

MAIS 202 - Project Deliverable 2

GitHub: <https://github.com/jingxiangmo/Azza>

Problem Statement

The objective is to create a conversational agent that can provide reliable and relevant answers to user questions. We hope to finetune using the SQuAD dataset and further improve our finetuning using Google LaMDA if time allows.

Milestones

MVP I (complete): Simple contextualized question-answering conversational agent deployed on Hugging Face.

MVP II (complete): Training and fine-tune on SQuAD.

MVP III: Create a functional web application for our conversational agent.

MVP IV: Further fine tuning with Google LaMDA.

Data Preprocessing

Stanford Question Answering Dataset (SQuAD)

<https://rajpurkar.github.io/SQuAD-explorer/>

Key-phrase extraction model

<https://huggingface.co/ml6team/keyphrase-extraction-kbir-inspec>

Wikipedia API

[Wikipedia-API · PyPI](#)

We are using the Stanford Question Answering Dataset (SQuAD), a diverse dataset containing over 100,000 questions and answers, for training and evaluating machine reading comprehension tasks. We use SQuAD to select a quote from an online context and answer questions. This method allows us to build a conversational agent that finds the accurate context.

When the user enters a question, a key-phrase extraction model is used to find the sentence's most crucial word or group of terms. The Wikipedia API is then used to look up the closest article to that key phrase to obtain the article summary: this is the context where the model will find the answer to the original question.

Machine Learning Model

BERT-large (Model):

[bert-large-cased-whole-word-masking-finetuned-squad · Hugging Face](#)

Frameworks/tools used: Pytorch, Gradio, HuggingFace

The feedback we received for deliverable 1 made us realize that building a chatbot is challenging. The material covered in MAIS 202 is insufficient; therefore, a “simple” logistic regression model would not cut it.

The model we use and finetune is BERT-large: An unsupervised learning transformers model pre-trained on a large corpus of English data. Regarding its architecture, the model is composed of 24 layers and 1024 hidden dimensions for a total of 336 million parameters. Given a query and a context, it finds the answer to the question in the context given. In our project, context is found at the end of the preprocessing step.

We are not finetuning the BERT model ourselves (for the moment). We use a model that has already been fine tuned on SQuAD and can perform well for new questions too.

b) Please see BERT-large documentation.

c) Please see BERT-large documentation.

d) The first challenge was finding a context to the question that was asked for it to be answered by the model. This is why we had to come up with the keyword extraction + Wikipedia API idea. This created another issue to be fixed: sometimes Wikipedia summaries are too long and specifically longer than 512 tokens, the upper limit of the Bert_tokenizer trainer on SQuAD we used.

Preliminary Results

Please see BERT-large documentation.

Next Steps

Once our model is fine-tuned and performant enough, we want to implement it on a functional web application. We have already started working on this using Gradio, which is an easy way to demo our model with a friendly web interface.