REPORT - NLP (CSCE-689, Programming Assignment #3 Viterbi)
Name - Navneet Gupta
UIN - 226000691

1. Compile and Execution

It is developed with python 2.7 . The original training data and results are in the zip itself, along with the source code.

Steps -

1. Unzip the file.
2. cd into that folder
3. python viterbi.py probs.txt sents.txt (Use python 2.7) - (first file specifies probabilities and second file has test data)
4. Output will get printed on the terminal
5. It will read the test data line by line and for each line will dump the output.

2. Results and Analysis

Results -

PROCESSING SENTENCE: mark has fish

FINAL VITERBI NETWORK
P(mark=noun) = 0.0720000000
P(mark=verb) = 0.0060000000
P(mark=inf) = 0.0000000100
P(mark=prep) = 0.0000000100
P(has=noun) = 0.0000004620
P(has=verb) = 0.0014040000
P(has=inf) = 0.0000001320
P(has=prep) = 0.0000021600
P(fish=noun) = 0.0000864864
P(fish=verb) = 0.0000000210
P(fish=inf) = 0.0000000309
P(fish=prep) = 0.0000000351

FINAL BACKPTR NETWORK
Backptr(fish=noun) = verb
Backptr(fish=verb) = noun
Backptr(fish=inf) = verb
Backptr(fish=prep) = verb
Backptr(has=noun) = verb
Backptr(has=verb) = noun
Backptr(has=inf) = verb
Backptr(has=prep) = noun

BEST TAG SEQUENCE HAS PROBABILITY = 0.0000432432
fish -> noun
has -> verb
mark -> noun

FORWARD ALGORITHM RESULTS
P(mark=noun) = 0.0720000000
P(mark=verb) = 0.0060000000
P(mark=inf) = 0.0000000100
P(mark=prep) = 0.0000000100
P(has=noun) = 0.0000004627
P(has=verb) = 0.0014040182
P(has=inf) = 0.0000001327
P(has=prep) = 0.0000023100
P(fish=noun) = 0.0000866446
P(fish=verb) = 0.0000000379
P(fish=inf) = 0.0000000309
P(fish=prep) = 0.0000000351

PROCESSING SENTENCE: mark bears fish

FINAL VITERBI NETWORK
P(mark=noun) = 0.0720000000
P(mark=verb) = 0.0060000000
P(mark=inf) = 0.0000000100
P(mark=prep) = 0.0000000100
P(bears=noun) = 0.0000924000
P(bears=verb) = 0.0009360000
P(bears=inf) = 0.0000001320
P(bears=prep) = 0.0000021600
P(fish=noun) = 0.0000576576
P(fish=verb) = 0.0000042042
P(fish=inf) = 0.0000000206
P(fish=prep) = 0.0000000234

FINAL BACKPTR NETWORK
Backptr(fish=noun) = verb
Backptr(fish=verb) = noun
Backptr(fish=inf) = verb
Backptr(fish=prep) = verb
Backptr(bears=noun) = verb
Backptr(bears=verb) = noun

Backptr(bears=inf) = verb
Backptr(bears=prep) = noun

BEST TAG SEQUENCE HAS PROBABILITY = 0.0000288288
fish -> noun
bears -> verb
mark -> noun

FORWARD ALGORITHM RESULTS
P(mark=noun) = 0.0720000000
P(mark=verb) = 0.0060000000
P(mark=inf) = 0.0000000100
P(mark=prep) = 0.0000000100
P(bears=noun) = 0.0000925442
P(bears=verb) = 0.0009360122
P(bears=inf) = 0.0000001327
P(bears=prep) = 0.0000023100
P(fish=noun) = 0.0000578162
P(fish=verb) = 0.0000042243
P(fish=inf) = 0.0000000206
P(fish=prep) = 0.0000000262

PROCESSING SENTENCE: mark likes to fish for fish

FINAL VITERBI NETWORK
P(mark=noun) = 0.0720000000
P(mark=verb) = 0.0060000000
P(mark=inf) = 0.0000000100
P(mark=prep) = 0.0000000100
P(likes=noun) = 0.0000004620
P(likes=verb) = 0.0000046800
P(likes=inf) = 0.0000001320
P(likes=prep) = 0.0000021600
P(to=noun) = 0.0000000004
P(to=verb) = 0.0000000000
P(to=inf) = 0.0000010193
P(to=prep) = 0.0000003861
P(fish=noun) = 0.0000000263
P(fish=verb) = 0.0000000535
P(fish=inf) = 0.0000000000
P(fish=prep) = 0.0000000000
P(for=noun) = 0.0000000000
P(for=verb) = 0.0000000000

P(for=inf) = 0.0000000000
P(for=prep) = 0.0000000031
P(fish=noun) = 0.0000000002
P(fish=verb) = 0.0000000000
P(fish=inf) = 0.0000000000
P(fish=prep) = 0.0000000000

FINAL BACKPTR NETWORK
Backptr(fish=noun) = prep
Backptr(fish=verb) = noun
Backptr(fish=inf) = verb
Backptr(fish=prep) = noun
Backptr(for=noun) = verb
Backptr(for=verb) = noun
Backptr(for=inf) = verb
Backptr(for=prep) = verb
Backptr(fish=noun) = prep
Backptr(fish=verb) = inf
Backptr(fish=inf) = inf
Backptr(fish=prep) = noun
Backptr(to=noun) = verb
Backptr(to=verb) = noun
Backptr(to=inf) = verb
Backptr(to=prep) = verb
Backptr(likes=noun) = verb
Backptr(likes=verb) = noun
Backptr(likes=inf) = verb
Backptr(likes=prep) = noun

BEST TAG SEQUENCE HAS PROBABILITY = 0.0000000001
fish -> noun
for -> prep
fish -> verb
to -> inf
likes -> verb
mark -> noun

FORWARD ALGORITHM RESULTS
P(mark=noun) = 0.0720000000
P(mark=verb) = 0.0060000000
P(mark=inf) = 0.0000000100
P(mark=prep) = 0.0000000100
P(likes=noun) = 0.0000004627

P(likes=verb) = 0.0000046801
P(likes=inf) = 0.0000001327
P(likes=prep) = 0.0000023100
P(to=noun) = 0.0000000006
P(to=verb) = 0.0000000000
P(to=inf) = 0.0000010196
P(to=prep) = 0.0000004320
P(fish=noun) = 0.0000000294
P(fish=verb) = 0.0000000536
P(fish=inf) = 0.0000000000
P(fish=prep) = 0.0000000000
P(for=noun) = 0.0000000000
P(for=verb) = 0.0000000000
P(for=inf) = 0.0000000000
P(for=prep) = 0.0000000051
P(fish=noun) = 0.0000000003
P(fish=verb) = 0.0000000000
P(fish=inf) = 0.0000000000
P(fish=prep) = 0.0000000000

Analysis -
For given three sentences, the tags seems to be correct semantically too.
Time complexity - O(n|S|^2), where n is the number of words and |S| is the max number of tags for a word.
Space Complexity - O(n|S|)

3. Problems and Limitations -

a. This program expects given training data in the form of probabilities. It cannot handle raw training data(annotated data).
b. In case of longer sentences, probabilities will become too small. Here, we show answer up to 10 digits of precision. This can cause problems in case of underfitting. So, ideally we should take logarithms of the calculations.
c. Here, the program works on the Bigram model. Bigram model may not work for some of the cases. In those cases, we should try Trigram or ngram models.
d. No smoothing has been applied. So, unseen probabilities have emission probabilities of 0.