# Laboratory Assignment 3 - Dialogue System

*Filipe Vaz (100520), Tomás Arêde (103239)*

[1]Instituto Superior Técnico, Speech Language Processing, Group 10

filipe.vaz@tecnico.ulisboa.pt, tomas.arede@tecnico.ulisboa.pt

## Abstract

This work presents the development of a multimodal spoken dialogue system that integrates speech recognition, natural language understanding, and text-to-speech synthesis. Utilizing models from the HuggingFace Transformers library, we implemented and evaluated various components, including Whisper for speech-to-text, multiple large language models for conversational intelligence, and Microsoft SpeechT5 and Kokoro for speech synthesis. Our exploration included performance comparisons across different architectures such as GPT2, Qwen, and LaMini variants using metrics like BLEU and WER. OpenAI Whisper Large was the model that showed better transcription power. Qwen1.5-0.5B-Chat was the text-generation model that showed, in our opinion, to be more capable of delivering a human-like conversation while being decent on QA factual answers and still being a good trade-off between performance and computational cost. For the speech synthesis the Kokoro model shower better performance. The final chatbot system includes history tracking, web search integration, and a user-friendly interface, supporting both voice and text interactions. Our findings highlight the trade-offs between model accuracy, inference speed, and user experience in real-world chatbot deployment.

**Index Terms**: speech recognition, human-computer interaction, computational paralinguistics

## 1. Introduction

For this lab assignment we developed a simple (spoken/conversational) question answering system, reusing different models associated to the HuggingFace Transformers library: speech recognition model, large language model for natural language undestanding and generation and text-to-speech models. The final goal is to join all of these and create the best chat bot we can.

## 2. Speech Recognition Model

We started by recording our voice saying two different sentences: "My fat dog is 9 years old." and "Sally sells sea shells on the sea shore," an easy one and a hard one. Using the model *openai/whisper-tiny.en* and evaluating the output with WER we got a perfect match for the first example but a WER of 0.25 for the second one. This makes sense since the output was "Sally Salcy, shells on the sea shore." and looking to the formula $WER = (S+D+I)/N = (S+D+I)/(S+D+C)$ where $S$ is the number of substitutions, $D$ is the number of deletions, $I$ is the number of insertions, $C$ is the number of correct words, and $N$ is the number of words in the reference (N=S+D+C) we can understand the result. To solve this second we now used the model *openai/whisper-large.en* where we got a perfect match.

## 3. LLMs for conditional language generation

The options of models that we could try are a lot. We tried three different types of LLM models: text generation, sequence-to-sequence and extractive QA. For our purpose of creating a normal chat bot, the text generation models are the most interesting models. However, learning never takes space, so we also tried for the other two types of models, but due to time constraints we only tried these last models for only the first 10 examples from the validation split from the TriviaQA dataset. For the text generation models we computed the BLEU error for the first 500 examples of the validations split from the same dataset. The results can be seen on the next table.

| | | **BLEU** |
|---|---|---|
| **Text Generation** | GPT2 | 0.0005 |
| | SmolLM-135M-Instruct | 0.0033 |
| | LaMini-GPT-124M | 0.0218 |
| | Qwen1.5-0.5B-Chat | 0.0671 |
| | LaMini-Cerebras-111M | 0.0147 |
| **Seq2Seq** | Flan-T5 | 0.7118 |
| **Extractive QA** | BERT-Large-Uncased | 0.8483 |

Table 1: *BLEU metrics for the different models used.*

First of all it is important to note that the BLEU metric is not the best to test the models, since the ground truths are often just a couple of words and the models answer with more than that, giving low BLEUs, meaning that even if the models are answering correctly or at least close to the topic, they will be heavily penalized. We can see that the model that achieved highest result was the extractive QA which makes sense, since it outputs also just a couple of words and it's made exactly for this type of questions and answers. This was followed closely by the sequece-to-sequence model that with the right prompt, also delievered answers similar to the ground truths. Regarding the text generation models, the best result was for Qwen and the worst for GPT2.

In general we can say that, for a general-purpose chatbot, text generation models are best due to their fluency and flexibility, handling arbitrary inputs and generating human-like responses, while extractive QA is ideal for fact-based queries and seq2seq suits structured tasks, like summarization, translation, task specific bots, being in general not as good for conversation. However, as we've seen on most text-generation models that we tested, they often hallucinate or give incorrect facts, meaning that we need to create a good prompt design to try to force them to stay on-topic.

For all tests, we instructed each model to respond briefly

(no more than 5 words), tailoring the prompts to suit each model's capabilities. Each model exhibited distinct behaviors:

- **GPT-2** struggled to follow complex prompts and frequently produced repetitive or hallucinated outputs, even with simple instructions. This behavior may suggest signs of overfitting or limited generalization. These observations are consistent with findings in studies on GPT-2 overfitting and hallucination patterns [1].
- **SmolLM** encountered similar issues, often returning empty strings, repetitive responses, or non-sensical outputs.
- **LaMini-GPT** showed greater alignment with prompts and produced more assertive and relevant responses. Even so, a lot of answers were either incorrect or didn't make sense, but for such a small model (124M parameters) when compared to bigger models like Qwen (0.5B parameters), it wasn't that far behind.
- **Qwen** demonstrated strong prompt adherence and consistently delivered more concise and accurate, or contextually appropriate responses. It wasn't perfect, but it was better than the previous.
- **LaMini-Cerebras** also followed prompts well and stayed on topic, but its answers were generally less accurate than Qwen's, often containing more factual errors.

