As a bit of a changelog of events, here has been my progression:

Tues night/Wed morning:

Created python file to generate sentences based on linguistics knowledge. Made corpus file of various length (testing with N=500)

As of Thurs 3pm, my best WER is **16.49%.**

Thurs afternoon:

From inspecting output and error, it appears that the largest issue is in classifying the word “a” and the word “wanna.” After listening to the audio recordings, I selectively chose which alternate pronunciations proved useful in classifying missed or substituted words and added them into the language model generation pearl script. Notably, adding TO(2) T AH accounted for reducing the in-sample error by another 1%. (**15.55% error**)

I then added the words “wanna” and “gimme” into sentences at random by modifying the python corpus generator script that I made. I think this will account for some natural words that might be in the test data. Just adding those into the corpus brought the error down to **13.67%**.

But again, from looking at the output, the word “a” is consistently being misclassified or left out altogether. The audio recordings make it seem as though it can be heard, so I’m going to play around with alternate pronunciations. In the corpus.dic, it is being mapped to the sound AH, which I would expect to work for most cases. However, for the cases where people stress the “a” as “AYYY”, I am adding in a mapping from A(2) to EY. Error 13.45%... this is fun. That did not really help, I’ll come back to this issue in a second.

There are sentences in the form I want … instead of I want to order. I am adding this variation and some others in my python script to generate more varied corpus sentences. **12.36% error**.

Hey… “Hot peppers” might be useful to add into the sentences. **11.28% Error-in**

Alright the biggest issue now looks like identifying some “a”s. Also I played around with changing the custom pronunciations, sometimes not having the T AH or the A → EY works better… hmm. First I will focus on the AN, there are many A’s that are falsely substituted for AN’s. I have thought about it for a while and don’t know how better to improve this aside from reducing its frequency in the corpus. I will heavily weight against using “an extra” as it is much less common. **10.20% error**

My goal is to get sub 10 in-sample error! Let me do a little bit of a risk here which might be harmful for other applications, but since the word a commonly comes after order (and these are some of the cases where it’s not being

picked up, let me try to add a mapping for A(3) → R AH. **9.76% error** we did it! Although I can't be sure that will help out of sample error all too much unless the out of sample testing is structured very similarly to this… I'll say it's a calculated risk because it mentioned this structure in the instructions.

I also found the language model has difficulty specifically with lin_012.wav, with an abnormal 55.55% error on that sample. Since it's such an outlier, I will disregard under the assumption that it is not the norm.

Last thing to test is a slightly better look at corpus size. Now that I have what I find works well, trying with sizes 300, 500, 800:

300 → 10.41% WER
500 → 9.67% WER
800 → 9.54% WER

Hmmm At this point, I think I must be overfitting the data, and this is obviously true for other sentences that don't pertain to pizza deliveries… however, since the pizza order is supposed to follow a specific structure, maybe the overfitting will be okay.

800 → 9.76%
1000 → 9.98%
1500 → 9.33%

There is certainly a lot of variability at play due to the randomness of the sentences. If I had more time, I would likely look further into changing the weights for the sentences. I'm not going to push the corpus size any further since the instructions even warned that the script is not built for large file sizes (multiple thousand), so I will run a few more tests at 1500 and then submit.

If I have time tonight, I'd like to make a quick bash script to run this procedure many times to get the best output and then use that as the best ASR model for the data. Unfortunately, though, I don't think I have the time.