Indian Institute of Technology Gandhinagar

Assignment 3:

Pretraining and fine-tuning an LLM

CS 613: NLP

Group T10: Exquisiters

Aditi Dey (20110007)

Anavart Pandya (20110016)

Ayush Gupta (20110031)

Mann Kumar Jain (20110108)

Pinki Kumari(22210028)

Ronak Kalra (20110171)

Sakshi Jain (20110181)

Vedang Chavan (20110222)

Task 1: Bert-base-uncased model.

Task 2: Parameters of the selected model from the code

```
+---------------------------------------------------+------------+
|                      Modules                      | Parameters |
+---------------------------------------------------+------------+
|          embeddings.word_embeddings.weight        |  23440896  |
|        embeddings.position_embeddings.weight      |   393216   |
|       embeddings.token_type_embeddings.weight     |    1536    |
|             embeddings.LayerNorm.weight           |    768     |
|              embeddings.LayerNorm.bias            |    768     |
|    encoder.layer.0.attention.self.query.weight    |   589824   |
|     encoder.layer.0.attention.self.query.bias     |    768     |
|     encoder.layer.0.attention.self.key.weight     |   589824   |
|      encoder.layer.0.attention.self.key.bias      |    768     |
|    encoder.layer.0.attention.self.value.weight    |   589824   |
|     encoder.layer.0.attention.self.value.bias     |    768     |
|   encoder.layer.0.attention.output.dense.weight   |   589824   |
|    encoder.layer.0.attention.output.dense.bias    |    768     |
| encoder.layer.0.attention.output.LayerNorm.weight |    768     |
|  encoder.layer.0.attention.output.LayerNorm.bias  |    768     |
|     encoder.layer.0.intermediate.dense.weight     |  2359296   |
|      encoder.layer.0.intermediate.dense.bias      |    3072    |
|        encoder.layer.0.output.dense.weight        |  2359296   |
|         encoder.layer.0.output.dense.bias         |    768     |
|      encoder.layer.0.output.LayerNorm.weight      |    768     |
|       encoder.layer.0.output.LayerNorm.bias       |    768     |
|                pooler.dense.weight                |   589824   |
|                 pooler.dense.bias                 |    768     |
+---------------------------------------------------+------------+
There are 12 such encoder layers.
So Total Trainable Params: 109482240
109482240
```

The model's parameter count was computed to be **109,482,240** through the code. This count closely aligns with the parameter figure of **108,432,384** as reported in the corresponding paper. The slight variance between the calculated and reported counts can potentially stem from differences in implementation details, such as parameter initialization methods and biases. The exact number of parameters from the different layers were also calculated, and have been attached in the end.

Task 3: Pretrain the selected model on the train split of ‘wikitext-2-raw-v1’

The Models have been pushed to 🤗. We used the following hyperparameters:
Learning Rate = 5e-5

Epochs = 5
Batch Size: 36718
Sequence Length: 512

Models were trained using the specified hyperparameters for 5 epochs, each epoch involving batches of 36718 samples, with sequences padded or truncated to a maximum length of 512 tokens. The above hyperparameters are also inspired by the research paper.

Task 4: The Perplexity scores

EPOCH	**Mean Perplexity**
1	3009244.135
2	2351987.914
3	2162127.378
4	1900714.996
5	1743680.844

The perplexity score decreases after each epoch during training.. There are multiple reasons for this:

- Initially, as the model learns more from the training data, its ability to predict sequences improves, resulting in decreased perplexity on the test data.
- The choice of hyperparameters and the model architecture influence the convergence pattern.
- If we train on more epochs, the model starts memorizing the training data, causing overfitting. This might lead to an increase in perplexity on the test data.

Task 5: The pre-trained model has been pushed to 🤗.

