**Question Answering System using Transformer Based Models**

Pramit Gopal Yeole Vignan Vennampally   
[yeole.pr@northeastern.edu](mailto:yeole.pr@northeastern.edu) [vennampally.v@northeastern.edu](mailto:vennampally.v@northeastern.edu)

Vidyun Akila Sundhara Ramaan  
[akilasundhararaama.v@northeastern.edu](mailto:akilasundhararaama.v@northeastern.edu)

1. **Abstract**

The goal of this project is to model a Question Answering (QA) System. The dataset used to train the model is the SQuAD 2.0 dataset. We implement transformer-based models ALBERT, RoBERTa, and DistilBERT for this purpose. For each of the models, we employ the transfer learning approach where the pre-trained model is imported and trained on the pre-processed SQuAD dataset for the QA downstream task. Following the training, the models were evaluated on the test set where the RoBERTa model achieved an EM and F1 score of 0.65 and 0.53 which was around 25% higher than performance of ALBERT and around 35% higher than the performance achieved using DistilBERT.

1. **Introduction**

The amount of data being generated in today’s world has risen tenfold in the past couple of years. This ever-growing data contains everything from transactional data, research data, social media posts, and open-source information. Much of this data is unstructured and majorly text. With such a vast resource, searching for specific information on the web has also become a cumbersome task. A solution to this problem is a Question Answering System which has applications in customer support, Legal QA, conversational chatbots, and also getting instant answers to questions over search.

Question Answering (QA) is the task of finding an answer to a posed question from a given context if it’s provided. Question Answering systems allow users to pose a question and get an answer from the respective document or article. QA systems are mainly of 2 types namely, Extractive Question Answering where answers are extracted from the given context and Generative Question Answering where a newly generated text serves as the answer for the posed. A typical approach to develop a QA system has been using rule-based systems, Information retrieval systems, Machine Learning based approaches, and neural network-based approaches such as RNNs, and CNNs. But a major downside of these methods was their inability to handle long range text sequences and contextual understanding. With the development of Transformer Models based on self-attention mechanism that enables models to capture log-term dependencies and contextual relationships in unstructured text.

For this project, we implement Transformer based models namely, ALBERT, RoBERTa, and DistilBERT to develop a Question Answering System. The models are trained and evaluated on the training and evaluation set of the [SQuAD 2.0 dataset](https://rajpurkar.github.io/SQuAD-explorer/dataset/train-v2.0.json). The mentioned models were chosen for their low computational requirements and high performance relative to state-of-art Large Language models like BERT, GPT, LLaMA, etc. These models have initially been trained on large amounts of text data using unsupervised pre-training mainly on 2 tasks, Masked Language Modelling and Next Sentence Prediction. These pre-trained models enable transfer learning as these pre-trained models can then be fine-tuned on specific downstream NLP tasks such as Question answering. To enable model training, new features such as answer start and answer end were created. Following this, tokenization and encoding using the AutoTokenizer function using the base versions of the pre-trained models was carried out. The models were then trained utilizing the AdamW optimizer with weight decay and a lookahead wrapper optimizer and then evaluated on the valid set.

1. **Background**

The SQuAD dataset has been a well-known benchmark for evaluating various question-answering models and has been the major contributor in advancing the research initiatives across the world. The SQuAD leader board is another place where models can be seen to have achieved performance better than human performance (EM: 86.83%, F1: 89.45%) on the SQuAD 2.0 dataset. The IE-Net (Ensemble) transformer model has seen the best performance till date with a F1 Score of 93.214% and EM 90.939%. Many Large Language Models (LLMs) have been developed in the context of advanced downstream NLP tasks. Prior work on the SQuAD dataset has been done using models ranging from Logistic Regression, Neural networks, BiDAF, to LLMS like BERT, GPT, etc.

Even though the SQuAD dataset has been a leader in the evaluation benchmark of the state-of-art Language models in Question answering and other machine comprehension tasks, there are a few limitations that can be attributed to the limited coverage of topics in the corpus. Another limitation is that the topics covered in the corpus may not be representative of the complete natural language and some bias may exist in terms of type of questions and the context sourced from the web.

