Spoken Language Processing - Dialogue Systems Assignment

Instituto Superior Tecnico
Group 23
Daniele Avolio ist1111559, Eduardo Rodrigues ist1111684

1. Introduction

This lab assignment focuses on the development of a spoken dialogue system using state-of-the-art models for automatic speech recognition (ASR), Large Language Models for question answering (QS) and text-to-speech (TTS). Specifically, we explore the integration of models from the HuggingFace Transformers library, including OpenAI Whisper for ASR, LaMini-GPT2 for language understanding and generation, and SpeechT5 for TTS. During the development we tried different strategies that will be discussed in the following sections. Specifically, the main problems occurred during the question answering task, because we couldn't get the correct responses for the dataset. In the next sections we will discuss the strategies we tried and the results obtained. It's important to mention that we ran everything on Google Colab, since we didn't have the necessary hardware to run the models on our own machines. This was nice but also a bit slow, since we had to wait for the models to be loaded every time we wanted to run the code and download the models every time we wanted to use them. Moreover, we coulnd't get access to Llama3 since we didn't get accepted in a reasonable time.

2. OpenAI Whisper Task

The first task was pretty easy because it involved the use of OpenAI Whisper model, a state-of-the-art model for ASR. The model was used to transcribe some audios file that we needed to record using our own voce.

The main evaluation metric used for this task was the Word Error Rate (WER), which is defined as the sum of the number of substitutions (S), deletions (D), and insertions (I) divided by the total number of words in the reference transcript (N). The formula is given by:

$$WER = \frac{(S + D + I)}{N} = \frac{(S + D + I)}{(S + D + C)}$$

What we did was only to record 2 audios with our phones and then we used the model to transcribe them. The results were pretty good, but we noticed that using the computer microphone that's integrated in the laptop, the results were very bad, since the audio quality was very low. We didn't try to use a bigger model in this task, since the results were already good enough using the base model. However, later in the assignment we tried to use a bigger model for the question answering task, since the small model was not able to give us the correct answers.

The final result showed that a perfect WER was achieved for the two audios, since the model was able to transcribe them correctly.

There is not much to say about this task, since it was very straightforward and we didn't have any problems with it.

3. LLM for Conditional Language Generation

In this section we discuss the task that involved the use of a Large Language Model (LLM) for conditional language generation. Initially we tried to follow the instruction given in the assignment using GPT-2 model, but we couldn't get the correct answers for the dataset since the model is not powerful enough to understand the questions in a way that it can give the correct answers. Even for this, we tried to give the model a more sophisticated prompt, guiding it in the right direction of answering in a very short and brief way. However, the results were still wrong.

Some of the prompts that we used are:

"Answer the Question based on the Context and keep the
Answer very short and straightforward
Context: {*row['context']*}
Question: {*row['question']*}
Answer: The answer is:"

We then tried to use a bigger model, LaMini-GPT2 and other models of the same LaMini family, like Flan or T5. With this we even tried to explore more prompting strategies, like few-shot learning, for example:

"Answer the Question based on the Context and keep the
Answer very short and straightforward
Follow this example:
What is the capital of Portugal?
Lisbon
Context: {*row['context']*}
Question: {*row['question']*}
Answer: The answer is:"

However, the results were still wrong. We then tried to use bigger models, for example: LaMini-GPT-124M,LaMini-GPT-774M,LaMini-GPT-1.5B,LaMini-T5-738M,LaMini-Flan-T5-783M However this didn't change the results, since the answers were still wrong. As we stated in the initial part of the report, we couldn't get access to Llama3, which is a way bigger model for which the smallest version has 8B parameters. We believe that using this model would give us the correct answers, since it's way bigger than the models we tried.

However, in the last days we found another way to access LLama3, but using Google Colab we were not able to run the full model, so we used a quantized version of it: *"unsloth/llama-3-8b-bnb-4bit"*. The model was performing much better than the other models we tried, but still was not able to give us the answer in a straightforward way. We believe that this is due to the fact that the model is quantized and has

only 4 bits, so it's not able to perform as well as the full model. Moreover, to achieve a better performance we would need to fine-tune the model, but we didn't have the time to do it since we only found the model in the last days of the assignment and we still needed to write the report and complete the other tasks.

The final prompt used for the model was:

"You will answer the next question based on the provided context. Provide a very direct response without adding any additional information. Use a maximum of 4 or 5 words. Do not include any other explanations.
Context: {row['context']}
Question: {row['question']}."

