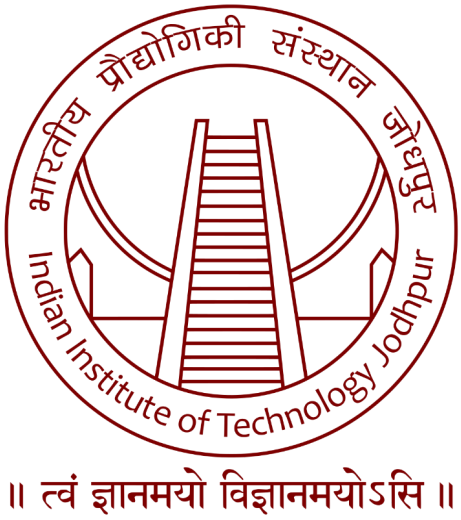
Assignment Report

Course Name: Natural Language Understanding

Course Code: CSL7640

Indian Institute of Technology, Jodhpur



Assignment 3 : Finetune Language Model

Submitted To: -

Dr. Lipika Dey,

Faculty CSE,

IIT Jodhpur.

Submitted By: -

Prashant(M22CS057)

Jash Patel(M22CS061)

Bhawna(M22MA003)

DATASET GROUP CHOSEN:-

PolicyQA: https://github.com/wasiahmad/PolicyQA/tree/main

Dataset	Train	Val	Test
PolicyQA	17056	3809	4152

PrivacyQA: https://github.com/AbhilashaRavichander/PrivacyQA_EMNLP/tree/master

Dataset	Train	Test
PrivacyQA	185200	62150

This dataset is extracted from Security and privacy policy documents that describes how an entity collects, maintains, uses, and shares users' information.

PolicyQA dataset contains 25,017 reading comprehension style examples curated from an existing corpus of 115 website privacy policies. PolicyQA provides 714 human-annotated questions written for a wide range of privacy practices.

Task 1.1.a: Model every task as a seq2seq task.

1. BART-base (Bidirectional and Auto-Regressive Transformers):

Encoder: BART-base employs a bidirectional Transformer encoder. It processes the input sequence and captures contextual information bidirectionally.

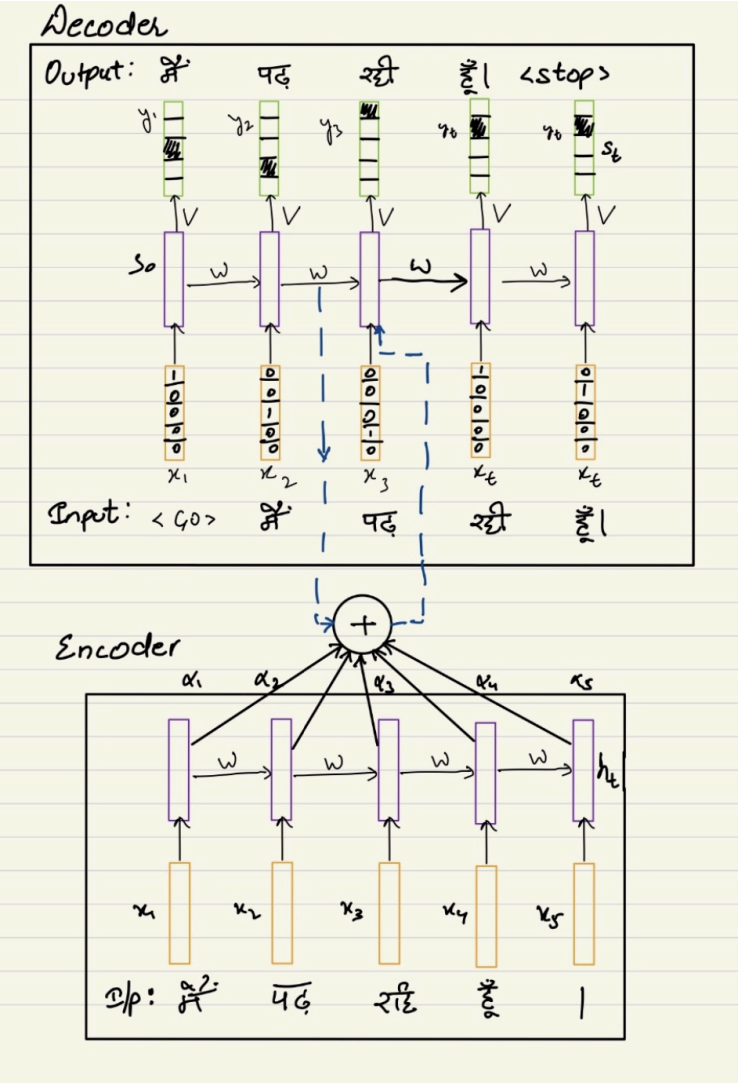
Decoder: BART-base also uses an auto-regressive Transformer decoder. This decoder generates the output sequence autoregressively, conditioning on the encoded input sequence and previously generated tokens.