Majority of the latest work on question answering systems has been implemented using state-of-art transformer models like BERT by Google which achieved a 93.2% accuracy on the SQuAD dataset which 3% over human evaluation. Recent developments in the Question Answering Systems range from inclusion of generative question answering combined with the power of extraction from which the respective models were trained on. One such example is ChatGPT modelled using the latest GPT 4.0 LLM which incorporates every aspect of question answering and other downstream NLP tasks.

1. **Approach**

We followed a simple approach to model the QA system. The training corpus contains data fields ‘id’, ‘title’, ‘context’, ‘question’, ‘text’, ‘answer\_start’, and ‘is\_impossible’. Through a preliminary analysis, the field ‘Id’ was dropped and new fields of ‘answer\_start’ and ‘answer\_end’ were created to represent the accurate answers to the questions posed from the given context. If the question is answerable, the answer field indicated the start and end positions of the answer in the given context else it tokenizes the answer to the max length of the tokenizer. In order to train the model, a lookahead wrapper optimizer and a base AdamW optimizer were utilised. The lookahead optimizer averages the parameters with an estimate of the future parameters at each time step. This improves the convergence condition of the base optimizer, and it is done till a fixed number of steps. In our model, the fixed no. of steps which is essentially a hyper parameter, is taken to be 5. The base optimizer AdaW was used to minimize overfitting and is used to improve generalization of the deep learning models.

The main hyper-parameters taken into consideration for our model are:

* Learning rate: 1e-4
* Validation Batch size = 16
* Training Batch size = 16
* Epochs = 5

For the Project 3 models were taken, DistilBert, Albert, Roberta

**4.1 DistilBERT**

DistilBERT is a smaller version of BERT developed by Google. Main difference between BERT and DistilBERT is the Size and Computation Resources they use. DistilBERT is nearly half the size of the BERT and require few computation resources to train and run.

**BERT:**

BERT stands for Bidirectional Encoder Representation from Transformers is a Pre trained NLP model developed by Google in 2018. BERT is based on Transformer architecture which is an attention mechanism that learns contextual relationships between words in a text. BERT is Pretrained on large Unsupervised Data using two tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). In MLM tasks BERT learns to predict masked word in a sentence using surrounding words, in NSP task the model learns to predict if two sentences are consecutive or not. This allows BERT to learn complex patterns which can be fine-tuned for specific NLP tasks like Question Answering System.

BERT uses two special types of token inputs [CLS] and [SEP]. [CLS] is there to represent sentence level classification and is used when classifying. [SEP] is used to separate two pieces of text. BERT also uses Segment Embeddings to differentiate a question from text and Position Embeddings that helps in positioning the words in sequence. These three are the inputs to the first layer.

The Output of these Transformer Layers is the word embeddings that are multiplied by weight vectors and passed on to SoftMax Function to get the probability distribution of all the words. The word with the highest probability is assigned as the Start word. Similar process is followed to generate probability distribution of End word.

**4.2 ALBERT (A Lite BERT)**

ALBERT is the variant of BERT developed to reduce limitations of BERT like the computation and memory requirements. ALBERT Specifically implement two parameter reduction techniques to reduce computation and increase performance.

**Factorized Embedding Parameterization:** ALBERT divides the embedding matrix into two pieces to ensure the size of hidden layers and embedding dimensions are different. This increases the size of hidden layer without modifying the size of embedding dimension. Alberta adds a linear layer to the embedding matrix after the embedding phase is done.

**Cross Layer Parameter Sharing:** Alberta has 12 encoder blocks each that shares all the parameters. This process reduces the parameter size and also adds regularization to the model.

In Alberta masked inputs are generated using n-gram masking with maximum n-gram span of 3. Alberta also uses Sentence Order Prediction (SOP) rather than NSP that helps in achieving the goal better.

**4.3 ROBERTa (Robustly Optimized BERT)**

Roberta is developed by Facebook AI as an improved version of BERT. Roberta differs from BERT in the following aspects.

**Training Data:** Roberta uses 160GB of text for pre-training which is more than what BERT is trained on.

**Pretraining Duration:** Training Process of Roberta is longer and uses more Optimization Steps which contributes to improved performance.

**Training Batch Size:** Roberta uses larger batch sizes during training which helps in achieving more stable gradients and better model convergence.

**Byte-Pair Encoding:** Roberta uses Byte-Pair Encoding for tokenization which is more efficient to handle out-of vocabulary words compared to Word Piece tokenization used by BERT.

