1. Answer the following to the best of your ability:(**10points**)

a) Define Corpus:

corpus is body of text

corpus analysis is the technique to handle large quantities of text, thereby spotting patterns and styles of speech.

The corpus was analysed using three techniques: keyword analysis, examining the frequency of significant words in comparison to other corpora; collocation analysis, investigating which words are used together and offering insights into the meanings given to key terms; and semantic tagging, which compares the relative frequency of use of groups of words which share similar meanings

b) How might you make a corpus for the following problem: I want to be able to learn characteristics of a politician’s language

Keyword analysis: Keyword analysis reveals which words occur frequently in the text under study

Collocation: I analysed keywords further through looking at their collocates, or words which frequently occur together with the keyword

Semantic tagging: words grouped into categories of meaning, or semantic fields, to look at which semantic groupings are used more or less frequently in comparison to another corpus

2. a) Describe briefly 4 difficulties with identifying word boundaries algorithmically?(**8points**)

i) The characters obtained are not necessarily the tokens required for further processing.

ii) ambiguity exists among uses of punctuation marks, such as periods, commas, quotation marks, apostrophes, and hyphens, since the same punctuation mark can serve many different functions in a single sentence

iii) some definitions contain (such as in NUMBER PREFIX, where \243 and \245 represent the currency symbols for usd or inr)

iv) The nature of the tokenisation task in unsegmented languages like Chinese, Japanese is hard than delimited language like english

b) What is the possible differences in the following two implementations of a word identifier(**5points**)

tokens = nltk.word\_tokenize(sentence)

and

tokens = sentence.split(“ “)

nltk.word\_tokenize(sentence) gets all possible words from sentence

split divides the sentence to text separated by space

c) Why do we use ‘tokens’ instead of ‘word’(**5points**)

token can be anything like verb,noun,sign,etc

word is some actionable item only

3. With the following sentence “The Cat in the Hat”(**12points**)

a) List the Uni-grams

[“the”,”cat”,”in”,”the”,”Hat”]

b) List the Bi-grams

[(“the”,”cat”) ,(”cat”,,”in”),(”in”,”the”),(”the”,”Hat”)]

c) List the Tri-grams

[(“the”,”cat”,”in”)(“cat”,”in”,”the”),(“in”,”the”,”Hat”)]

4. Answer the following about predictive models:(**10points**)

a) What is a backoff model?

We go back and calculate the probability of next word in the text. This we can group previous text as unigram,bigram,etc and predict the future probability. If the new word occurs with 0 probaility it is reffered ad bad else good

b) Give an example of how a backoff may help your model.

Let's say you are using 4-grams to calculate the probability of a word in text. You have "this is a very" followed by "sunny". Let's say "sunny" never ocurred in the context "this is a very" so for the 4-grams model "sunny" has probability 0, and that's not good because we know that "sunny" is more probable than say "giraffe".

Backoff means you go back to a n-1 gram level to calculate the probabilities when you encounter a word with prob=0. So in our case you will use a 3-gram model to calculate the probability of "sunny" in the context "is a very".

5. Why do we need *sent\_tokenize\_list = sent\_tokenize(text)* in NLTK instead of just breaking sentences apart by punctuation?(**5points**)

Because useless words like punctuation , signs, etc are taken as tokens

6. Briefly explain Transformation Based Tagging and how it differs from Ngram tagging for Part-of\_Speech (**8points**)

7) Answer the following:(**12points**)

a)What is a False Negative?

The predication should be true but actual is false

b)What is a True Positive?

The prediction and actual are true

c)When should Accuracy be used as a metric?

a weighted arithmetic mean of Precision and Inverse Precision (weighted by Bias) as well as a weighted arithmetic mean of Recall and Inverse Recal

