Daily Log

Tuesday October 15

(Anchor day) I spent today watching the first half of the 8th lecture video on machine translation, seq2seq, and attention. I learned about an earlier machine translation model – statistical machine translation. The idea is that you want to find the best English sentence x given a foreign (say, French) sentence y. The alignment is which English words correspond to which French words, and you use a heuristic search algorithm to search for the best sentence probability.

Now, you generally use neural machine translation (NMT) rather than a statistical model. Sequence to sequence (seq2seq) is a neural model that uses two RNNs. The encoder RNN produces the encoding of the source sentence. The decoder RNN generates the target sentence. The output of each step of the decoder is fed in as the next step's input.

Thursday October 17

Today, I finished the lecture on machine translation. I learned about the pitfalls of "greedy decoding". The greedy decoding model takes the most probable word at each step. The problem, however, is that you can't undo previous decisions once you have more information later. Beam search decoding keeps track of the k most probable partial translations, rather than just the highest scoring one. This increases your likelihood of finding the best sentence, but it's not a guarantee. Only an exhaustive search can guarantee finding the best sentence, but this is obviously inefficient and infeasible.

The lecture also discussed BLEU, which evaluates translation performance with a similarity score to human translation. It uses n-gram precision and penalizes translations that are too short. Finally, the lecture discussed **attention**, which is a solution to the bottleneck problem. In the bottleneck problem we have an issue of earlier information being lost in trying to encode all the sentence's information in the final hidden state. Attention fixes this by using a direct connection to the encoder in each step of the decoder network.

Monday October 21

I'm starting to work on the machine translation assignment. This will be the first task I have to run remotely, on a GPU machine. Last class, I'd accidentally messed up my PyCharm terminal in the process of trying to make it a conda terminal, but I managed to fix that today. I discovered that TJ has our own GPU server (zoidberg) that I can use for this project, which I'm going to explore next class.

Tuesday October 22

I began today's class by connecting to zoidberg from my machine, which went smoothly. Once I had ssh'ed into the server, I installed Miniconda. I haven't tried creating a conda environment

yet, but I'll do that once I start needing to run code on zoidberg. I was mostly reading through the notes on the translation model today and figuring out which methods I need to implement.

Thursday October 24

I downloaded Miniconda onto my own computer today, because I realized my Anaconda version was out of date. I researched how to get PyCharm terminal run Anaconda because that hadn't been working recently. I configured my terminal with my new Miniconda installation and was able to configure my virtual environment for my machine translation task. I started working on my model, but haven't been able to produce output yet because I got some errors that I'll need to debug next week.
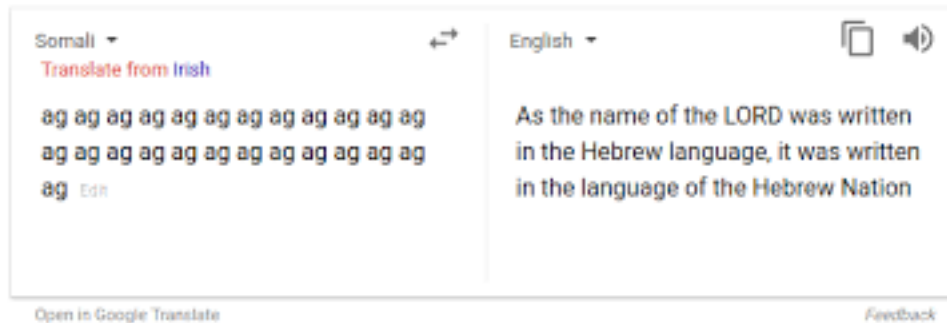
Timeline

Date	Goal	Met
Sunday Oct 13	Finish my dependency parsing neural network model. Work on week 4 materials with RNNs for sentence classification.	Finished dependency parsing and almost finished watching week 4 lectures.
Sunday Oct 27	Training the RNN model for translation and work on week 5 lecture videos. Start reading literature for my question answering project.	I've made progress on the RNN model, but I'm not quite close to being done with it yet. Now that I've configured my conda environment and found a GPU machine, I should be able to make progress on it next week. I think the goal I set for this week will take more time that I'd originally planned for it.
Sunday Nov 3	Work on training the RNN model for translation. Get the debugging cases to work (not necessarily the entire model yet.	
Sunday Nov 10	Finish training the RNN translation model. Start week 5 lecture videos and begin reading literature on my question answering project.	

Reflection

While exploring machine translation models in the lectures this past week, I gained some interesting information on how these models are trained and unpredictable ways in which they can go wrong. Before going into translation models, I learned about how RNNs can generate language in the style of its training data. For example, a model trained on the Harry Potter corpus can make up sentences that sound like the Harry Potter books, and one trained on recipes can generate some text that sounds like recipes. Even though neither of these two examples end up making logical sense, they do (mostly) make grammatical sense and sound similar to their source text. In a trans-

lation model, instead of the model just generating some text in a certain style, it uses the input sentence and feeds that into the decoder to generate the final sentence.

One example that was discussed in the class was translation between languages that do not have much data for parallel corpera (the same writings written in two different languages). In such cases, the Bible is often the only source of a parallel corpus. In the example below, journalists typed a nonsense syllable into Google Translate in Somali. Because this nonsense syllable did not convey any useful information, the decoder network just began to generate text in the same way it would had there been no input at all – it began to generate text that vaguely sounded like its input data, which in this case, was the translated Bible.

Somali ▾	⇄	English ▾	🗗 ◀)
Translate from Irish			
ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag Edit		As the name of the LORD was written in the Hebrew language, it was written in the language of the Hebrew Nation	
Open in Google Translate			Feedback

From this example, I learned how these models can behave in unexpected ways and it's often very difficult to catch until you stumble upon them by mistake. In particular, I learned the importance of training a model on a large and diverse corpus of text whenever possible to avoid situations like this one.

As far as my project progress goes, I definitely made progress in these past two weeks (one short and one normal week), but I didn't originally give myself enough time for the translation model project because I didn't realize how much would be involved in it. I've now given myself some more time to work on it, and now that my conda environment is properly configured I should be able to make more progress on it.