Here is the link to the pushed model: *link*

Task 6 : The final pre-trained model has been fine-tuned on the train split of the dataset for the below tasks:

a. Classification: SST-2
b. Question-Answering: SQuAD

Task 7: For each of the following metrics corresponding to their particular task, the scores have been calculated on the test splits of the dataset:

a. Classification: Accuracy, Precision, Recall, F1
Scores:

METRIC	SCORE
Recall	1.0
Precision	0.551
Accuracy	0.551
F1	0.7105

- A recall of 1.0 means that the model is capturing all the positive instances in the dataset.
- A precision of 0.551 indicates that about 55.1% of the predicted positive instances are true positives or positive sentiment.
- The accuracy of 0.551 means that the model correctly predicts the class of the instances about 55.1% of the time. In this case, it seems that the model is not performing much better than random guessing.
- An F1 score of 0.7105 is relatively good, suggesting a reasonable balance between precision and recall. However, the low precision indicates that there are still false positives that need to be addressed.

b. Question-Answering: squad_v2, F1, METEOR, BLEU, ROUGE, exact-match
Scores:

METRIC	SCORE
exact	7.479
f1	12.104
total	11873
HasAns_exact	14.98
HasAns_f1	24.245
HasAns_total	5928
NoAns_exact	0
NoAns_f1	0
NoAns_total	5945
best_exact	50.088
best_exact_thresh	0
best_f1	50.092
best_f1_thresh	0

meteor	0.00098
bleu	0
precisions	[1.5e-05, 0.0, 0.0, 0.0]
brevity_penalty	1
length_ratio	5.60
translation_length	66491
reference_length	11873
rouge1	2.11E-05
rouge2	0
rougeL	2.11E-05
rougeLsum	2.11E-05

In assessing the ROPES (Reading for Open-Ended Problem Solving) dataset, F1 score is the preferred metric, measuring a system's adeptness at applying passage knowledge to novel situations. For the evaluation of generative Question-Answering (QA) models, METEOR stands out as the optimal metric, providing a thorough evaluation of the model's performance, especially in generating responses that emulate human-like language and expression.

Task 8: Number of parameters in the model after fine-tuning are

After Fine-Tuning For Classification:

```
+--------------------------------------------------------+------------+
|                        Modules                         | Parameters |
+--------------------------------------------------------+------------+
|         bert.embeddings.word_embeddings.weight         |  23440896  |
|       bert.embeddings.position_embeddings.weight       |   393216   |
|      bert.embeddings.token_type_embeddings.weight      |    1536    |
|            bert.embeddings.LayerNorm.weight            |    768     |
|             bert.embeddings.LayerNorm.bias             |    768     |
|    bert.encoder.layer.0.attention.self.query.weight    |   589824   |
|     bert.encoder.layer.0.attention.self.query.bias     |    768     |
|     bert.encoder.layer.0.attention.self.key.weight     |   589824   |
|      bert.encoder.layer.0.attention.self.key.bias      |    768     |
|    bert.encoder.layer.0.attention.self.value.weight    |   589824   |
|     bert.encoder.layer.0.attention.self.value.bias     |    768     |
|   bert.encoder.layer.0.attention.output.dense.weight   |   589824   |
|    bert.encoder.layer.0.attention.output.dense.bias    |    768     |
| bert.encoder.layer.0.attention.output.LayerNorm.weight |    768     |
|  bert.encoder.layer.0.attention.output.LayerNorm.bias  |    768     |
|     bert.encoder.layer.0.intermediate.dense.weight     |  2359296   |
|      bert.encoder.layer.0.intermediate.dense.bias      |    3072    |
|        bert.encoder.layer.0.output.dense.weight        |  2359296   |
|         bert.encoder.layer.0.output.dense.bias         |    768     |
|      bert.encoder.layer.0.output.LayerNorm.weight      |    768     |
|       bert.encoder.layer.0.output.LayerNorm.bias       |    768     |
|                bert.pooler.dense.weight                |   589824   |
|                 bert.pooler.dense.bias                 |    768     |
|                   classifier.weight                    |    1536    |
|                    classifier.bias                     |     2      |
+--------------------------------------------------------+------------+
There were 12 such encoder layers.
Total Trainable Params: 109483778
109483778
Num of Parameters Increased by 1538 after fine-tuning for classification.
```

