Combining ASR and NMT into *speech2tëẍt*

March 15, 2022

520.638 project proposal written by Nick Hinke and Katie Brandegee

Team Members: Our team is comprised of two members: Nick Hinke and Katie Brandegee. We are both currently in our last semester of the 5 year Robotics BS/MSE program, and both studied Mechanical Engineering during our time as undergraduates at JHU. Between the two of us, we are involved in a wide variety of domains within the field of robotics, ranging from medical imaging to autonomous vehicles, all of which provide a myriad of opportunities for deep learning applications. As such, we found it quite challenging to narrow down the options and ultimately pick a project. Ultimately, we discovered that despite our combined lack of experience in the field, we both share an interest in Natural Language Processing (NLP), and in travelling the world! As such, we decided to pursue a project that will involve both of these interests.

Project Summary: Our plan for this project is to train a network that is able to take audio inputs, convert that to text in the input language, and then translate that text into another language. This entire process spans both automatic speech recognition (ASR) and natural machine translation (NMT). Though both tasks are relatively well-defined, our project will focus on bridging these processes together, while simultaneously trying to limit the error introduced when converting the text outputs from the ASR process into translated text from the NMT process.

Proposed Approach: As briefly described above, our proposed solution to the problem of translating spoken natural language involves a two-stage pipeline. Within the first stage, the pipeline must convert the spoken natural language into comprehensible text that can be processed by the model. This conversion task will be accomplished using what is known as "Automatic Speech Recognition" (ASR) [3]. Then, this text output can be fed as an input to the second stage of the pipeline, where the text will be translated to text in a different language. Similarly, this translation task will be achieved using what is known as "Neural Machine Translation" (NMT) [5], the same technique that is used within Google Translate [7]. The key benefit to this two-stage framework is the ability to design, train, and evaluate each stage of the pipeline individually before chaining them together to build an end-to-end model.

Our initial proposed architecture for the first stage involving ASR is comprised of several important steps. First, the audio samples must be converted into spectograms, which can be fed as input *images* to a relatively simple convolutional neural network (CNN) whose job is to extract robust features from the sample. The features extracted by the CNN can then be fed into some type of recurrent neural network (RNN) that is capable of capturing the temporal information available within the converted audio samples. Finally, the outputs from the RNN can then be fed into a few traditional fully-connected layers ending with a softmax layer to determine individual character probabilities for the duration of the audio sample. It should also be mentioned that connectionist temporal classification algorithm (CTC) will be used as a loss function during training and for decoding during testing [6]. This sequence of steps will allow the first stage to transform the natural language audio signal into text information that can be fed into the second stage.

Similar to the first stage, our initial proposed architecture for the second stage involving NMT consists of several vital steps. First, the output text information from the first stage must be transformed into viable numerical inputs by using a technique such as one-hot encoding. Then, the numerical inputs can be fed into an RNN-based encoder-decoder architecture, as is typically the choice for this task [4]. The RNNs used within the encoder-decoder model will likely be long short-term memory networks (LSTMs), as they are an excellent candidate for capturing sequential information within the textual data. This sequence of steps as outlined above will allow the second stage to translate the text information provided by the first stage into a different language.

By combining the two distinct stages of the pipeline as outlined above, we can achieve an end-to-end model capable of converting audio samples of natural language into text in a second *different* language. While both stages individually involve multiple steps consisting of networks of varying architectures, none of the sub-networks themselves are exceedingly complex. This should result in a reasonable–and more importantly, *achievable*–scope for the project, but will require a very considerable amount of time for hyperparameter tuning across the various networks and optimizing the interfaces between networks.

Potential Datasets: We plan on using the Common Voice dataset for the automatic speech recognition portion of this project, and parts of the Tatoeba dataset for the natural machine translation portion. The Common Voice dataset was created by Mozilla and includes audio files and the corresponding text labels in several different languages [1]. The Tatoeba dataset was created by Tatoeba and contains text in one language translated to another across many different

languages [2]. Both datasets have breadth in material and language choice, and will complement each other nicely.

Extended Scope: Depending on how difficult and manageable this project turns out to be, we have developed several other "stretch goals" to pursue upon completion of the project as outlined above. Our initial plan involves only translating from English audio to Spanish text; however, once we build the complete model, we may attempt to extend the input and output capabilities to a variety of other languages. If we do this, we also want to create an interface that allows for the user to choose the languages and makes it easy to input audio and receive translated text. Finally, if we find that the project complexity is still too low, we could add a third stage to the pipeline involving speech synthesis of the translated outputs. It is worth noting, however, that these are indeed *stretch* goals, as we anticipate an already sizeable load given the project described above.

Concerns: Though in theory taking an audio sentence input and converting it to text and then converting that text to text in a different language does not require supplemental datasets, it might prove difficult for the network to output translated sentences without further contextual information. This context may be related to how different languages structure their sentences, or colloquialisms that depend on the sentence context. Therefore, we might need more information in order to output readable sentences. Additionally, just as it is easy to extend the scope of this project as mentioned above, if it does prove to be too difficult for unforeseen reasons, we will be able to focus on one half of this project instead. In other words, if the stated project scope seems unreachable, we can narrow our focus to just one of the two stages within the pipeline. If this were to happen, we could investigate how different neural network architectures with differing hyperparameters vary the performance of the end-to-end model.

References

[1] Common voice. https://commonvoice.mozilla.org/en/datasets, 2022. Accessed: 2022-03-14.

[2] Tatoeba project: Spanish-english. http://www.manythings.org/anki/, 2022. Accessed: 2022-03-14.

[3] K. Doshi. Audio deep learning made simple: Automatic speech recognition (asr), how it works. https://towardsdatascience.com/audio-deep-learning-made-simple-automatic-speech-recognition-asr-how-it-works-716cfce4c706. Accessed: 2022-03-14.

[4] S. Kostadinov. Understanding encoder-decoder sequence to sequence model. https://towardsdatascience.com/understanding-encoder-decoder-sequence-to-sequence-model-679e04af4346. Accessed: 2022-03-14.

[5] Q. Lanners. Neural machine translation. https://towardsdatascience.com/neural-machine-translation-15ecf6b0b. Accessed: 2022-03-14.

[6] H. Scheidl. An intuitive explanation of connectionist temporal classification. https://towardsdatascience.com/intuitively-understanding-connectionist-temporal-classification-3797e43a86c. Accessed: 2022-03-14.

[7] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Łukasz Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016.