ChatBot for goal planning according to the "Abundant Life Planning" framework

Tamara Ilić SV45/2020

1. Motivation

This spring, I had the opportunity to participate in an entrepreneurship fellowship in New York State and spend two weeks at the University of Buffalo. During my time there, I attended lectures by Professor Bob Neubert. One of his lectures focused on life planning, specifically his "Abundant Life Planning" framework. I was inspired by this concept and believed it could significantly impact students by helping them set a clear direction for their lives and understand their goals. Recognizing that students might need guidance to create their Abundant Life Plan, I thought a question-answering bot could be highly beneficial.

2. Research questions

The goal of my project was to develop a question-answering bot specialised in the Abundant Life Planning framework. This bot aims to assist students in creating their plans more easily and effectively. The dataset for this project was derived directly from Professor Neubert's lectures, presentations, and additional resources I obtained during his classes. I compiled a set of 214 questions, each with corresponding context and answers based on specific paragraphs from the gathered materials. The questions were designed to test the model's understanding of the context, with some questions being contextually similar but phrased differently to ensure robust learning.

3. Related work

In recent years, several question-answering models have been developed using various natural language processing techniques. BERT (Bidirectional Encoder Representations from Transformers) has been a significant advancement in this field, showing remarkable performance on the SQuAD (Stanford Question Answering Dataset) benchmark. Fine-tuning pre-trained models like BERT on specific datasets has been a common approach to improve performance on domain-specific tasks. However, these models typically require large amounts of training data to achieve high accuracy.

4. Methodology

Data Gathering and Preparation

I collected data from Professor Neubert's lectures and compiled it into a comprehensive document. From this document, I formulated 214 questions, each linked to a specific paragraph containing the answer. To enhance the model's contextual understanding, I included questions with the same answers but different phrasing.

Model Choice

Initially, I intended to use the ‘bert-base-uncased’ model. However, during training, it yielded poor results, with an exact match score of 0 and an F1 score of 20. Realising the need for a larger dataset for effective training, I switched to the ‘google-bert/bert-large-uncased-whole-word-masking-finetuned-squad’ model, which is pre-trained on the SQuAD dataset and larger than the base model.

Fine-Tuning

Fine-tuning this model involved several preparatory steps. Since BERT is token-based, I used a tokenizer to convert each sample into tokens, marking the answer within the context. I configured the tokenizer with a maximum context length of 384 tokens, utilising a sliding window technique to ensure the answer was not truncated. I set up an optimizer with a learning rate of 2e-5 and divided the data into batches of 4. The training was conducted over 10 epochs on a training set comprising 75% of the dataset, which was shuffled and stratified by themes.

5. Discussion

To evaluate the model, I computed the F1 score for partial overlaps of the answer and the exact match metric. The best results obtained were as follows:

```
{'exact_match': 40.74074074074074, 'f1': 66.14986810339855}
```

Initially, I used the 'bert-base-uncased' model with the same hyperparameters, but it failed to learn effectively due to the small size of the dataset. Research indicated that successful implementations of this model typically utilised datasets with hundreds of thousands of samples, achieving F1 scores around 0.8.

To address this issue, I discovered that for smaller datasets, a useful approach is to fine-tune the model first on a general question-answering dataset, followed by fine-tuning on the specific domain. Consequently, I used the ‘google-bert/bert-large-uncased-whole-word-masking-finetuned-squad’ model, which is pre-trained on the SQuAD benchmark dataset. I added a final layer for my specific fine-tuning and dataset.

Initially, I created tokens from the entire paragraph of text containing the answer. This approach resulted in poor understanding as the model struggled to identify the specific context needed for accurate answers. To improve this, I switched to truncating the context to 384 characters with a sliding window to ensure the answer was not cut off.

I experimented with different batch sizes and epochs. Training on a larger batch size of 16 resulted in underfitting, while a batch size of 4 yielded better results. Initially, I trained the model for 3 epochs, and then extended to 10 epochs, which provided slightly better results.

In conclusion, training the BERT transformer model effectively requires a large dataset. For smaller datasets, using a pre-trained model on a generalised dataset like SQuAD, followed by specific domain fine-tuning, proves to be an effective strategy.

6. References

Pretrained Model: [Huggingface BERT Large Uncased]

Dataset: Google Sheets Dataset
Hosted Model: Huggingface Model