Fine-Tuning** | 100% (10/10) | 90% (9/10) |

* Reflection on results and their implications.
  + Trade-offs Between Efficiency and Accuracy: LoRA is highly efficient and delivers competitive accuracy for most tasks but may struggle with nuanced or ambiguous inputs. This makes it suitable for applications with resource constraints.
  + Full Fine-Tuning: While achieving the best overall performance, its high computational cost makes it less practical for scenarios with limited resources.
  + Head-Only and Partial Fine-Tuning: These methods provide a good balance between efficiency and accuracy. They are particularly useful when computational resources are limited, yet a moderate level of accuracy is required.
  + LoRA could be used for fast and cost-effective deployments with general tasks. Full Fine-Tuning could be used for tasks requiring high precision or nuanced understanding. Opt for Head-Only Tuning or Partial Fine-Tuning could be used for a balance between performance and efficiency.

**challenges**

some explaination

**Deliverable 3: Multilingual Model**

**Option 1: Qualitative Evaluation**

40 prompts in Persian that systematically vary by one of these characteristic (e.g., register, domain, syntactic complexity, tense, and other):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | **Sentence** | **Translation** | **Characteristic** | **Type** |
| 1 | لطفاً پنجره را باز کنید. | Please open the window. | Register | Formal |
| 2 | پنجره رو باز کن. | Open the window. | Register | Informal |
| 3 | آیا می‌توانید به من کمک کنید؟ | Can you help me? | Register | Formal |
| 4 | می‌تونی کمکم کنی؟ | Can you help me? | Register | Informal |
| 5 | لطفاً دفتر خود را به من بدهید. | Please give me your notebook. | Register | Formal |
| 6 | دفتر‌تو بده. | Give me your notebook. | Register | Informal |
| 7 | آیا امکانش هست که چند دقیقه صبر کنید؟ | Can you wait a few minutes? | Register | Formal |
| 8 | میشه یه لحظه صبر کنی؟ | Can you wait a moment? | Register | Informal |
| 9 | امروز هوا آفتابی است. | It's sunny today. | Domain | General |
| 10 | سیستم عامل نیاز به به‌روزرسانی دارد. | The operating system needs to be updated. | Domain | Technical |
| 11 | من عاشق کتاب خواندن هستم. | I love reading books. | Domain | General |
| 12 | این الگوریتم برای کاهش زمان پردازش طراحی شده است. | This algorithm is designed to reduce processing time. | Domain | Technical |
| 13 | یک فنجان قهوه لطفاً. | A cup of coffee, please. | Domain | General |
| 14 | دستگاه شما به وای‌فای متصل نیست. | Your device is not connected to Wi-Fi. | Domain | Technical |
| 15 | من به خرید می‌روم. | I'm going shopping. | Domain | General |
| 16 | پایگاه داده به طور خودکار پشتیبان‌گیری می‌شود. | The database is being automatically backed up. | Domain | Technical |
| 17 | من به خانه رفتم. | I went home. | Syntactic Complexity | Simple |
| 18 | بعد از این که کارم تمام شد، به خانه رفتم. | After I finished work, I went home. | Syntactic Complexity | Complex |
| 19 | او کتاب را خرید. | He bought the book. | Syntactic Complexity | Simple |
| 20 | او کتابی را که دیروز دیدم خرید. | He bought the book I saw yesterday. | Syntactic Complexity | Complex |
| 21 | ما فیلم دیدیم. | We watched the movie. | Syntactic Complexity | Simple |
| 22 | وقتی باران تمام شد، ما به تماشای فیلم رفتیم. | When the rain stopped, we went to watch the movie. | Syntactic Complexity | Complex |
| 23 | او دوید. | He ran. | Syntactic Complexity | Simple |
| 24 | او به سمت درختی که در دوردست بود دوید. | He ran to a tree in the distance. | Syntactic Complexity | Complex |
| 25 | من امروز صبح ورزش کردم. | I exercised this morning. | Tense | Past |
| 26 | من هر روز ورزش می‌کنم. | I exercise every day. | Tense | Present |
| 27 | فردا صبح ورزش خواهم کرد. | I will exercise tomorrow morning. | Tense | Future |
| 28 | او کتابی خواند. | He read a book. | Tense | Past |
| 29 | او کتابی می‌خواند. | He is reading a book. | Tense | Present |
| 30 | او کتابی خواهد خواند. | He will read a book. | Tense | Future |
| 31 | ما به مسافرت رفتیم. | We went on a trip. | Tense | Past |
| 32 | ما به مسافرت می‌رویم. | We are going on a trip. | Tense | Present |
| 33 | ما به مسافرت خواهیم رفت. | We will go on a trip. | Tense | Future |
| 34 | این جمله طعنه‌آمیز است. | This sentence is sarcastic. | Miscellaneous | Sarcasm |
| 35 | شما همیشه دیر می‌رسید. | You are always late. | Miscellaneous | Ambiguity |
| 36 | آیا می‌توانید دلیل این موضوع را توضیح دهید؟ | Can you explain why? | Miscellaneous | Reasoning |
| 37 | هرچند خسته بودم، تا آخر شب بیدار ماندم. | Although I was tired, I stayed up late. | Miscellaneous | Concessive |
| 38 | اگر باران نبارد، به پارک می‌رویم. | If it doesn't rain, we will go to the park. | Miscellaneous | Conditional |
| 39 | کتابی که دیروز خریدی، کجاست؟ | Where is the book you bought yesterday? | Miscellaneous | Embedded clause |
| 40 | آن‌ها تصمیم گرفتند سفر را به تعویق بیاندازند. | They decided to postpone the trip. | Miscellaneous | Complex verb |

**Multilingual Analysis:**

Some explaination

**Deliverable 4: Project Summary**

**Key Findings:**

Some explaination

**Challenges:**

Some explaination

**Work Process:**

Some explaination