NLP Non-Deep Theoretical Question

1. **Explain with your own words, using a short paragraph for each, what are**
    a. **Phonetics:** is a branch of linguistics that studies how humans produce and perceive sounds. Phonetics deals with two aspects of human speech: production—the ways humans make sounds—and perception—the way speech is understood.
    b. **Phonology:** Phonology is a branch of linguistics that studies and classifies human sound based on the culture and language. For example, water in an American context the 't' sounds would not be as pronounced as in a British context. However, the distinction is not relevant in a phonological context since it delivers the same meaning to the word water. On the other hand, the word paro (I stop) and pato (I duck) in Spanish although one sound in the middle is changed very subtly it creates different meanings to 2 almost the same sound.
    c. **Morphology** is the study of words, how they are formed, and their relationship to other words in the same language. It analyzes the structure of words and parts of words such as stems, root words, prefixes, and suffixes. Morphology also looks at parts of speech, intonation and stress, and the ways context can change a word's pronunciation and meaning.
    d. **Syntax** is the arrangement and structure of a sentence. It provides the definition of a sentence that helps the speaker and the listener communicate and understand. For example, we know that in a basic sentence, there are subject (noun) and predicate (verb). Elements of syntax include word order and sentence structure, which can help reveal the function of an unknown word.
    e. **Semantics** is the meaning of individual words. If one word is unknown, the meanings of surrounding words can give clues to the word's probable meaning. The semantics of an unknown word can be scattered and covered through multiple sentences which we call 'context'.
    f. **Pragmatics** is a field of linguistics concerned with what a speaker implies and a listener infers based on contributing factors like the situational context, the individuals' mental states, the preceding dialogue, and other elements.
2. **What is the difference between stemming and lemmatization?**
    a. **Stemming:** Stemming algorithms work by cutting off the end or the beginning of the word, considering a list of common prefixes and suffixes that can be found in an inflected word. This indiscriminate cutting can be successful on some occasions, but not always, and that is why we affirm that this approach presents some limitations.
        i. Cons:
            1. **Over-stemming** is when two words with different stems are stemmed to the same root. This is also known as a false positive. For example, universal, university and universe are stemmed to univers which is wrong behavior. Though these three words are etymologically related, their modern meanings are in widely different domains, so treating them as synonyms in NLP/NLU will likely reduce the relevance of the search results
            2. **Under-stemming** is when two words that should be stemmed to the same root are not. This is also known as a false negative. For example, alumnus,

alumni, alumnae all have the same stem but their suffices are not common enough to be stemmed.

ii. Pros: In general, the advantages of stemming are that it's straightforward to implement and fast to run. The trade-off here is that the output might contain inaccuracies, although they may be irrelevant for some tasks, like text indexing.

b. Lemmatization: takes into consideration the morphological analysis of the words. To do so, it is necessary to have detailed dictionaries which the algorithm can look through to link the form back to its lemma

i. Pros: lemmatization provides better results by performing an analysis that depends on the word's part-of-speech and producing real, dictionary words. As a result, lemmatization is harder to implement and slower compared to stemming.

ii. Cons: Computationally intensive and takes a lot of time to run.

3. On logistic regression:

a. Stochastic Gradient Descent is a probabilistic approximation of Gradient Descent. It is an approximation because, at each step, the algorithm calculates the gradient for one observation picked at random, instead of calculating the gradient for the entire dataset. Compared to Gradient Descent, Stochastic Gradient Descent is much faster, and more suitable to large-scale datasets.

b. The learning rate α determines how rapidly we update the parameters. If the learning rate is too large, we may "overshoot" the optimal value. Similarly, if it is too small, we will need too many iterations to converge to the best values.

c. Since the cost function for logistic regression is convex, gradient descent will always converge to the global minimum.

4. What problems does TF-iDF try to solve?

a. **Term frequency** works by looking at the frequency of a *particular term* you are concerned with relative to the document. There are multiple measures, or ways, of defining frequency:

i. Number of times the word appears in a document (raw count).

ii. Term frequency adjusted for the length of the document (raw count of occurrences divided by number of words in the document).

iii. Logarithmically scaled frequency (e.g. log(1 + raw count)).

iv. Boolean frequency (e.g. 1 if the term occurs, or 0 if the term does not occur, in the document).

b. **Inverse document frequency** is a measure of how much information the word provides, i.e., if it is common or rare across all documents. The closer it is to 0, the more common a word is. This metric can be calculated by taking the total number of documents, dividing it by the number of documents that contain a word, and calculating the logarithm.

Multiplying these two numbers of results in the TF-IDF score of a word in a document. The higher the score, the more relevant that word is in that document. TF-IDF enables us to associate each word in a document with a number that represents how relevant each word is in that document. Then, documents with similar, relevant words will have similar vectors, which is what we are looking for in a machine learning algorithm.

5. **Summarize how the skip-gram method of Word2Vec works using a couple of paragraphs.**

The main idea behind the Skip-Gram model is: it takes every word in a large corpora (we will call it the focus word) and takes one-by-one the words that surround it within a defined 'window' to then feed a neural network that after training will predict the probability for each word to appear in the window around the focus word.

If two different words have very similar "contexts" (that is, what words are likely to appear around them), then the skip-gram model needs to output very similar results for these two words. And one way for the network to output similar context predictions for these two words is if *the word vectors are similar*. So, if two words have similar contexts, then the model is more likely to learn similar word vectors for these two words.

6. **What are the differences between a RNN and a LSTM?**

The basic difference between the architectures of RNNs and LSTMs is that the hidden layer of LSTM is a gated unit or gated cell. It consists of four layers that interact with one another in a way to produce the output of that cell along with the cell state. These two things are then passed onto the next hidden layer. Unlike RNNs which have got the only single neural net layer of tanh, LSTMs comprises of three logistic sigmoid gates and one tanh layer. Gates have been introduced in order to limit the information that is passed through the cell. They determine which part of the information will be needed by the next cell and which part is to be discarded.

RNNs are particularly suited for tasks that involve sequences (thanks to the recurrent connections). For example, they are often used for machine translation, where the sequences are sentences or words. The LSTM was introduced to solve a problem that standard RNNs suffer from, i.e. the vanishing gradient problem. The vanishing gradient problem is essentially a situation in which a deep multilayer feed-forward network or a recurrent neural network (RNN) does not have the ability to propagate useful gradient information from the output end of the model back to the layers near the input end of the model. It results in models with many layers being rendered unable to learn on a specific dataset. It could even cause models with many layers to prematurely converge to a substandard solution.

7. **What would you expect if we used one of our classifiers trained on IMDB on Twitter data, and why?**

The classifier would not do as well since here in the Twitter Sentiment140 dataset provides a multiclass classification problem of degree:

- 0: Negative
- 2: Neutral
- 4 : Positive

Which requires more nuance in the parameters to detect slight difference in polarity in each tweet. Moreover, the dataset is quite larger with 1.6 million tweets which will need more time to train in order to have sufficient accuracy on the classification. Our current model is geared towards binary classification which works really well in the extreme but not neutral tone sentence.