After Fine-Tuning For Question Answering:

```
+--------------------------------------------------------+------------+
|                        Modules                         | Parameters |
+--------------------------------------------------------+------------+
|         bert.embeddings.word_embeddings.weight         |  23440896  |
|       bert.embeddings.position_embeddings.weight       |   393216   |
|      bert.embeddings.token_type_embeddings.weight      |    1536    |
|            bert.embeddings.LayerNorm.weight            |    768     |
|             bert.embeddings.LayerNorm.bias             |    768     |
|    bert.encoder.layer.0.attention.self.query.weight    |   589824   |
|     bert.encoder.layer.0.attention.self.query.bias     |    768     |
|     bert.encoder.layer.0.attention.self.key.weight     |   589824   |
|      bert.encoder.layer.0.attention.self.key.bias      |    768     |
|    bert.encoder.layer.0.attention.self.value.weight    |   589824   |
|     bert.encoder.layer.0.attention.self.value.bias     |    768     |
|   bert.encoder.layer.0.attention.output.dense.weight   |   589824   |
|    bert.encoder.layer.0.attention.output.dense.bias    |    768     |
| bert.encoder.layer.0.attention.output.LayerNorm.weight |    768     |
|  bert.encoder.layer.0.attention.output.LayerNorm.bias  |    768     |
|     bert.encoder.layer.0.intermediate.dense.weight     |  2359296   |
|      bert.encoder.layer.0.intermediate.dense.bias      |    3072    |
|        bert.encoder.layer.0.output.dense.weight        |  2359296   |
|         bert.encoder.layer.0.output.dense.bias         |    768     |
|      bert.encoder.layer.0.output.LayerNorm.weight      |    768     |
|       bert.encoder.layer.0.output.LayerNorm.bias       |    768     |
|                   qa_outputs.weight                    |    1536    |
|                    qa_outputs.bias                     |     2      |
+--------------------------------------------------------+------------+
There were 12 such encoder layers.
Total Trainable Params: 108893186
108893186
```

Task 9: The fine-tuned model has been pushed to 🤗 : Fine-Tuned Classification, Fined-Tune-QA

Comments and Discussion:

Performance

The model’s performance is poor and the scores from the performance metrics is not up to the mark as:

- The model's Exact Match (EM) score of 7.48% indicates a relatively low accuracy in providing exact answers to questions in the dataset.
- The F1 score of 12.10% suggests a modest balance between precision and recall, but there is room for improvement.
- The model performs better on questions with answers (HasAns) with an EM score of 14.98% and an F1 score of 24.24%. This indicates some ability to answer questions when information is available.
- The model's performance on questions with no answers (NoAns) is notably lower, with an EM and F1 score of 0.00%. This suggests challenges in identifying unanswerable questions.
- The best exact match and F1 scores are relatively higher at 50.09%, indicating that the model can achieve better performance under certain conditions or thresholds.
- The METEOR score of 0.00098 is quite low, suggesting that the model may struggle with generating responses that closely align with human-like language and expression.
- The BLEU score of 0.00 indicates minimal overlap between the model's generated text and reference text.
- The ROUGE scores are also very low, suggesting limited similarity between the model's outputs and reference text.

Work Distribution

Name	Contribution
Aditi	Calculated number of parameters of selected model, Helped in Pre-training
Anavart	Fine-Tuned for Classification
Ayush	Fine-Tuned for Question & Answering, Calculated Metric Scores for Question-Answering, Performance Inference, Documentation
Mann	Pre-Training the BERT Model using MLM and NSP, Pushed model to HuggingFace, Calculated perplexity, Documentation
Pinki	Pre-Training the Bert Model using MLM and NSP

Ronak	Fine-Tuned for Question & Answering, Calculated metrics for the Classification task
Sakshi	Fine-Tuned for Question & Answering, Documentation
Vedang	Pre Training the BERT Model, Fine-tuning for classification, Error analysis and debugging, Calculated perplexity