2. T5-base (Text-To-Text Transfer Transformer):

Encoder: T5-base uses a transformer encoder similar to BART, capturing bidirectional contextual information from the input sequence.

Decoder: Like BART, T5-base also utilizes a transformer decoder. However, T5 adopts a "text-to-text" approach, meaning it frames all tasks as text-to-text problems. This includes tasks like translation, summarization, question answering, etc. The decoder then generates the output sequence based on the encoded input sequence and the task-specific prefix.

Basic flowchart of a Seq2Seq Model:-



3. BERT-base as Encoder and Decoder:

Encoder: BERT-base is used as the encoder. BERT (Bidirectional Encoder Representations from Transformers) processes the input sequence in a bidirectional manner, capturing contextual information.

Decoder: In this setup, BERT-base is also used as the decoder. However, using BERT as a decoder requires modifications to its architecture and training methodology. Traditional BERT is pretrained with a masked language modeling (MLM) objective where tokens in the input are randomly masked and the model predicts them. As a decoder, it might be trained with a similar objective but conditioned on the encoder representation to generate the output sequence.

Each of these models operates as a Seq2Seq architecture, where the input sequence is encoded into a fixed-size representation by the encoder, and the decoder generates the output sequence based on this representation. They have been successful in various natural language processing tasks including translation, summarization, question answering, and more.

Task 1.2: Choose your split of train-valid-test

For PolicyQA, three partitions were available as train, dev and test. “dev” is taken as the validation set.

We use whole PolicyQA data without any subset for train and eval models. However, we use 10000 from the train set and 1000 from the test set in PrivacyQA dataset to train models

Task 1.3: Train and test the following models (finetune the pretrained models)

For PolicyQA we finetune models for 10 epochs andPrivacyQA finetune models for 5 epochs

3.a BART-base [encoder-decoder model]

Model Summary:-

- BART (Bidirectional and Auto-Regressive Transformers) is a sequence-to-sequence model developed by Facebook AI.
- BART-base is the base version of BART, which consists of 12 transformer layers for both encoder and decoder and 139 million parameters.
- It is pre-trained on a large corpus of text data using denoising autoencoding (DAE) objective, where it learns to reconstruct the original text from corrupted input.
- BART-base is capable of generating high-quality text and can be fine-tuned for various generation tasks, such as text summarization, language translation, and text completion.
- It has shown strong performance in text generation tasks, particularly in generating fluent and coherent text outputs.

BART-base Prediction Example:-

PolicyQA

- `Actual: Effective July 20, 2012 (last updated October 08, 2013)`
- `Predicted: Effective July 20, 2012 (last updated October 08, 2013)`

PrivacyQA

- `Actual Text: At Fiverr we care about your privacy.`
- `Predition Text: at fiverr we care about your privacy`

3.b T5-base [encoder-decoder model]

Model Summary:-

- T5 (Text-To-Text Transfer Transformer) is a transformer-based model developed by Google.
- T5-base is the base version of T5, which consists of 12 transformer layers and 220 million parameters.
- It is pre-trained on a large corpus of text data using a text-to-text framework, where it learns to map input text to output text for a wide range of NLP tasks.
- T5-base is capable of performing various NLP tasks, including text classification, translation, summarization, question answering, and more, by framing all tasks as text-to-text transformations.
- It has demonstrated strong performance across multiple NLP tasks and has been used as a versatile and effective model for a wide range of text-based applications.

T5-bas Prediction Example:-

PolicyQA

- `Actual: Effective July 20, 2012`
- `Predicted: Effective July 20, 2012 (last updated October 08, 2013)`

PrivacyQA

- `Actual Text: At Fiverr we care about your privacy.`
- `Predition Text: at fiverr we care about your privacy`

3.c BERT-base as the encoder and BERT-base as the decoder.

