Team SPGMA

Lee Ying Yang A0170208N - Naive 1 & Viterbi 1
Tang Yong Ler A0199746E - Naive 2
Alvin Tan Jia Liang A0203011L - Viterbi 2
Lin Da A0201588A - Viterbi 2

Question 2a)

We have tested the various smoothing values, namely (0.01, 0.1, 1, 10) on the naive prediction function to determine which smoothing value gives us the best accuracy results for the prediction on the test set. The best smoothing value that we have selected is 0.1.

Question 2c)

The accuracy of our predictions is: 65.31%(4s.f).
Out of 1378 tag predictions, 900 were correctly predicted.

```
Naive prediction accuracy:     900/1378 = 0.6531204644412192
```

Question 3a)

Given $j^* = argmax_i P(Y = j \mid X = w)$

We can evaluate the RHS of equation by applying Bayes' Rule:

$$P(Y = j \mid X = w) = \frac{P(X = w \mid Y = j)\, P(Y = j)}{P(X = w)}$$

Thus,

$$j^* = argmax_j \frac{P(X = w \mid Y = j)\, P(Y = j)}{P(X = w)}$$

Calculations of the variable probabilities:

$P(X = w \mid Y = j)$ can be reused from output probabilities generated from Naive 1.

Since we have access to information from the training data:

- $P(Y = j)$ can be computed with smoothing value of $\sigma_y = 0.1$

$$P(Y = j) = \frac{count(Y = j) + \sigma y}{\sum_k count(Y = k) + (numStates + 1) * \sigma_y}$$

- $P(X = w)$ can be computed with smoothing value of $\sigma_x = 0.1$

$$P(X = w) = \frac{count(X = w) + \sigma x}{\sum_k count(Y = k) + (numWords + 1) * \sigma_x}$$

Question 3c)

The accuracy of our predictions is: **69.30%(4s.f)**.
Out of 1378 tag predictions, 955 were correctly predicted.

```
Naive prediction2 accuracy:    955/1378 = 0.693033381712627
```

Question 4c)

The accuracy of our predictions is: **75.18%(4s.f)**.
Out of 1378 tag predictions, 1036 were correctly predicted.

```
Viterbi prediction accuracy:   1036/1378 = 0.7518142235123367
```

Question 5a)

The following are the methods we have implemented to improve performance:

1. Replace all tokens that are twitter username mentions with the format “@username” with a placeholder mention token of “-@mention-”
2. Replace all tokens that are start with hashtags with the format “#hashtagcontent” with a placeholder token of “-#hashtag-”
3. Replace all tokens that match the pattern of a url with the placeholder token “-httplink-”
4. Replace all tokens that match numerical patterns of with the placeholder token “-number-” with the regular expression pattern of r“\$[0-9]+”
5. Convert all words to lowercase.

These conversions are done on the fly while calculating output probabilities and when at inference. For methods 1-4, This is due to many of such token patterns being recognized as rare words in the training data and unknown words in the testing data as there are endless possible unique values despite mostly representing the same meaning. By mapping these patterns to placeholder tokens, the emission probabilities are more consolidated and new tokens matching these patterns will no longer be treated as unknown tokens.

Question 5c)

The accuracy of our predictions is: **80.99% (4s.f)**.
Out of tag predictions, 1116 were correctly predicted.

```
Viterbi2 prediction accuracy:  1116/1378 = 0.8098693759071117
```