For this motivation, we didn't run the model on the full dataset and didn't compute the BLEU score of it, since it was useless. For example, for the first 10 questions of the dataset, using a n-gram of 2, the BLEU score was 0.03.

3.1. Honorable Mention

We tried to solve the problem using a different approach, which was to use *Roberta* model, and this was working very well! We even applied some text preprocessing to the output of the model, since the model was giving us the answer a *capitalized* format, and the dataset was in a *lowercase* format, so we just lowercased the output of the model. The problem is that the model is working more torwards the extraction of the information from the context, rather than generating the answer. This is a problem because later on we should use the same module without context, and this would not work. Even so, Roberta was able to achieve a BLEU score of 1.0 for the first 10 questions of the dataset, using a n-gram of 2. However, because of the problem mentioned above, we didn't evaluate the model on the full dataset.

4. SpeechT5 Task

In this task we were asked to use the SpeechT5 model to generate speech from text in the initial example. However, the entire task longer than this. The first thing to do was to generate speech from the 5 answers of the previous task. We used the SpeechT5 model to do this, and the results were good, and the audio was understandable. Of course, the problem was in the previous step because the answers included useless information.

The next step was to get the text from the audio of the *TriviaQA* dataset, in particular the*SQA-5 split*. This step took a lot of time since the dataset was very big and we had problems with Google Colab. To solve this we implemented a script that was requesting a *json* containing the information of the dataset with the download link of the audio files, other than the question, context and answer. We downloaded the audio files locally and then we used the OpenAI Whisper model to transcribe them. The results were not so good since the audios were not very clear, and in this case we used the bigger model *Whisper-Medium*. The model is very heavy, in fact for 10 audios of 5 seconds each, circa, the model took 4 minutes to transcribe them. The transcription even in this case was not perfect, in fact the the *llama-3-8b-bnb-4bit* model was not able to answer the questions correctly. Even in this case we tried to use different prompting solutions, changing the order of the context, question and answer, but the results were still wrong. For the final step, we needed to record our voice for the first 2 questions of the TriviaQA dataset, use Whisper for the transcription and then use SpeechT5 to generate the audio. As for the other audios, the results were good and the transcription was pretty much correct, but the problem was always in the asnwer generation. Even in this case we are not providing a table with the benchmark of the model, since all the answers were not respecting the format of the dataset. However, if we needed to evaluate the general correctness of the asnwers, they were not incorrect at all, so the model was kind of working.

4.1. Note on the speech generation

Since all students received the tip that the audio generation model that we used is trained on text data and not on numerical data, we created a custom function that was replacing the numbers with the corresponding words. This was done using the *num2words* library. The functions returns the text with the numbers replaced with the words, and then we used the SpeechT5 model to generate the audio. The results were good, and the audio was understandable otherwise the numbers would have been read as they are written and the audio would have generated just jibberish.

5. Main tasks

Here we had to join together the previous tasks and create a spoken question answering system, integrating speech recognition, language understanding and generation, and text-to-speech models. Here the difference is that we must keep the context from previous questions and answers. What we did was a pretty much naive approach, since we ran out of time and we couldn't get the actual system to work. We just created a very similar pipeline to the one we used in the previous tasks, but we added a part where we were keeping the context of the previous questions. We noticed that if we continued to use the same context for the next questions, the answers were not getting better, so we tried to change the context every 5 questions, removing the oldest one and adding the new one. However, this didn't change the results, since the answers were still wrong. We believe that the problem is in the question answering model, since the other models were working pretty well. If the whisper model was not working, we could have used a bigger model, but the problem is that the model is working pretty well, so we don't know how to solve the problem.

The main thing would have been to use a better LLM and not a quantized version of it, but we didn't have the time or the resources, other than the direct access to LLama3.

6. Conclusions

Overall, we are not satisfied with the results of the assignment, since we were not able to get the correct answers for the dataset. We believe that if there was more time we could have tried different approaches, like fine-tuning the models, or using a bigger model for the question answering task. We are happy with the results of the other tasks, since we were actually getting positive results from the speech generation and recognition tasks. However, the main task was not working, and this is a big problem since that was the most important task. We believe that the best strategy here is to fine-tune the models, especially the *llama-3-8b-bnb-4bit* model, since it was the one that was performing the best among all the other question answering models. Using the training data from the TriviaQA dataset, we could have achieved better results, but it's not possible to do this in the remaining time.