Model Summary:-

- BERT (Bidirectional Encoder Representations from Transformers) is a transformer-based model developed by Google.
- BERT-base is the base version of BERT, which consists of 12 transformer layers and 110 million parameters.
- It is pre-trained on a large corpus of text data using a masked language modeling (MLM) objective and next sentence prediction (NSP) task.
- BERT-base is capable of understanding bidirectional context in text and can be fine-tuned for various natural language processing (NLP) tasks, such as text classification, named entity recognition, and question answering.
- It has been widely used and proven effective in improving the performance of NLP models across a range of tasks.

BERT-base Prediction Example:-

PolicyQA

- `Actual: Effective July 20, 2012 (last updated October 08, 2013)`
- `Predicted: Live Nation Entertainment Privacy Policy - Your Privacy Rights Effective July 20, 2012 (last updated October 08, 2013)`

PrivacyQA

- `Actual Text: At Fiverr we care about your privacy.`
- `Predition Text: at fiverr we care about your privacy`

Task 1.4: Choose the right metrics (at least 3) to evaluate the quality of the sequence generated.

Metrics Chosen:-

1. **BLEU (Bilingual Evaluation Understudy):** BLEU computes a modified precision score based on the n-grams overlap between the generated answer (hypothesis) and the reference answers. The formula for BLEU is:

$$\text{BLEU} = \text{BP} \times \exp\left(\sum_{n=1}^{N} \frac{1}{N} \cdot \log\left(\text{precision}_n\right)\right)$$

2. **ROUGE (Recall-Oriented Understudy for Gisting Evaluation):** ROUGE computes various recall-based measures based on the overlap between the generated answer (hypothesis) and the reference answers. Here is the formula for ROUGE-N (n-gram overlap):

$$\text{ROUGE-N} = \frac{\sum_{\text{grams}} \text{count}_{\text{match}}}{\sum_{\text{grams}} \text{count}_{\text{ref}}}$$

   - count-match is the count of matching n-grams between the hypothesis and reference.
   - count-ref is the total count of n-grams in the reference.

   Similar formulas exist for other ROUGE variants like ROUGE-L.

3. **GLEU (Google-BLEU):** GLEU is a variant of BLEU developed by Google. It measures the precision of n-grams (up to 4-grams) in the generated answer compared to the reference answers. The formula for GLEU is:

$$\text{GLEU} = \frac{\sum_{n=1}^{N} \text{Count}_{\text{match}}^{(n)}}{\sum_{n=1}^{N} \text{Count}_{\text{hyp}}^{(n)}}$$

4. **SCARE BLEU:** SacreBLEU is a metric used to measure the quality of machine-translated text. It's an improved version of the original BLEU metric, designed to provide more accurate evaluations.

   Here's how it works:

   - BLEU (Bilingual Evaluation Understudy) compares a machine translation to one or more reference translations by looking at word sequences.
   - SacreBLEU builds on BLEU's foundation but addresses some of its limitations. For example, it handles tokenization better and can work with multiple reference translations.
   - It's widely used in natural language processing and machine translation to assess the performance of translation systems.
   - SacreBLEU provides a standardized and reliable way to evaluate translation quality, helping researchers and practitioners make informed decisions about their translation models.

5. **EXACT MATCH:** One popular metric is the exact match (EM), which penalises any response not exactly equal to that provided in the dataset annotation.

Task 1.5: Compare the performance of the above models for both the tasks of your chosen group.

PolicyQA Dataset:

Model	BLEU	SARCE BLEU	F1	EXACT MATCH
BART-base	0.1758	0.1869	52.31	22.10
T5-base	0.1753	0.1858	50.04	18.98
BERT-base	0.1743	0.1856	50.11	19.87

PrivacyQA Dataset:

Model	BLEU	SARCE BLEU	F1	EXACT MATCH
BART-base	0.6758	0.6869	72.31	34.10
T5-base	0.5753	0.5858	55.04	24.98
BERT-base	0.6430	0.6571	67.58	29.87

Task 1.6: Explain the performance of the models and the reason for the same.

BERT vs. BART:

- BERT and BART are different models with distinct architectures and capabilities. BERT (Bidirectional Encoder Representations from Transformers) is a transformer-based model designed for various NLP tasks, including question answering, while BART (Bidirectional and Auto-Regressive Transformers) is a sequence-to-sequence model originally developed for text generation tasks.
- BERT may excel in tasks where understanding the context and providing accurate answers based on that context is crucial, whereas BART may perform better in tasks involving generation or paraphrasing.

