Report

Data processing

Tokenization: WordPiece

- BERT commonly uses a variant of Byte Pair Encoding (BPE) known as WordPiece for tokenization. This approach allows the model to break down words into smaller subwords, enabling it to handle out-of-vocabulary words more effectively.
- Additionally, BERT uses unique tokens for specific functions:
  - `[CLS]`: Added at the beginning of each sequence; used for classification tasks.
  - `[SEP]`: Separator token; used to distinguish different sentences in tasks that require understanding of multiple sentences.
  - `[PAD]`: Padding token; used to make all sequences in a batch the same length.
  - `[MASK]`: Mask token; used in the training phase where random tokens are replaced by this token.

Conversion of Answer Span Positions

In the BERT model, text is tokenized into subwords or word pieces, which means that the original character positions in the text do not directly correspond to the token positions after BERT tokenization. To align the answer span's start and end positions with the tokenized output, we use the following approach:

1. **Tokenization**: We first tokenize the text using BERT's tokenizer, which also provides us with the mapping between the original text and the tokenized text.
2. **Offset Mapping**: The mappings indicate each token's start and end character positions in the original text, which converts the answer span's start and end positions from character-based to token-based indices.

Determining Final Start and End Positions

After the model predicts the probabilities for each token being the start and end of the answer span, we apply the following rules to determine the final start and end positions:

1. **Probability Threshold**: We set a minimum probability threshold for a token to be considered as a potential start or end position. Tokens with probabilities below this threshold are ignored.
2. **Span Length Constraint**: We set a maximum allowable length for the answer span. If the end token is too far from the start token, that pair is considered invalid.
3. **Maximization**: Among the remaining valid start and end token pairs, we choose the pair that maximizes the sum of the start and end probabilities.

Modeling with BERTs and their variants

This project aims to tackle two main tasks in the realm of Chinese extractive question answering: paragraph selection and span selection. While our paragraph selection model, fine-tuned on a Chinese Pre-trained BERT, achieves an impressive accuracy of 93.9%, the span selection model lags behind with a 75.6% accuracy. This discrepancy underscores the need for further optimization in span selection. To address this, we explore two different model configurations, drawing inspiration from recent advancements in the field (Read More).

Baseline: Chinese Pre-trained BERT

In this project, we employ a BERT (Bidirectional Encoder Representations from Transformers) model pre-trained specifically for the Chinese language. The model has undergone training with random input masking applied independently to word pieces, making it highly effective for various NLP tasks in the Chinese context. In the end, we reach an accuracy of 75.6% with the setting below.

Loss Function:
We use the Cross-Entropy Loss function as the objective to train our model. It is particularly effective for classification problems and works well with the softmax activation in the output layer of the model.

Optimization Algorithm:
We use `torch.optim.AdamW`, which is an extension of the classic Adam optimizer with weight decay. AdamW has proven effective for training neural networks.

Hyperparameters:

- Learning Rate: 3e-5

- Batch Size: 8

Variant: Chinese Pre-trained BERT with WWM and RoBERTa Techniques

To further validate the effectiveness of our primary BERT model, we also experimented with additional techniques proven to be effective in boosting accuracy under the same setting of loss function, optimization algorithm, learning rate, and batch size.

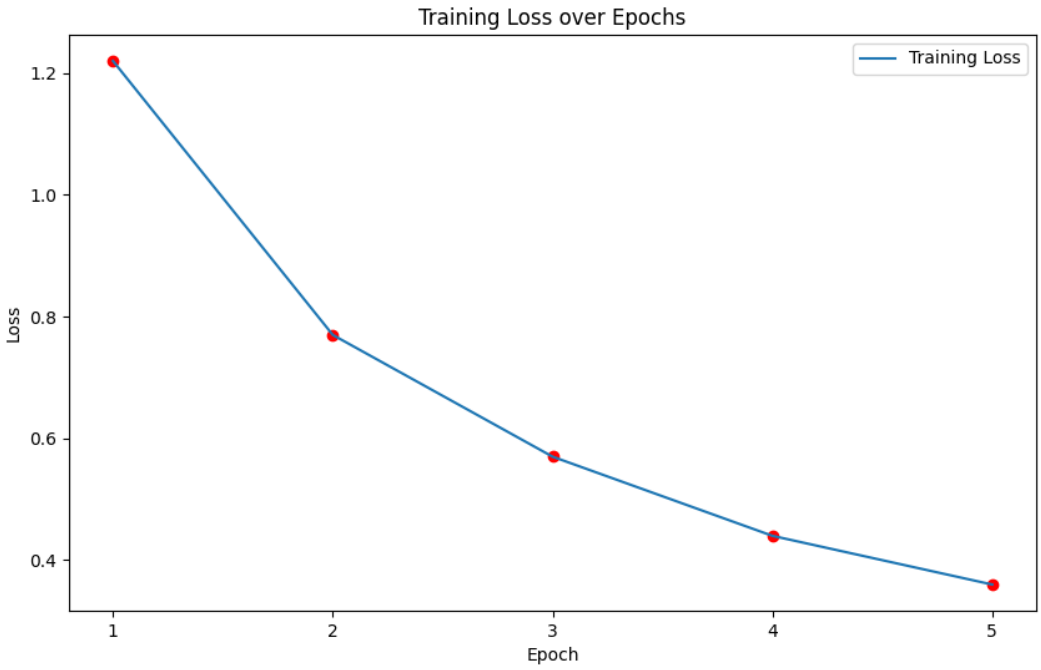
- **WWM:**
  This variant of BERT incorporates WWM (whole word masking), which masks entire words instead of random word pieces. This approach is particularly beneficial for logogram languages like Chinese, where the context of a signal character can change abruptly after concatenating with another.
- **RoBERTa:**
  RoBERTa (Robustly Optimized BERT Pretraining Approach) is another variant of BERT that has been optimized for more robust performance. It modifies key hyperparameters in BERT, including removing the next-sentence pretraining objective and training with much larger mini-batches and learning rates.

Comparative Performance

This enhanced BERT model yields an even better performance, achieving an accuracy of 80.7% on the test dataset. This is a significant improvement over the basic BERT model and validates the effectiveness of WWM and RoBERTa techniques in the context of Chinese language processing.

Learning curve

We later plot out the baseline BERT model's loss curve to visualize the training process's effectiveness.



Pre-trained vs Not Pre-trained

We compared the same BERT models with and without pre-load weights for span selection tasks. Both models were trained for 4 epoch and used the same settings, including the cross-entropy loss function, the Adam optimization algorithm, a learning rate of 3e-5, and a batch size of 8. The pre-trained model significantly outperformed the non-pretrained model, achieving an accuracy of 75.6% compared to 12%.