Report for HW1

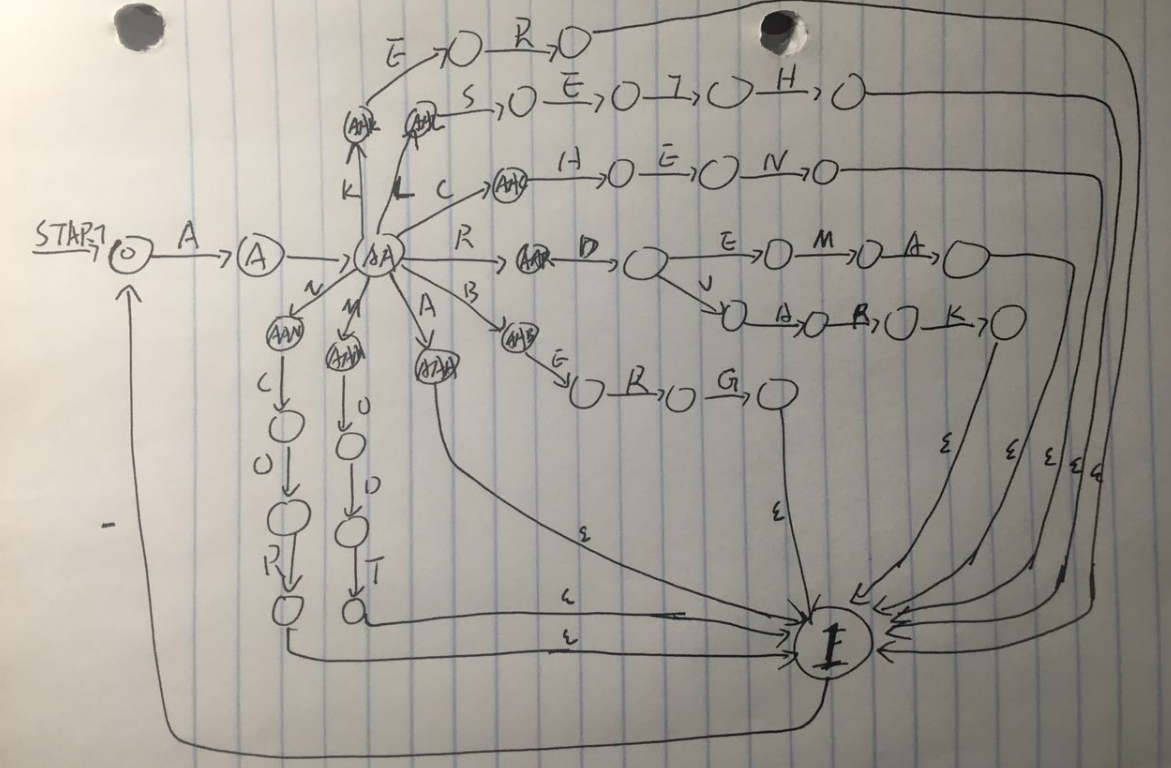
CS 519 Natural Language Processing

Prof. Liang Huang

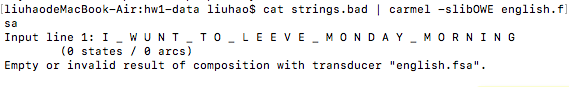
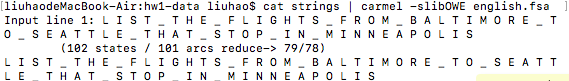
Group member: Gunea.Aditya, Hao.Liu, Ke.Huang

**Part1:**

1. The small FSA for vocab.small just like this:

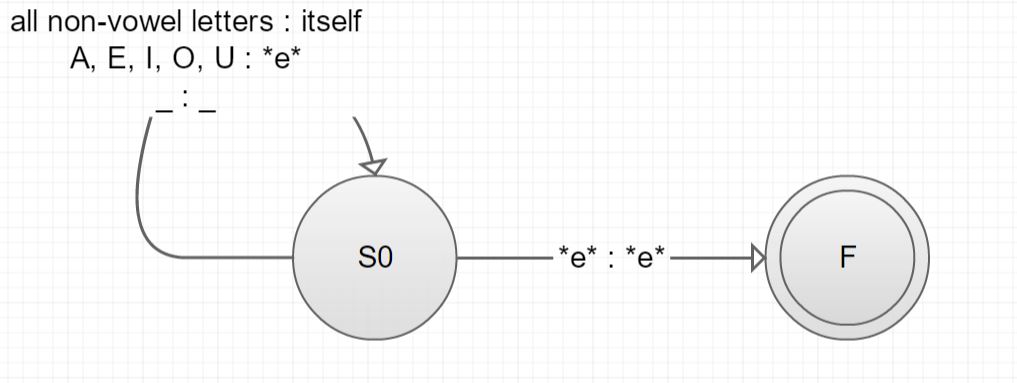


2. There are 256331 states and 361732 transitions in this FSA..

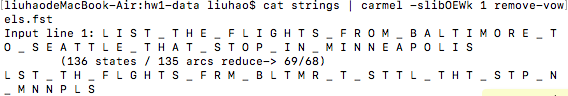


So we do accept strings in **strings**, and reject every string in strings.bad.

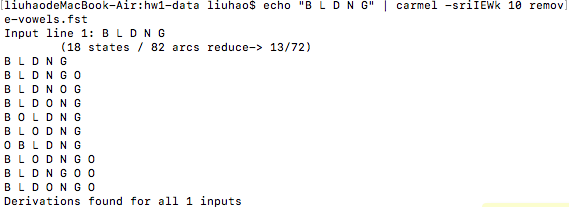
**Part2**





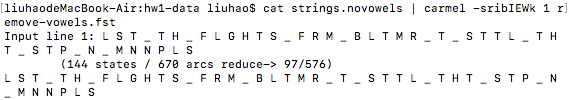






This is a result of applying some weights.





1. 

Accuracy is like 1.28%

1. Because the machine does not know where to restore vowels, and in this case it actually restore nothing.

**Part3:**

1. We could combine the FSA and FST by stick the FSA on the input side of FST. Using command like: cat strings.novowels | carmel -sribIEWk 1 english.fsa remove-vowels.fst
2. The new accuracy is 33%, its achieve by filter our input with FSA to make sure the output words is exist, even most of them still different with its original file.
3. This is a great improvement, but still not good enough. Probably the reason is because this model doesn’t know the frequency of the word and it also can’t understand the context of the word. So it will create some ridiculous sentence some times.
5. Here we will use **strings** to build a wFSA which will count the probability of each transition. This is much more like how our make.py working. We will go though and create each state at the same time we will create a counter for each state to record how many times this state has been reached. And there is also a big counter outside the loop to count how many state we have achieved during the building progress. And then the small counter divide by the big counter will give the probability of each transaction