# **Project Report: Building a Multilingual Speech Recognition Model for RAG Without Training**

1. **Introduction**

Speech is the fundamental human ability to express or communicate thoughts. It is the natural and efficient way to exchange information. Speech recognition is a software that translates spoken words into text. This technology enables us to use voice commands while working with electronic gadgets. Speech translation is the technology that allows people to communicate despite their native language. With progressive technology, these concepts are further expanded to build multilingual speech recognition models with artificial intelligence techniques which supports user interface with additional features, that can attend in the various fields of science and education. Further these models integrated with RAG can provide voice-activated systems that can provide more versatile and interactive user experience

* 1. **Objective**

A multilingual speech recognition model that processes the audio input given to it, recognizes the source language and translates it into selected target language. It utilizes deep learning techniques. Building of this model has several parts: speech recognition, source language verification, speech-to-text conversion, translation and conversion to target language. Integrating this model with a pre-trained RAG model available from Hugging Face’s Transformers library. The final model can be to generate response based on retrieved documents.

* 1. **Background**
* **Multilingual speech recognition**

Multilingual speech recognition with supervised learning has achieved great results and gained significant interest for their ability to transcript in more than one language. These models integrated with AI, developed by dominating tech companies such as Google and Microsoft, are being used for language translation.

* **Retrieval-Augmented Generation (RAG)**

RAG is an AI framework that fuses the strengths of conventional information retrieval system (databases) with the potential of generative large language models. When a query Is given to RAG, a RAG retriever searches a large database and ensure to pick most relevant documents based on the given query. This retrieved information is used to generate coherent responses for the input.

* **Integration of Speech Recognition and RAG**

Integrating RAG with multilingual Speech Recognition gives a robust system that can translate, transcribe and retrieve data in multiple languages. Many pretrained models for speech recognition and translation are also available, one such model could be whisper that has higher transcription accuracy. By combining these models, a pretrained multilingual speech recognition system can be generated.

1. **Methodology**
   1. **Model** **selection**

Whisper is an open AI automated speech recognition (ASR) system that is trained for 680,000 hours on multilingual supervised data collected from web. Being trained on such large and diverse dataset gives Whisper improved robust performance in speech processing. It enables transcription in multiple languages and translate from those languages to English as well.

* 1. **Audio Processing**

Whisper model is implemented as an encoder-decoder Transformer. The input audio is loaded using whisper model. This input audio is split into chunks of 30-seconds and is converted int a log-Mel spectrogram, that is ten passed into an encoder. A decoder is trained to predict the corresponding text caption, intermixed with special tokens that direct the single model to perform tasks such as language identification, phrase-level timestamps, multilingual speech transcription, and to-English speech translation.

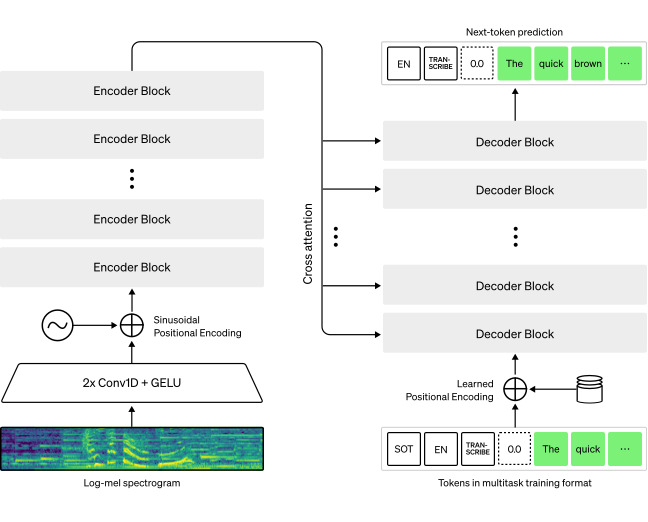


Fig 1. Speech Processing in Whisper

* 1. **Translation Tool**

Google Translator from ‘deep-translator’ library is used for translation as it is flexible and unlimited tool to translate between different languages in a simple way using multiple translators. This python tool is easy to use and support famous universal translators.

* 1. **Document retrieval**

A set of pre-defined dummy documents relevant to multilingual speech recognition queries are created. These documents a handled by manual mechanisms instead of advanced retriever libraries due to lack of resources and environment support

* 1. **Response Generation**

‘RagSequenceForGeneration’ model from the Hugging Face Transformers library is used.  It performs RAG-sequence specific marginalization in the forward pass. It encapsulates two core components: a question encoder and a generator. The question encoder is an autoencoding model, [DPRQuestionEncoder](https://huggingface.co/docs/transformers/v4.43.4/en/model_doc/dpr#transformers.DPRQuestionEncoder). This model inherits from [PreTrainedModel](https://huggingface.co/docs/transformers/v4.43.4/en/main_classes/model" \l "transformers.PreTrainedModel).

* **config** ([RagConfig](https://huggingface.co/docs/transformers/v4.43.4/en/model_doc/rag" \l "transformers.RagConfig)) — Model configuration class with all the parameters of the model. Initializing with a config file does not load the weights associated with the model, only the configuration. Check out the [from\_pretrained()](https://huggingface.co/docs/transformers/v4.43.4/en/main_classes/model" \l "transformers.PreTrainedModel.from_pretrained) method to load the model weights.
* **question\_encoder** ([PreTrainedModel](https://huggingface.co/docs/transformers/v4.43.4/en/main_classes/model" \l "transformers.PreTrainedModel)) — An encoder model compatible with the faiss index encapsulated by the retriever.
* **generator** ([PreTrainedModel](https://huggingface.co/docs/transformers/v4.43.4/en/main_classes/model" \l "transformers.PreTrainedModel)) — A seq2seq model used as the generator in the RAG architecture.
* **retriever** ([RagRetriever](https://huggingface.co/docs/transformers/v4.43.4/en/model_doc/rag" \l "transformers.RagRetriever)) — A retriever class encapsulating a faiss index queried to obtain context documents for current inputs.
  1. **Integration with User Interface**

Gradio, an open source python package, is used to build a demo web application to create user friendly interaction with the model.



Fig 2. Snapshot of Interface

1. **Evaluation**
   1. **Speech Recognition Accuracy**

Word Error Rate (WER) is a common metric of the performance of a speech recognition. The WER was 2% for the model.

* 1. **Translation Quality**

BLUE (BiLingual Evaluation Understudy) is a metric for automatically evaluating machine-translated text. It is a number between zero and one that measures the similarity of the machine-translated text. The BLEU score for the translation system was calculated to be 1.00. This indicates the generated translations perfectly match the translations in terms of n-gram overlap. However, this result may reflect a limited dataset size or exact matches rather than general translation quality.

1. **Conclusion**

In conclusion, the project successfully developed a multilingual speech recognition and translation system with good accuracy in both transcription and translation tasks. For future works, we can focus on expanding the datasets and adding relevant documents to further validate the model’s performance.

1. **References**

* [**https://deep-translator.readthedocs.io/en/latest/**](https://deep-translator.readthedocs.io/en/latest/)
* [**https://huggingface.co/docs/transformers/en/model\_doc/rag**](https://huggingface.co/docs/transformers/en/model_doc/rag)
* [**https://www.gradio.app/docs/gradio/interface**](https://www.gradio.app/docs/gradio/interface)
* [**https://platform.openai.com/docs/guides/speech-to-text**](https://platform.openai.com/docs/guides/speech-to-text)