Question Answering Chat Bot Using the SQuAD dataset

Andrew Frederick, Mike Moll, Manahil Nasim

AAI 520: Natural Language Processing

Professor Roozbeh Sadeghian

10/23/23

Abstract

This comprehensive report details the development of a chatbot, which was designed to provide coherent responses to user queries, using the Stanford Question Answering Dataset (SQuAD) as the primary data source. The report covers the extensive process of data selection, preparation, model architecture, training, model evaluation, and offers valuable concluding remarks. The objective of this project is to create a chatbot capable of providing accurate and informative responses to user queries on a wide range of topics. Overall, this project demonstrates the feasibility of developing intelligent and effective chatbots using large-scale language models and the SQuAD dataset. Such chatbots have the potential to revolutionize the way we interact with information and technology.

Introduction and Background Research

NLP, or natural language processing, is the field of AI that covers the processing and output of human language. One prominent application of the field is the ability to create chat bots, or computer interfaces that can interact with a human user by responding to conversational messages. The technology represents exciting advances across multiple industries like customer service and search engines, as well as revolutionizing the way we interface with technologies.

To create a chatbot that can answer user questions accurately, it is important to use a robust dataset that tests reading comprehension and general knowledge. The Stanford Question Answering Dataset (SQuAD) is a valuable resource for this purpose. SQuAD consists of question-answer pairs based on Wikipedia articles, which makes it a good representation of the types of questions that users are likely to ask a chatbot.

In addition to being comprehensive and well-structured, the SQuAD dataset is also challenging. This is because the questions often require the chatbot to understand the context of the article and reason over multiple sentences to generate the correct answer.

In order to build a chat bot we first set out to understand the base architectures that are being used. Transformers are models that use encoders and decoders to input and output sequential data. As such, they are well suited to the conversational aspect of a chat bot, that needs to read text strings and respond in a similar fashion. These models also excel over other deep learning architectures in keeping track of context in a multi-exchange conversation. This allows for more subtle intricacies of human language to be captured, making Transformers the leading method for most NLP Tasks today (Singh, 2023). For example, GPT or generative pre-trained transformer, is a specific transformer model developed by OpenAI and made available to the public through Chat GPT. Its utility for text summarization and generation, image creation, and even coding has made headlines and started a wider dialogue on the future effects of AI chatbots (VanBuskirk, 2023).

Data Selection and Preparation

For this project we first considered the Stanford Question Answering Dataset (SQuAD) which contains question answer pairs based on over 500 wikipedia articles designed to test reading comprehension (Stanford University, 2019). We felt that the variety of topics covered by the SQuAD dataset and the set format would be well suited to create and chat bot designed to answer user queries as best as possible, and decided to base this project on this dataset.

The preprocessing steps for the SQuAD dataset involved:

Tokenization: The context and questions were tokenized into subwords.

Padding: Padding is the process of extending text sequences that are shorter than the defined "MAX_SEQ_LENGTH" by adding special tokens, often referred to as padding tokens. This step ensures that all input sequences are of equal length, thereby enabling efficient parallel processing by the model.

Truncation: Truncation is applied to text sequences that exceed the defined "MAX_SEQ_LENGTH." In such instances, the "input_ids" are truncated, meaning that portions of the text are removed to meet the specified sequence length. This is a necessary operation to prevent memory overflows and to ensure that the input adheres to the model's constraints.

Answer localization: The start and end positions of the answer in terms of subword tokens were located.

The resulting processed data consisted of structured examples suitable for training a chatbot.

Model Architecture and Training

Model selection: To initialize the process of creating the chatbot we employed the BERT model's bidirectional nature and its pre-trained knowledge which makes it suitable for understanding context and providing accurate answers to questions. We employ the prebuilt BERT model, "bert-base-uncased," fine-tuned for question answering using the SQuAD dataset. The model is loaded and moved to the appropriate device (GPU if available) to accelerate computation.

Training Parameters:

The model is trained over multiple epochs, with hyperparameters such as batch size and learning rate configured to optimize performance. In our experiments, a batch size of 32 and a learning rate of 5e-5 were utilized. These values were fine-tuned for the best balance between training efficiency and model convergence.

