HW2 Part2: Speech recognition

Kinjal Jain

November 8th, 2019

1. Acoustic Model:
   I experimented with multiple models defined as part of the train_acoustic_model.sh file. The best results on the training data were obtained on

2. Dictionary:
   CMU dict was used for the purpose of serving as the dictionary.

3. Language model:
   This was generated using SRILM ngram-count tool. I experimented with *interpolate1*, *interpolate2* , *interpolate3* along with order 3. The best result was obtained for generic interpolate which defaults to the order passed (3 in this case). The grammar was created using the sentences generated using the different combinations of toppings, size of pizza, etc. For generation of multiple sentences, I used the “powerset” utility provided by the “itertools” package in python.
   The sentence generation file can be found at:
   https://drive.google.com/open?id=1dl3wX2KH9nntJ2351dyelSJU7pNQweTe

4. Hypothesis and lattice files:
   These can be located under the s5 folder. For the best results, check under the individual model folder.

5. Initial error rates achieved for different acoustic models:
   a. Mono: 13.05%
   b. Tri1: 12.28%
   c. Tri2a: 12.67%
   d. Tri2b: 11.90%
   e. Tri2b_mmi:
      i. Iteration 3- 11.71%
      ii. Iteration 4- 11.71%
   f. Tri2b_mmi_b0.05:
      i. Iteration 3- 11.71%
      ii. Iteration 4- 11.71%
   g. Tri2b_mpe:
      i. Iteration 3- 13.05%
      ii. Iteration 4- 13.05%
   h. Tri3b: 12.28%
   i. Tri3b_fmmi_b:
      i. Iteration 3- 11.52%
      ii. Iteration 4- 11.52%

iii. Iteration 5- 11.71%
iv. Iteration 6- 11.52%
v. Iteration 7- 11.71%
vi. Iteration 8- 11.52%

j. Tri3b_fmmi_c:

i. Iteration 3- 11.52%
ii. Iteration 4- 11.52%
iii. Iteration 6- 11.71%

The rest three iterations failed because of system memory issues. The experiment was tried thrice, but every time for 2 or 3 iterations it failed.

k. Tri3b_fmmi_d:

i. Iteration 3- 11.90%
ii. Iteration 4- **11.25**%
iii. Iteration 5- 11.25%
iv. Iteration 6- 11.71%
v. Iteration 7- 11.25%
vi. Iteration 8- 11.25%

l. Tri3b_mmi:

i. Decode 1- 11.71%
ii. Decode 2- **11.71%**

6. Few errors observed:

a. Because we were adding audio file name labels, and they were not present in our lexicon and dict used, it was inserting nothing or the most common word all the time which was unnecessarily increasing the word error rate.

b. Some words like ***HAM, EXTRA, HOT*** were mislabelled mainly due to their lesser count value in the training text. The language model was not assigning them a good probability.

7. Solutions for the above errors observed:

a. For extra filename labels, I prune the last word of every file in the test dataset, which is nothing but the train text file itself for us. This is done as part of the following code:

```
awk -F' ' '{$NF=""}1' data/text > data/text1
cat data/text1 > data/text
rm data/text1
```

It should be noted that when testing is done with unseen data this label cutting should be done before the scoring is done.

b. Adding more sentences which contain these less frequent words helped in mitigating the Word errors observed due to them. This is evident by the updated

WER after the language model was built with a bigger and much more exhaustive corpus. The corpus used to build the language model is located in the submission as **s5/data/local/tmp/final_corpus.txt**

8. Final WER achieved after building new language model:
    a. Mono: 7.16%
    b. Tri1: 3.25%
    c. Tri2a: 2.6%
    d. Tri2b: 1.74%
    e. Tri2b_mmi:
        i. Iteration 3- **1.08%**
        ii. Iteration 4- 0.65%
    f. Tri2b_mmi_b0.05:
        i. Iteration 3- **1.08%**
        ii. Iteration 4- 0.65%
    g. Tri2b_mpe:
        i. Iteration 3- **1.3%**
        ii. Iteration 4- 1.3%
    h. Tri3b: **1.52%**
    i. Tri3b_fmmi_b:
        i. Iteration 3- 0.43%
        ii. Iteration 4- 0.65%
        iii. Iteration 5- **0.00%** (mostly because we have exhaustively entered sentences with given lexicons for training), and so it seems to be overfitting
        iv. Iteration 6- 0.87%
        v. Iteration 7- 0.87%
        vi. Iteration 8- 2.0%
    j. Tri3b_fmmi_c:
        i. Iteration 3- 0.43%
        ii. Iteration 4- **0.22%**
        iii. Iteration 7- 0.43%

        The rest three iterations failed because of system memory issues. The experiment was tried thrice, but every time for 2 or 3 iterations it failed.

    k. Tri3b_fmmi_d:
        i. Iteration 3- 0.00%
        ii. Iteration 4- 0.00%
        iii. Iteration 5- **0.22%**
        iv. Iteration 6- 0.00%
        v. Iteration 7- 0.00%
        vi. Iteration 8- 0.00%
    l. Tri3b_mmi:

i. Decode 1- 1.3%
ii. Decode 2- **1.08%**

Please note that some iterations weren’t done because of my system issues, as the processes weren’t getting forked because of memory issues. But, **I believe a good low WER was achieved Tri2b_mmi itself (1.08%) and so for test data also, the same model can be considered to give a low WER, and it is not required to train more complex models.**

The zip file for this submission is located at:
https://drive.google.com/file/d/1iRUgR_mcmNpXOOuxXAqg08TtGAGFmcYO/view?usp=sharing