My Research Passion

Everyone should have access to speech technology in their native language.

This is the conviction that drives my research. I want the world to be a place where someone who's born blind doesn't have to settle for a lesser education because of lack of audiobooks. I've worked with people in that exact situation to build a Kyrgyz speech synthesizer, because they want more opportunities for other blind Kyrgyz speakers. I've played a small role to make speech synthesis available for free, but there is a hard limit as to what I can accomplish on my own.

The Google AI residency is the next step to accomplish my goals. I want to publish influential work and get more people caring about low-resource languages. With my knowledge of linguistics and passion for machine learning, the Google AI team are the perfect place for me to flourish and share my research with others.

Collaborating with researchers at Google, I will help create new approaches and algorithms, testing theories, and diving deeper into multi-task learning. In particular, I have many ideas for multilingual experiments, extracting information from large datasets to transfer knowledge to smaller domains. The competitive (yet collaborative) research atmosphere at Google excites me, and I will bring my enthusiasm and dedication with me.

Overcoming Obstacles

Working with low-resource languages, my thesis topic is Multi-Task Learning for deep neural net acoustic modeling in Automatic Speech Recognition (ASR).

Acoustic models in ASR are statistical classifiers, accepting as input some speech signal (ie. time chunks of an audio file), and returning probabilities over phonetic transcriptions. Using Multi-Task Learning, my research accomplishes this audio $\rightarrow$ phonetic transcription with a feedforward neural net with multiple output layers. Each output layer represents a different task the net is required to perform. If two tasks are related, they will contain overlapping information. Learning related tasks in parallel should improve performance on any one of the tasks, because the weights in the net will be biased towards general, task-independent representations of the data.

The Multi-Task approach offers an elegant way to exploit small datasets, if you can come up with the right tasks. My early experiments show that by adding tasks along dimensions relevant to linguistics (place, voicing, manner), we can lower Word Error Rates for smaller data sets. I spent hours carefully defining each task, consulting both the machine learning and linguistic literature. Via trial-and-error, I eventually found tasks that showed improvement over my baseline. However, defining each task by hand is not a scalable approach. Since then, I've devoted my research to finding scalable, automatic solutions to task creation.

You can view Multi-Task classifier as a kind of ensemble model, and once you do, you open up a flood of inspirations from past work. Looking for automated ensemble model approaches, I came across the Random Forest.

The Random Forest is trained not by combining different architectures or different labels, but by merely resampling the data. Each subsample has its own decision plane, specific to the data and noise in that particular sample. By averaging the votes of all trees, the noise is ignored, but the generalities remain. Taking inspiration from Random Forests, my current research investigates Multi-Task neural nets, where the different tasks are independent subsets of the data. I use the Kaldi Speech Recognition toolkit to run experiments.

Recently I ran into an obstacle running my experiments. Typically, I perform my experiments in this way: After I've defined a new training procedure, I run a toy model on a small subset of the data, to quickly make sure there's no bug in the code. Then, I run some pratice models with fewer parameters and epochs on a larger subset of the data, to get a ball-park idea of model performance. Finally, I define an architecture with a real number of parameters and epochs on all the data, and set a loop to train multiple versions of the same model (to get a standard deviation for the differences in random initializations).

My most recent obstacle came during the "practice round" stage. I was testing different sized random re-samples of the data as auxiliary tasks, and the results suggested that the size of the re-sample made no difference at all. I tried re-samples all the way from 50% to 5%, but the performance was identical. Not only was the performance identical regardless of the size of the re-sample, but the baseline did just as well as all the other models. This was strange to me because in my expert-crafted tasks, I never got identical performance to the baseline.

To get a visual intuition, I plotted the performance of the different tasks over time, and realized a major issue. The smaller tasks were training for less time, and as such, the main task was allowed to overfit the data after all the smaller tasks ran out. I was getting sequential learning instead of parallel learning.

I went to the `kaldi-help` Google group to see if I could get some help there. I posted my problem on the forum, and Dan Povey replied with some pointers. In a follow-up email he suggested that there might be a bug in my scripts resulting in a error for the vizualizations. I double checked my scripts, and the original data I used to plot, and there was no bug. He said in that case, it might be as simple as files being overwritten without any error messages to notify.

As it stands, he suggested I clear out all my generated experiment files and retrain from scratch, to make sure there's no file overwrite happening. I wiped my log files and the experiment is running on an Amazon instance as I write this.

This issue does not have a clear solution yet. I decided to share with you because it is representative of the main issues I face during my research. The open-endedness of research does not phase me. I know my goal (make speech technology for all languages), and I know it is so big that I can happily spend a career chipping away at it, and Google AI is the perfect place to start chipping.

Thank you for your time and consideration.