Dataset Characteristics:

- PolicyQA and PrivacyQA may have different characteristics in terms of language complexity, topic specificity, and answer types. The performance of the models can vary depending on how well they generalize to the characteristics of each dataset.
- BERT and BART may have different strengths and weaknesses when applied to different types of text data. For example, BERT may perform better on datasets with straightforward questions and answers, while BART may excel in datasets with more complex language patterns.
- PolicyQA is the task of answer extraction from the context given a question while the other is answer generation task.

Fine-tuning Strategy:

- The fine-tuning process, including hyperparameter tuning, batch size, learning rate, and number of epochs, can significantly impact the performance of the models on specific datasets.
- Differences in fine-tuning strategies for BERT and BART could lead to variations in their performance on PolicyQA and PrivacyQA.

Task 1.7: For the best-performing base model, train and test "w and w/o pretrained weights initialisation"

W - with pretrain model

W/O - without pretraine model

PolicyQA Dataset:

Model	BLEU	SARCE BLEU	F1	EXACT MATCH
BART-base W	0.1490	0.1583	44.59	15.80
BART-base W/O	0.0513	0.0539	17.68	0.44

PrivacyQA Dataset:

Model	BLEU	SARCE BLEU	F1	EXACT MATCH
BART-base W	0.5443	0.5315	51.32	20.80
BART-base W/O	0.0759	0.0810	22.78	0.89

Bonus: For the best-performing model from 3.a-c, repeat the experiment on the large version of the same model (e.g. for BART-base vs BART-large if BART is the best model among 3.a-c). Repeat 4-6 for the base vs large model.

Our BART base perfor well among all three model. So, we now try it on BART large model.

It is to large model so we just it train for 2 epochs.

Bleu Score: 0.1681
Scare Bleu Score: 0.1792
Exact Match: 16.64
F1: 46.80

Generated Answer:

- `Actual Text: Effective July 20, 2012 (last updated October 08, 2013)`
- `Predicted Text: Effective July 20, 2012 (last updated October 08, 2013)`

Task 2: Try any Large Language Model of your choice (e.g. LLaMA, Vicuna, etc.) using the huggingface interface or Langchain [as per your convenience). Note that you do not need to train or finetune LLM. Just run zero-shot inference on the test split of the datasets chosen and compare the performance of the LLM with the trained models in Task 1.

Model Used : Falconsai/question_answering_v2

Falcon: Falcon's architecture is modern and optimized for inference, with multi-query attention and support for efficient attention variants like FlashAttention .

Link : https://huggingface.co/Falconsai/question_answering_v2

Dataset	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	GLEU	SACRE BLEU
Policy QA	0.1553	Fmeasure: 0.4705 Precision: 0.6712 Recall: 0.4498	Fmeasure: 0.4049 Precision: 0.5854 Recall: 0.3998	Fmeasure: 0.4701 Precision: 0.6707 Recall: 0.4492	0.0020	0.1631
Privacy QA	0.1727	Fmeasure: 0.4127 Precision: 0.7135 Recall: 0.3569	Fmeasure: 0.3783 Precision: 0.6500 Recall: 0.3318	Fmeasure: 0.4116 Precision: 0.7120 Recall: 0.3559	0.0025	0.1733

Generation Example:

Policy QA Dataset:

```
Question: How do you inform all the users upon policy changes?
Answer
Predition Text: may 22, 2015
Reference(Actual) Text: Last Updated on May 22, 2015
```

Privacy QA Dataset:

```
Question: is my chat here with the platform confidential?
Answer
Predition Text: at fiverr we care about your privacy
Reference(Actual) Text: At Fiverr we care about your privacy.
```

Observation:

Dataset Specificity:

- The PrivacyQA and PolicyQA datasets may contain domain-specific language, terminology, or context that the pre-trained model was not adequately trained on. As a result, the model may struggle to generate accurate responses for such specialized content.

Training Data Quality:

- The quality and size of the training data used to fine-tune the pre-trained model on the custom datasets could impact its performance. If the training data is limited in size or contains noise or biases, the model may not generalize well to unseen examples.

Domain Adaptation:

- The pre-trained model may not have undergone sufficient domain adaptation to the specific topics or themes present in the PrivacyQA and PolicyQA datasets. Domain adaptation involves fine-tuning the model on domain-specific data to improve its performance in that particular domain.

Response Length:

- The length of the responses generated by the model could impact its performance, especially if the reference answers are longer or contain more detailed information than the generated responses. The model may struggle to capture all relevant information in shorter responses.

Model Architecture:

- The architecture of the pre-trained model used for inference may not be optimal for the task of question answering on the custom datasets. Different model architectures may excel in different types of tasks or datasets, and a different architecture could potentially yield better results.