The best results, not only in terms of the BLEU metric, but in terms of an overall decent response, were LaMini-GPT-124M, Qwen1.5-0.5B-Chat and LaMini-Cerebras-111M, with the second being way heavier than the other two. To test this, we created a simple interface of a chatbot for each of these models, to try to determine which was the best. On the conversation test we tried to give simple questions (for example "What is the capital of France?") and all the models answered correctly, but we also tried to have a human-like conversation, asking how he was, advices he would give to a certain position, ... and only Qwen was successful on this task. Since the goal of our group is to build a general and human-like chat bot we ended up choosing Qwen. If we wanted some specific bot, for a certain area, for example politics, football, etc... we would probably end up choosing one of the other two and fine-tuning it with a dataset of that subject, making the chatbot specialized in it. We tried to fine-tune the Qwen model for the CoQA dataset but it took a lot of time and we didn't have either the time or the computational resources for it. We tried with GPU, reducing the number of epochs, increasing the batch size and none of that reduced the training time for something that would be plausible. We ended up not fine-tuning the Qwen model.

With our LLM model decided, we implemented chat history tracking to use the previous conversations as context and also search the web, triggered by certain words, using Serper. We built a simple interface to test the chatbot experience and got very satisfactory results.

## 4. Text-to-Speech models

In order to produce speech outputs from the text generated by the LLM for the first 5 questions of the TriviaQA dataset, we used SpeechT5 text-to-speech model. We also experimented with Kokoro, an alternative to SpeechT5, and got a less distorted voice and a much more pleasing experience. This is likely because Kokoro is focused solely on TTS and uses high-fidelity neural vocoders, while SpeechT5, being a multitask model, often sacrifices clarity for flexibility. Kokoro's outputs sound cleaner, more natural, and better suited for real-time interaction.

We now implement the full chat-bot ecosystem, using OpenAI Whisper large to make the transcriptions, Qwen model to process them and the Kokoro to make the conversion of text-to-speech of the output of the LLM. We tested for the first 10 questions of the audio sample TriviaQA dataset and, using BLEU metric with a maximum order of 2-grams and considering the smoothing, we got a result of 0.02732, which is still low and represents a reduction of $\sim 60\%$ of the previous BLEU metric with Qwen. Overall the answers also look worse than the previous test, and in some cases even it even retrieved some chinese characters.

We then did the same for our recording of the first two questions: "To where did Ethel red flee?" and "of what is ozone a reactive part of oxygen", meaning that the Whisper model didn't identify the recording as a question. We ended up with the answers "Normandy" and "Ozone". The ground truths were "normandy" and "allotrope", meaning that for the first one the answer was correct and wrong for the second.

## 5. Chat-Bot Build

Using a frontend template provided by the AI software Claude, we created the backend using Flask where we implemented what we developed previously. Instead of using OpenAI Whisper Large (as we've seen that it led to the best results), we decided to implement the Whisper Base, a decision that would compromise the performance a little, in order to improve speed of processing. We chose the Base version instead of the Tiny, since it showed better perfomance and reasonable speed. To achieve reasonable speed with Large, it would require the use of a GPU, which is unreasonable to expect from the end-user. Regarding the rest, the interface allows the user to record its voice or write a message, uses the Qwen model to process the transcription/message and retrieves the output in the format of message with the possibility of being played with Kokoro. Besides that, we also gave the user the possibility to use a much better model called Gemini from Google, where the user can obviously see the differences between our simple model and a much more advanced one. The testing of the model can be seen on the follow link : `https://youtu.be/dx3BoAoKp2U`.

## 6. Conclusion

In this lab, we successfully developed a spoken question-answering chatbot that combines speech recognition, a large language model, and text-to-speech synthesis into a cohesive and interactive application. Our experiments showed that Qwen1.5-0.5B-Chat provided the most balanced performance in generating human-like, on-topic responses, while Kokoro offered superior audio quality for synthesized speech. Although BLEU scores and other metrics indicated limitations in quantitative evaluation, qualitative results affirmed the effectiveness of our chosen pipeline. Despite challenges in fine-tuning due to computational constraints, the final system achieved satisfactory performance for general-purpose conversational tasks. Future improvements could include specialized fine-tuning, enhanced prompt engineering, and the incorporation of real-time feedback mechanisms.

## 7. References

[1] D. Sulimov, "Prompt-efficient fine-tuning for gpt-like deep models to reduce hallucination and to improve reproducibility in scientific text generation using stochastic optimisation techniques," https://arxiv.org/html/2411.06445v1#Chx2.