1. **Results**

**Data:**

We Perform this experiment on SQuAD2.0 Dataset. SQuAD 2.0 combines the 100,000 questions from SQuAD 1.1 with more than 50,000 new, unanswerable questions. These unanswerable questions were generated to resemble answerable ones, with the goal of making it more challenging for models to identify whether a question can be answered based on the provided context. The dataset is based on more than 500 Wikipedia articles, and each article is divided into multiple paragraphs. A set of questions is associated with each paragraph, and the correct answers can be found as spans of text within the paragraph. The dataset is commonly used for evaluating reading comprehension and question-answering models in the field of natural language processing (NLP).

Train Data (5000 Samples): from Official SQuAD 2.0 training set

Dev Data (500 Samples): Sampling from Official SQuAD 2.0 dev set

**Experiments & Performance Evaluation:**

To Evaluate the models two metrics are used are EM Score and F1 Score:

**Exact Match:**

For each Question + Answer pair, if the characters of the model prediction exactly match with the ground truth the EM score will be 1 or else it is 0. EM is a strict metric that returns value as 0 even if the prediction differs from the ground truth by one character.

**F1 Score:**

F1 Score is a classification metric based on Recall and Precision. When both FP and FN are of important, we use F1 Score as a metric. In this Case, it is computed over the individual words in the prediction against the ground truth. Precision is the number of shared words to the total number of words in prediction. Recall is the ratio of number of shared words to total number of words in ground truth.

**Experiment Details:**

**Distil BERT:**

We are using Distil BERT uncased as we would like to treat the upper case and lower-case letters as the same which would reduce the model complexity and helps in generalizing better. Training the model on 5,000 data points and evaluation on 500 steps. Number of epochs to train 5, Name of dev metric used to determine best check point F1 Score, batch size 16, Number of look ahead steps=5, Optimizer used is Adaboost with a learning rate of 1e –4, Training Process took around 20 minutes.

**Alberta:**

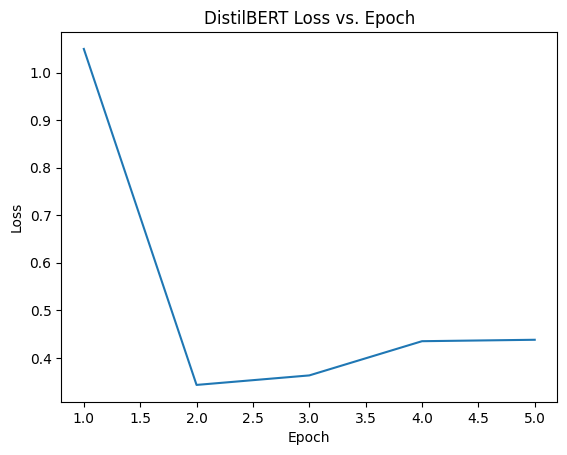
We are using Alberta-base-v2 which is the second version of the base model. Version 2 has improved dropout rates, additional training data, and longer training. It has 12 repeating layers, 128 embedding dimension, 768 hidden dimension, 12 attention heads, 11M Parameters. Training the model on 5,000 data points and evaluation on 500 steps. Number of epochs to train 5, Name of dev metric used to determine best check point F1 Score, batch size 16, Number of look ahead steps=5, Optimizer used is Adaboost with a learning rate of 1e –4, Training Process took around 10 minutes.

**Roberta:**

Roberta-base is the base version of the Roberta model. Training the model on 5,000 data points and evaluation on 500 steps. Number of epochs to train 5, Name of dev metric used to determine best check point F1 Score, batch size 16, Number of look ahead steps=5, Optimizer used is Adaboost with a learning rate of 1e –4, Training Process took around 30 minutes.

