Bryant McArthur
Wade McMillan
February 18, 2022
Dr. Lonsdale, NLP

Final Project Proposal

The basic problem that we intend to address is speech to speech translation. This means that we want to create a relatively fast pipeline for receiving speaker input and outputting it into another language. We intend to have several selectable options for both input languages and output languages. We have the minimum expectations of English, Spanish, and Polish because combined we are relatively fluent in these languages allowing for a level of human evaluation metrics on success, however we don't expect to stop there and will aspire to include several more languages.

The need for speech to speech translations spans to many practical applications across a variety of business and personal uses. With the world use of video calls being increasingly used and the economy being global already it is essential for businesses to be able to conduct business quickly, accurately and in a variety of languages. Having a speech to speech translation system that is multilingual would allow a business presentation to reach multiple markets, individuals, and companies simultaneously without the need for expensive translators.

Outside of business the need for speech to speech translation in an affordable and easy to use platform exists due to the amount of long distance communication that can occur. Technology has surpassed the distance barrier when it comes to communication, and in this project we intend to address this area of the need by attempting to make an easy to use interface for multilingual translation, that is both fast and has high accuracy.

We considered using a direct approach to translate the Audio input directly to the Audio output without any intermediary steps by training the model with massive amounts of speech in various languages. After careful consideration we thought it best to employ more intermediary steps in order to very slightly lower the accuracy of the translation yet significantly decrease our spatial and temporal complexities of the pipeline in order to run our code more smoothly and efficiently.

We intend to create our Speech to Speech translator by first using Automatic Speech Recognition (ASR), next a form of Machine Translation (MT), and finally a Text to Speech Synthesizer (TSS) all embedded in a simple interface. The speaker will talk into the microphone in any given language, the program will print the text in the source language and target language, and then proceed to speak the translation in the target language and optionally save an audio file to a given destination.

As mentioned in previous paragraphs we intend to use speech to text, machine translation engines (or api calls to such engines), and text to speech. We also intend to build this pipeline so that it attaches to a GUI interface for easy use. Our current design plan will allow for improvements to be added between the steps as we find opportunities to improve and refine our initial plan.

For example we might find the need to include a basic cleaning of filler words from the speech input, or use one of the toolkits we have used in class to see if the input can be parsed into valid sentences. If not we could output a warning to the user, and reprompt them to speak again more clearly.

We will code everything in Python 3 which will enable us to easily create an interface for the user to interact with. Python has a good tool we can use named PySimpleGUI that would look professional.

Second, we must employ an Automatic Speech Recognition system. There are many options for an ASR system in English, however our difficulty comes in finding a system that works well multilingually. Microsoft Translator uses their system Microsoft Azure for speech recognition, and for this reason Azure seems like a compatible option for us to be able to interpret speech in many different languages.

Third, we must write or utilize an effective machine translation system. Statistical MT (SMT) has historically been at the forefront of MT but in the last decade or so Neural MT (NMT) has surpassed SMT in accuracy. We could build a statistical machine translation system by finding the probabilities of possible translations trained on a huge set of aligned bilingual corpora for all the different languages we want to translate between. A more reasonable approach would be to call a Neural model already in place, such as Google or Microsoft Translate. Again, Azure has a viable resource that may meet our needs.

Fourth, taking the target language text back into speech is the inverse of ASR. For this reason, we can most likely use the same tool we end up using in part ii.

In order to implement Azure we will have to gain access to their database by subscribing and using a subscription key after installing “azure.cognitiveservices.speech” via pip.

During our MT phase it will be useful to import JSONs in order to cleanly represent the data and translation properties as a list of dictionaries. We will also need to import and utilize “requests” in python to access Microsoft’s server. It is possible and likely other imports and systems will be needed as we develop the project but this is a rough guess of already known needs.

We plan to leverage the microsoft machine translation resources in a powerful way for this project. Immediately we can identify that we will need a subscription to Azure and access to both a speech resource which allows us to receive and store spoken input easily. It also will with a little work allow us to turn text of different input languages to spoken output. It has a variety of tones and has a male/female voice option that we plan to make selectable as options so that people can be more represented by the voice they identify with as well as having higher potential of being received well by the audience as dialects can give it a much more personalized effect.

Additionally we will use a machine translation system that has been trained by Microsoft. This will ensure we have state of the art accuracy, fluency, speed, and variety of languages to optimize user experience.

We can evaluate the system at three different points, the source text, the translation text, and translation audio.

It is straightforward for the user to double check the printed text is what was spoken into the microphone, so that will trivially be checked at runtime.

We will use BLEU scores to evaluate the accuracy of the text for reference translations we already have. BLEU scores are not always 100% accurate but are widely used in the industry for their simplicity in calculation and still give a good general idea on how well the MT system performed.

Evaluating the translation audio seems to be the most difficult task. We may end up just evaluating manually how accurate and fluid the output speech was in the languages that we know, or finding bilingual volunteers to help evaluate for other languages. Another more complicated option would be to find audio files where we already have a good given translated speech and use that as a reference to see how close our output was. This would be a similar idea to how BLEU score works except with audio but training the computer to recognize a metric in evaluation will be difficult.

As discussed we have selected the group option for this project. As such we have designed a plan for collaboration as well as a healthy division of labor that will ensure fairness and productivity throughout the project. We have decided that both of us will acquire experience working with the speech resource.

Bryant will be primarily in charge of working on the speech-to-text, and the machine translators, Wade will have the responsibility of text-to-speech, and the GUI. Most of the work will be done in collaboration to increase speed, and ensure we both have a full understanding of all parts of the project. To maximize quality assurance tasks the roles will be switched when regarding the evaluation metrics for the different portions of the project. For example while Bryant is in charge of text-to-speech, Wade has the task of ensuring his implementation is well done, and implementing, and recording the evaluation methods. If our scores don't reach expectations for the different parts of this task we will use group time to resolve this issue.

We will plan equal hours of work for the week and make sure we discuss at least weekly our progress, issues, and potential ideas for improvement. Based on previous work experience we have done we believe that we will be able to have an effective and agreeable group that will produce impressive results.