

Training:

- In this first phase of training, we used data from the `simple-train-tags` and `simple-train-sentences` files to train our program. To do so we created a ***transitionsMap*** - stores the transitions and their probabilities, and ***observationsMap*** - stores the observations data that's words with their tags and their log probabilities based on frequencies. Here is the output of the transitions and observations map.

```
Run: POS
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA.app/Contents/lib/idea_rt.jar=54672:/Applications/IntelliJ ID
Transition map: {P={DET=-0.2876820724517809, N=-1.3862943611198906}, DET={N=0.0}, #={DET=-1.6094379124341003, PRO=-0.35667494393873245, N=-2.3025850929940455}, MOD={V=0.0}, '
Observation map: {P={in=-0.2876820724517809, for=-1.3862943611198906}, DET={the=-0.40546510810816444, a=-1.5040773967762742, this=-2.1972245773362196}, MOD={should=0.0}, V={
```

- After training our program we used the actual **simple test files** with the viterbi algorithm. For the viterbi algorithm, we used the pseudocode from the course website and backtracked the maps in the arraylist of maps to get and compared the returned list of tags to the ones read from the tags file.
- Based on this, our calculated the percentage accuracy of the program is approximately 86%

```
Run: POS
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA.app/Contents/lib/idea_rt.jar=54672:/Applications/IntelliJ ID
Transition map: {P={DET=-0.2876820724517809, N=-1.3862943611198906}, DET={N=0.0}, #={DET=-1.6094379124341003, PRO=-0.35667494393873245, N=-2.3025850929940455}, MOD={V=0.0}, '
Observation map: {P={in=-0.2876820724517809, for=-1.3862943611198906}, DET={the=-0.40546510810816444, a=-1.5040773967762742, this=-2.1972245773362196}, MOD={should=0.0}, V={
TEST FILE: Simple
32 words are correctly tagged, but 5 are incorrectly match using the viterbi algorithm
Accuracy: 86.48648648648648
```

- To test the brown files, we trained our program using the **brown-train-tags** and **brown-train-sentences** files and tested with the actual files **brown-simple-tags** and **brown-simple-sentences**. Using the viterbi algorithm, we found the percentage accuracy to be approximately 95.7%.
- Next, we hard coded the training data using PD7 training data to test if our program is working well. The results are as follows:

3 Training

Listing 1: Training Data

```
1 cat/N chase/V dog/N
2 cat/N watch/V chase/NP
3 chase/NP get/V watch/N
4 chase/NP watch/V dog/N and/CNJ cat/N
5 dog/N watch/V cat/N watch/V dog/N
6 cat/N watch/V watch/N and/CNJ chase/NP
7 dog/N watch/V and/CNJ chase/V chase/NP
```

3.1 Emission map

tag \ word	and	cat	chase	dog	get	watch	normalize by
CNJ	3	0	0	0	0	0	3
N	0	5	0	5	0	2	12
NP	0	0	5	0	0	0	5
V	0	0	2	0	1	6	9

3.2 Transmission map

From \ To	CNJ	N	NP	V	normalize by
#	0	5	2	0	7
CNJ	0	1	1	1	3
N	2	0	0	6	8
NP	0	0	0	2	2
V	1	6	2	0	9

```
Run: POS
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA.app/Contents/lib/idea_rt.jar=54777:/Applications/IntelliJ ID
Transition map: {P={DET=-0.2876820724517809, N=-1.3862943611198906}, DET={N=0.0}, #={DET=-1.6094379124341003, PRO=-0.35667494393873245, N=-2.3025850929940455}, MOD={V=0.0}, '
Observation map: {P={in=-0.2876820724517809, for=-1.3862943611198906}, DET={the=-0.40546510810816444, a=-1.5040773967762742, this=-2.1972245773362196}, MOD={should=0.0}, V={
TEST FILE: Simple
32 words are correctly tagged, but 5 are incorrectly match using the viterbi algorithm
Accuracy: 86.48648648648648

TEST FILE: Brown
34837 words are correctly tagged, but 1557 are incorrectly match using the viterbi algorithm
Accuracy: 95.72182227839754

Hard coded samples using the P07 training data:
Tags for 'dog chase cat': [N, V, N]
Tags for 'cat watch chase': [N, V, NP]
Tags for 'chase get watch': [NP, V, N]
Tags for 'chase watch dog and cat': [NP, V, N, CNJ, N]
Tags for 'dog watch cat watch dog': [N, V, N, V, N]
Tags for 'cat watch watch and chase': [N, V, N, CNJ, NP]
Tags for 'dog watch and chase chase': [N, V, CNJ, NP, V]

Console Test
Write a sentence or enter 'q' to quit:
|
```

- Lastly, we did a console test using the train data from the **simple-train-tags** and **simple-train-sentences** that we found to be at least 86% accurate.

```
Run: POS
Tags for 'cat watch watch and chase': [N, V, N, CNJ, NP]
Tags for 'dog watch and chase chase': [N, V, CNJ, NP, V]

Console Test
Write a sentence or enter 'q' to quit:
My name is Abdibaset
[PRO, N, V, DET]
I am a student at Dartmouth
[PRO, V, DET, N, V, DET]
What is the best test case for this problem?
[PRO, V, DET, N, P, N, P, DET, N]
How are you doing today?
[PRO, V, PRO, N, .]
Is this assignment overtly so boring and hard at the same time?
[PRO, V, DET, N, V, P, DET, N, V, DET, N, .]
When are traveling back home?
[PRO, V, P, DET, N]
How do you feel today friend?
[PRO, V, PRO, V, DET, N]
Do you play soccer? That's surprising!
[PRO, V, P, DET, N, .]
q
[PRO]

Process finished with exit code 0
```

- We also did console testing using the trained data from **brown-train-tags** and **brown-training-sentence** and here is the output for the same sentences. The training with **brown train data** gives more accurate tagging because it is a large data set.

```
Run: POS
Tags for 'cat watch chase': [N, V, NP]
Tags for 'chase get watch': [NP, V, N]
Tags for 'chase watch dog and cat': [NP, V, N, CNJ, N]
Tags for 'dog watch cat watch dog': [N, V, N, V, N]
Tags for 'cat watch watch and chase': [N, V, N, CNJ, NP]
Tags for 'dog watch and chase chase': [N, V, CNJ, NP, V]

Console testing with Brown tags training
My name is Abdibaset
[PRO, N, V, DET]
I am a student at Dartmouth
[PRO, V, DET, N, P, NP]
What is the best test case for this problem?
[WH, V, DET, ADJ, N, N, P, DET, N]
How are you doing today?
[WH, V, PRO, VG, N]
Is this assignment overtly so boring and hard at the same time?
[V, DET, N, ADV, CNJ, VG, CNJ, ADV, P, DET, DET, N]
When are traveling back home?
[WH, V, VG, ADV, P]
How do you feel today friend?
[WH, V, PRO, V, N, P]
Do you play soccer? That's surprising!
[V, PRO, V, P, DET, N]
q
[NP]

Process finished with exit code 0
```