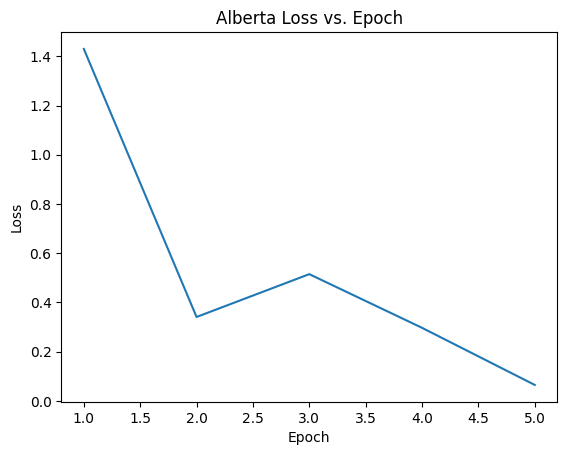
* 1. Distil BERT

|  |  |  |
| --- | --- | --- |
| Name | Split | Description |
| EM | Dev | 48.34 |
| F1 | Dev | 38.72 |



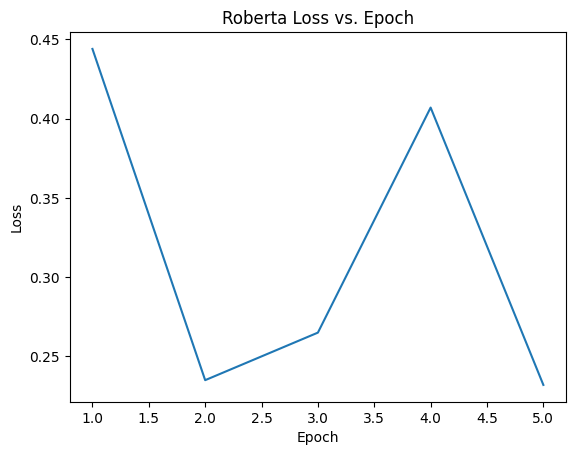
2. Alberta

|  |  |  |
| --- | --- | --- |
| Name | Split | Description |
| EM | Dev | 52.34 |
| F1 | Dev | 37.81 |



3. Roberta

|  |  |  |
| --- | --- | --- |
| Name | Split | Description |
| EM | Dev | 65.43 |
| F1 | Dev | 53.27 |



**Results Analysis:**

The best model to solve Question Answering System on SQuAD 2.0 Data is found to be Roberta with an F1 Score of 53.27 and EM score of 65.43. The model results are compared through running and evaluating the models on same number of training samples. The performance would further increase considering training the models on large number of training samples and increasing the Hyperparameter combinations. Considering the Computation limits of the GPUs present in the systems we trained the models on different sets of training samples like 100,500,1000,5000. The Final decision of considering 5000 samples is from the limitations of computation.

Roberta model is trained on more dataset and found to be more robust than BERT and ALBERT.

**Discussion:**

The current model implemented in our project give a reasonably good EM and F1 score when compared to human evaluation. As part of future scope of the project, we plan to incorporate hyperparameter tuning of the above Transformer models that may lead to improvement in the model performance with the defined metrics. Another aspect we would like to look into is training the models with higher computational power and observe the time complexity of the models and think of a trade-off between model performance and computational time.

1. **Conclusion**

The goal of this project was to develop an extractive question answering system by utilizing the state-of-art transformer models. The model was trained on the SQuAD 2.0 dataset that comprised a total of around 150000 answerable and unanswerable questions posed by humans. The train set was pre-processed by cleaning the data followed by tokenization of the text data. The trained models were then evaluated based on the EM score and F1 score metrics, where we could see that RoBERTa model outperformed ALBERT and DistilBERT by over 20% in each metric. This can be attributed to the longer training time of the RoBERT model as compared to the other models. This allows it to capture more contextual relations and language patterns. Whereas ALBERT and DistilBERT have been designed to be efficient computationally.

Overall, our project looked into the importance of using transformer-based models for a specific downstream NLP task like Question Answering. We look at the importance of selecting the appropriate model based on model performance, and available resources. These Language models have the power to alter the ways we interact with the available textual data in the form of Question Answering to conversational Chatbots.

1. **References**
2. Rajpurkar, Pranav, Robin Jia, and Percy Liang. "Know what you don't know: unanswerable questions for SQuAD." *arXiv preprint arXiv:1806.03822* (2018).
3. Pearce, Kate, et al. "A comparative study of transformer-based language models on extractive question answering." arXiv preprint arXiv:2110.03142 (2021).
4. Rajpurkar, Pranav, et al. "Squad: 100,000+ questions for machine comprehension of text." *arXiv preprint arXiv:1606.05250* (2016).

\*Each team member in our group has contributed equally to the success of this project.