These used to evaluate the text mining algorithms

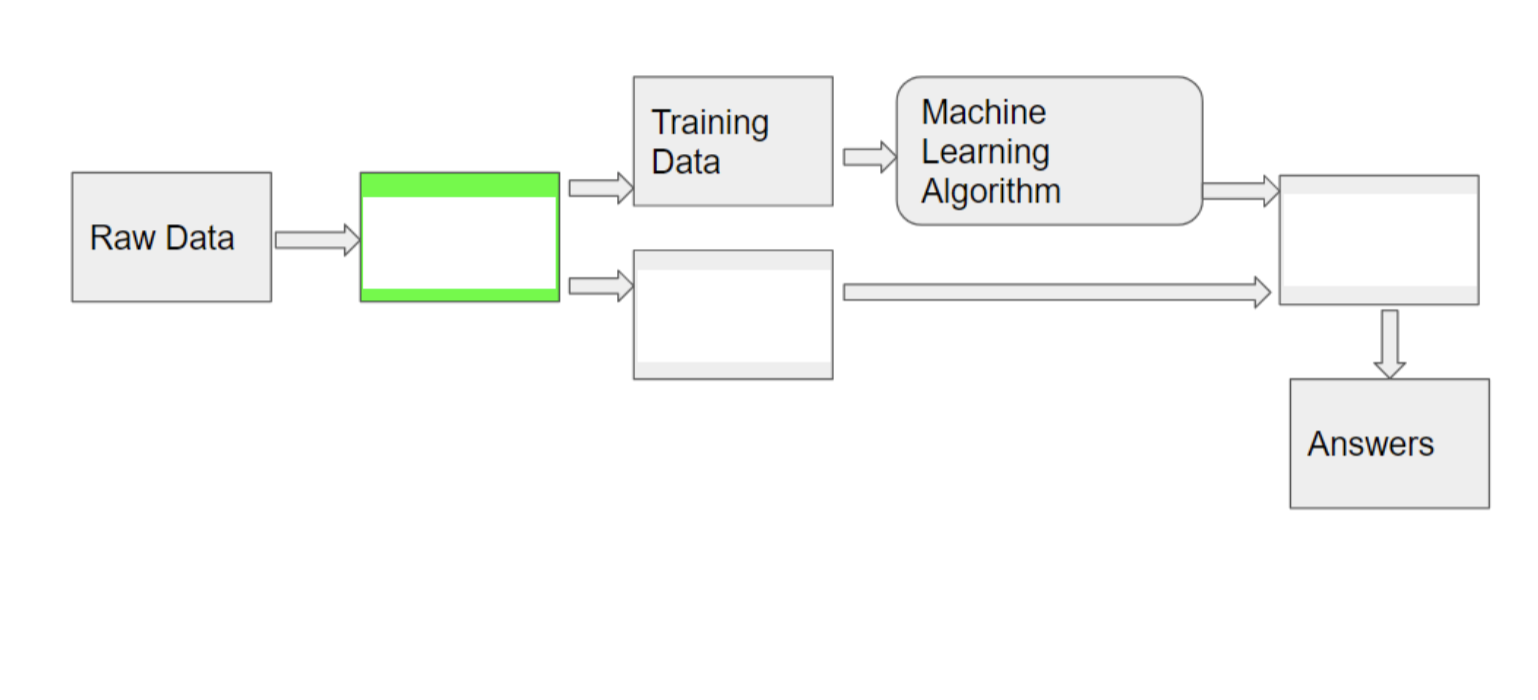
d)What is the difference between Precision and Recall? When would you use them?

for a text search on a set of documents, precision is the number of correct results divided by the number of all returned results.

text search on a set of documents, recall is the number of correct results divided by the number of results that should have been returned

These used to evaluate the text mining algorithms

8. Fill in the 3 empty boxes for a typical machine learning cycle:(**9points**)



9. What are the two differences when you test on your training data versus testing on your test data? (**4points**)

Testing known data has high prediction rate than unknown data

10. Explain (or draw) k-fold validation when k=5 (**6points**)

The general procedure is as follows:

1. Shuffle the dataset randomly.
2. Split the dataset into k groups(k=5)
3. For each unique group:
   1. Take the group as a hold out or test data set
   2. Take the remaining groups as a training data set
   3. Fit a model on the training set and evaluate it on the test set
   4. Retain the evaluation score and discard the model
4. Summarize the skill of the model using the sample of model evaluation scores

11. Show 3 examples where a Named Entity System can get confused by ambiguity (**6points**)

The pen is good. This cost 10 USD.

The 10 $ is assigned to unknown entity

The flowers are good . I like the lilly

Lily is never known as flower

The politician speech changed over time. Kennedy and Robert seems uninsterested

Kennedy and Robert can be politicians or reporters

**PART II**

1. Use given script to download 1 Wikipedia page
2. Run NLTK’s NER tool.
3. Examine both the PERSON and LOCATION Entities
4. Calculate Precision, Recall, and F-Measure for both Person or Location (whichever your document has)

pip install pymediawiki

To Harvest WIKI

from mediawiki import MediaWiki  
wikipedia = MediaWiki()  
  
p = wikipedia.page('Marymount University')  
content= p.content  
print(content)  
  
  
  
  
To Tag Named Entities  
1) Word Tokenize a Sentence  
2) POS tag the tokens  
3) print(nltk.ne\_chunk(tagged))

from mediawiki import MediaWiki

import nltk

nltk.download('maxent\_ne\_chunker')

from nltk.corpus import stopwords

from nltk.tokenize import word\_tokenize, sent\_tokenize

stop\_words = set(stopwords.words('english'))

wikipedia = MediaWiki()

p = wikipedia.page('Marymount University')

content= p.content

print(content)

tokenized = sent\_tokenize(content)

for i in tokenized:

# Word tokenizers is used to find the words

# and punctuation in a string

wordsList = nltk.word\_tokenize(i)

# removing stop words from wordList

wordsList = [w for w in wordsList if not w in stop\_words]

# Using a Tagger. Which is part-of-speech

# tagger or POS-tagger.

tagged = nltk.pos\_tag(wordsList)

print(tagged)

print(nltk.ne\_chunk(tagged))