

# NLP Problem Set 2

Rutvik Pansare, Kula Maganti, Tyler Arnett

## Q 2.1

Part 1.

	O-1 =1(ice cream)	O_1 =3(ice creams)	END  b.p. = 3 ice creams & H (0,1)
	H (0,0)  $v1(H) = P(\text{start} \rightarrow H) * P(1 H)$  $v1(H) = \frac{2}{3} * \frac{1}{6} = \frac{2}{18} = \frac{1}{9}$	H (0, 1)  $v2(H) = \max(v1(c), v1(H)) * P(3 H) * P(H C)$  $= \frac{1}{6} * \frac{1}{2} * \frac{1}{2} = \frac{1}{24}$  b.p. = 1 cream & cold (1,0)	
	C (1,0)  $v1(C) = P(\text{start} \rightarrow C) * P(1 C)$  $= \frac{1}{3} * \frac{1}{2} = \frac{1}{6}$	C (1,1)  $v2(C) = \max(v1(c), v1(H)) * P(3 C) * P(C C)$  $= \frac{1}{6} * \frac{1}{4} * \frac{1}{2} = \frac{1}{48}$ b.p. = 1 cream & cold (1,0)	
START			

Figure 1: Table

Part 2.

Q1. The tagger got 87.11% of the tags correct.

Q2. The baseline accuracy performed well in this case with an accuracy of around 95%. The baseline accuracy was computed by running the `nlk.pos_tag()` method on each word and finding the most common tag for the word. This result is stored in a lookup dictionary.

Q.3 The Viterbi algorithm should probably yield higher accuracy than "pick the most frequent POS"

because the meaning of the words depend on the context in which it is used. The Viterbi algorithm finds the POS tag by using the Markov chain assumption that if we want to predict the future in the sequence, all that matters is the current state. Hence the Viterbi algorithm keeps some form context in memory and calculates the most probable clause. On the other hand the most frequent POS tag method assumes that the POS tag will always be the most frequent tag which is an incorrect assumption. For example, the word "play" is usually marked as a verb as it is most commonly used as a verb but it can also mean a theatrical play in which case it is a noun. Hence the POS of a word highly depends on the context in which it is used which the Viterbi algorithm keeps in check and thus gives better result.

BONUS: Two tweets made by Obama were chosen for the US politician and 2 Shakespearean sentences were chosen. The output is in [2](#).

```

BOS At a time when voting rights are being rolled back , the Freedom to Vote Act would strengthen our democracy and
give more Americans a voice . EOS
BOS IN DT NN WRB NN NNS VBP VBG VBN RB , DT NNP TO NNP NNP MD VB PRP$ NN CC VB JJR NNS DT NN . EOS
BOS DT DT NN WRB NN NNS VBP VBG VBN RB , DT NNP TO VB NNP MD VB PRP$ NN CC VB JJR NNPS DT NN . EOS

BOS Many of the refugees fleeing Afghanistan stood by America and risked everything for a chance at a better life .
EOS
BOS JJ IN DT NNS VBG NNP NN IN NNP CC VBD NN IN DT NN IN DT JJR NN . EOS
BOS DT IN DT NNS VBG NNP VBD IN NNP CC NNP NN IN DT NN IN DT JJR NN . EOS

BOS He hath eaten me out of house and home EOS
BOS PRP VBD VB PRP IN IN NN CC NN EOS
BOS DT NN VBN PRP IN IN NN CC NN EOS

BOS In compliment extern , 't is not long after But I will wear my heart upon my sleeveFor daws to peck . EOS
BOS IN JJ NN , ' VBZ RB RB IN CC PRP MD VB PRP$ NN IN PRP$ NN NN TO VB . EOS
BOS DT NN IN , DT VBZ RB JJ IN CC PRP MD VB PRP$ NN IN PRP$ NN IN TO VB . EOS

Viterbi Accuracy: 0.8679706601466992

```

Figure 2: Output

As we can see there was a drop in precision by 1% i.e., the false positives were more when complicated English sentences were tested on the POS tagger. The tweet1 was tagged with around 4 incorrect tags hence accuracy is 85% Tweet2 was tagged with 3 incorrect tags so accuracy is 85% Shakespeare 1 was tagged with 3 in-correct tag and the accuracy was 70% Shakespeare 2 was tagged with 4 incorrect tags with accuracy of around 81% Hence we can see that when the Viterbi encounters a new word, the accuracy of the program reduces as it does not have the emission probabilities for that word. This scenario happened in the Shakespeare's sentences. The impact was more likely on false positives as the Viterbi algorithm will try to maximize the tag with highest state probabilities.

## References