Training Loop and Checkpoints:

The training loop iterates over the dataset for the defined number of epochs.A checkpoint of the model's state and optimizer parameters is saved at the end of each epoch to allow for resuming training and evaluating the model's performance at different points.

Model Evaluation

Our trained BERT model achieved an F1 score of approximately 0.166 and an exact match score of around 0.128. Our findings suggest that while BERT serves as a powerful model for a wide range of NLP tasks, question answering remains a challenging problem, and further research is needed to achieve higher accuracy.

Chatbot Interface

The foundation of the "Quanda" chatbot lies in its chat history, which keeps track of the conversation. When a user interacts with the chatbot, their inputs are appended to this chat history, allowing the chatbot to understand the context of the conversation. The chat history begins as an empty list.

Chatbot Function:

The central component of "Quanda" is the chatbot function, which is responsible for processing user inputs, generating responses, and updating the chat history. GPT-3's API is utilized to obtain text completions based on the chat history.

Chatbot Responses

The chatbot function follows a clear sequence of operations:

User Input: The user's input is added to the chat history.

Generating a Response: GPT-3 is called to generate a response. The chat history, which includes both user inputs and chatbot responses, serves as the prompt.

Response Length: The response length is governed by the 'max_tokens' parameter, allowing control over the length of generated responses.

Gradio Interface

Gradio, an interactive interface creation library, is employed to design the interface for "Quanda." The interface consists of two text boxes: one for the user to input queries and another to display chatbot responses. Additionally, the conversation history is displayed in a third text box to provide users with a context of the ongoing conversation.

Conclusion and Remarks

The project's foundation is based on the Stanford Question Answering Dataset (SQuAD), a challenging yet valuable resource for training chatbots. "Quanda" leverages large-scale language models, particularly the BERT architecture, for understanding context and generating precise answers to questions.

The creation of "Quanda" demonstrates the feasibility of developing intelligent and effective chatbots using advanced NLP models and structured datasets. By employing BERT and fine-tuning it for question answering with SQuAD, we have achieved a significant milestone in the field of chatbot development.

The development of "Quanda" was not without its challenges. While BERT, with its bidirectional capabilities and pre-trained knowledge, provided a strong foundation, achieving high accuracy in question answering remains a complex problem. Our trained BERT model achieved an F1 score of approximately 0.166 and an exact match score of around 0.128. These results emphasize the need for further research to improve chatbot performance.

Future Improvements

Fine-Tuning: The performance of "Quanda" can be enhanced through extensive fine-tuning. Exploring different fine-tuning techniques, hyperparameters, and model architectures could yield substantial improvements in accuracy.

Conversational Context: Enhancing the chatbot's ability to maintain conversational context over multiple exchanges is essential. Investigating dialogue models and context-aware architectures can lead to more coherent conversations.

In summary, the development of "Quanda" represents a significant step in the evolution of chatbot technology. It showcases the potential of large-scale language models and structured datasets to create effective conversational AI. However, there is ample room for improvement, and ongoing research and development will be essential to unlock the full potential of chatbots in diverse applications and industries.

References

Chidananda, R. (2018, March 28). *Cornell movie-dialog corpus*. Kaggle. https://www.kaggle.com/datasets/rajathmc/cornell-moviedialog-corpus

Singh, A. (2019, June 3). *Why do transformers yield superior seq-to-seq results?*. Medium. https://medium.com/saarthi-ai/transformers-attention-based-seq2seq-machine-translation-a28940aaa4fe

Stanford University. (2019). *Stanford Question Answering Dataset*. Kaggle. https://www.kaggle.com/datasets/stanfordu/stanford-question-answering-dataset.

Tatman, R. (2017, August 16). *Ubuntu dialogue corpus*. Kaggle. https://www.kaggle.com/datasets/rtatman/ubuntu-dialogue-corpus

VanBuskirk, A. (2023, April 25). *A brief history of the generative pre-trained transformer (GPT) language models*. WordBot. https://blog.wordbot.io/ai-artificial-intelligence/a-brief-history-of-the-generative-pre-trained-transformer-gpt-language-models/