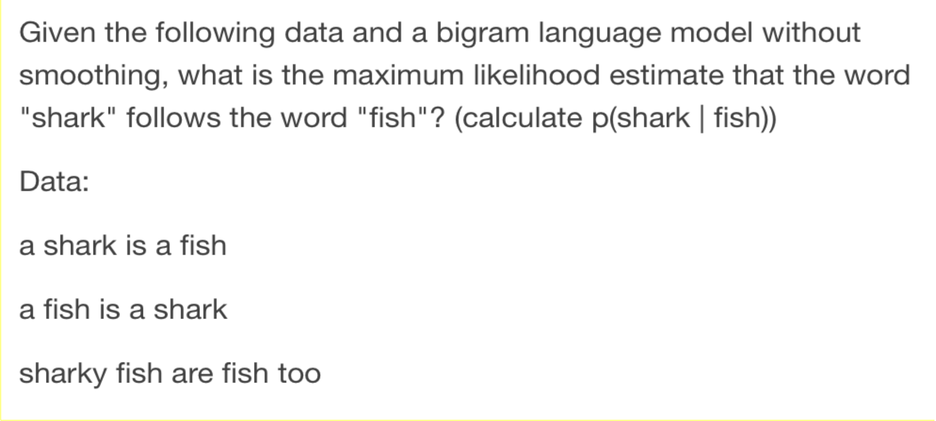
Name: Chakrya Ros

Section 1: Data & Pre-processing (10 points)

1. Describe in 2 - 3 sentences what a vocabulary is for an NLP system and what an unknown word is for NLP systems. Next, make an argument (2 - 3 sentences) about whether dealing with unknown words is relevant for [**no**, **some**, or **all**] NLP tasks.

- **A vocabulary for an NLP system refers to the lexicon that help in the preprocessing of the corpus text. It is the set of types that serve as storage location in memory for processed text corpus or collect and store metadata about the corpus.**
- **The unknown word for NLP systems refers to the word that we didn't find or unseen in the vocabulary or dictionary in training set.**
- **Dealing with unknown words is some for some NLP tasks because our training data will never contain all possible tokens. If the test set is given the unknown words, they would make some of NLP tasks error. For example, in language model that we did the homework2 for N-gram. If we didn't apply Laplace smoothing and the words have not seen in training data, it would be always given zero for calculate the conditional probability of words. Below is the example that we did in quiz 2 for language model without smoothing. The maximum likelihood estimate that the word "shark" follows the word "fish" is always zero. So I would say we have to deal with unknow words.**



Given the following data and a bigram language model without smoothing, what is the maximum likelihood estimate that the word "shark" follows the word "fish"? (calculate p(shark | fish))

Data:

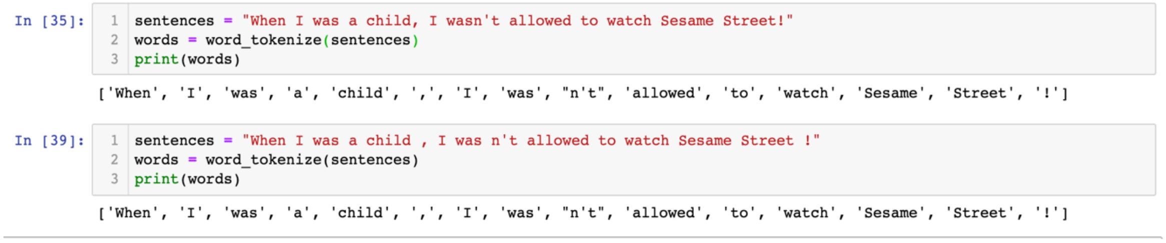
a shark is a fish

a fish is a shark

sharky fish are fish too

2. Name one effect the following two differences in tokenization might have on a downstream task and explain why it would happen (2 - 3 sentences).

Word tokenize from nltk.tokenize is effect the following two differences in tokenization that have on a downstream task. I applied these two sentence on word_tokenize from the nltk libray , I got the same output as shown below.



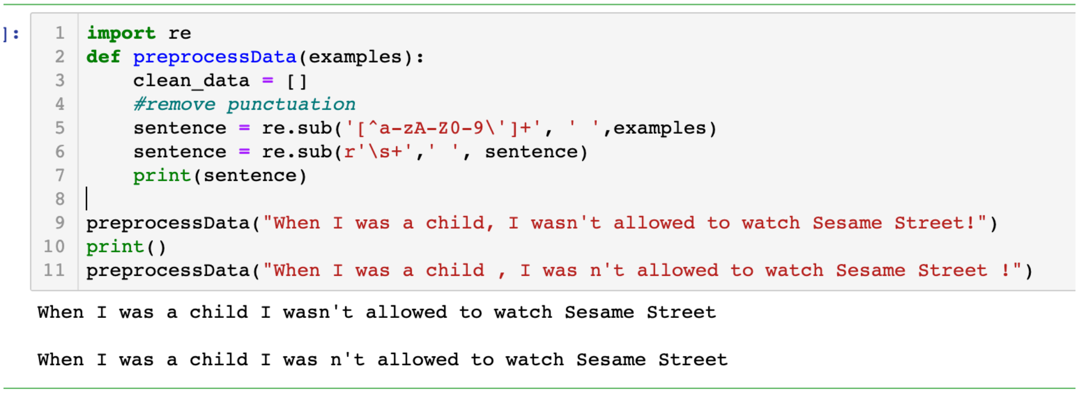
```
In [35]: sentences = "When I was a child, I wasn't allowed to watch Sesame Street!"
         words = word_tokenize(sentences)
         print(words)

['When', 'I', 'was', 'a', 'child', ',', 'I', 'was', "n't", 'allowed', 'to', 'watch', 'Sesame', 'Street', '!']

In [39]: sentences = "When I was a child , I was n't allowed to watch Sesame Street !"
         words = word_tokenize(sentences)
         print(words)

['When', 'I', 'was', 'a', 'child', ',', 'I', 'was', "n't", 'allowed', 'to', 'watch', 'Sesame', 'Street', '!']
```

If I don’t use the word tokenize from the pre-trained, these two sentences will produce different output. As shown below, I applied regex to split the word and remove punctuation and I got difference output.



```
]: import re
   def preprocessData(examples):
       clean_data = []
       #remove punctuation
       sentence = re.sub('[^a-zA-Z0-9\']+', ' ',examples)
       sentence = re.sub(r'\s+',' ', sentence)
       print(sentence)

   preprocessData("When I was a child, I wasn't allowed to watch Sesame Street!")
   print()
   preprocessData("When I was a child , I was n't allowed to watch Sesame Street !")

When I was a child I wasn't allowed to watch Sesame Street

When I was a child I was n't allowed to watch Sesame Street
```

Sentence 1, I got 12 tokens because “wasn’t” count as one, and sentence 2, I got 13 tokens because “was” and “n’t” count as two. Therefore, the word tokenize from the pre-trained is effect on the downstream task for the language model and sentiment analysis.

3. **True** or **False:** Given infinite training data, a language model will not have to deal with unknown words.

False

Citations:

- Slide 16 in Lecture 3
- Quiz 2
- https://www.kdnuggets.com/2019/11/create-vocabulary-nlp-tasks-python.html
- https://www.aclweb.org/anthology/P16-1014.pdf