**Theorical questions :**

Question 1 :

Word-based tokenzation contains a very large vocabularies, large quantity of out of vocabulary tokens and loss of meaning across very similar words. The character-based tokenization contains a very long sequences and less meaningful individual tokens. Subword tokenization's primary goal is to strike a balance between character-based and word-based algorithms. Subword tokenization enables the model to acquire meaningful context-independent representations while maintaining a manageable vocabulary size. Additionally, by breaking down words into their constituent subwords, subword tokenization enables the model to process words it has never encountered before.

Question 2:

The encoder-decoder model has a limitation which is the model encodes all the information of the input sentence into a fixed-length vector and then decodes it each output time step. Therefore, it would make the model performance bad when processes long input sentences.

To solve this problem, “attention” mechanism is proposed to the encoder-decoder model. The attention mechanism will obtain the significance of each input sequences, then it will be able to give a more weighted or more signified context of the sequences from the encoder to the decoder, and score each encoding based on how well they match the current output of the decoder. Thanks to this information, the decoder knows whether to give more attention to the current encoding when predicting outputs. Overall, with the upgrade from the attention mechanism, the encoder-decoder model with attention can learn better and have a better understanding of the context of the input sequences, therefore will result in better performance, especially with long input sentences.

Question 3:

By using multiple heads in the encoder and decoder of the transformer model, it allows the model to understand the input sequences in multiple ways and combine the information of different input sequences, this helps the model creates more detailed output, hence increasing the performance of the model.

Multihead attention mechanism provides more stable training and better performance than using singlehead attention in transformer. The reason why is multihead attention has less layers than singlehead attention when attending to the same number of positions.

Question 4:

To describe the position of a specific word or a character in a sentence so that the model understands the context and the sense of order of the word/character in the sentence, we use positional encoding.

Different from the RNNs, which input the sequence word by word, therefore helps the model has an proper understanding of the word’s order, the Transformers input each word in a sequence at the same time. As a result, the Transformer model has a huge increase in training time, but it loses the word’s order understanding. Which is the reason why we need to use positional encoding in the transformer model to improve the model’s performance.

Question 5:

Benchmarking is a method to find out the model’s current performance. The method involves evaluation and comparison. First of all, the model is evaluated based on its abilities to learn and to perform as expected. Then, the model is compared to other Machine Learning models over multiple metrics such as: prediction accuracy, training time… to identify the strengths and weaknesses of the model. Benchmarking is important to identify which model performs better in different problems and what are there to improve in the model.

Question 6:

|  |  |
| --- | --- |
| **BERT** | **GPT** |
| * BERT is bi-directional, it can process text from left-to-right or right-to-left * BERT uses the Encoder part of the Transformer model * BERT is trained on Masked Language Model (MLM) and Next Sentence Prediction * BERT has pre-trained models for different downstream NLP tasks which can be further fine-tuned on custom data | * GPT is autoregressive and unidirectional. Texts are processed in one direction * GPT uses the Decoder part of the Transformer model * GPT is trained on traditional Language Model problem which predicts next word * GPT has a single model for all downstream tasks and does not require fine-tuning |

Question 7:

The model is initialized using the pre-trained weights and trained on the target (often smaller) dataset when you already have a model trained to accomplish the task you desire but on a different dataset (usually with a smaller learning rate). The model is initialized using the pre-trained weights and trained on the target (often smaller) dataset when you already have a model trained to accomplish the task you desire but on a different dataset (usually with a smaller learning rate).

When you wish to train a model on any task using a small number of samples, use few shot learning. For instance, you might have a model that has been trained on a separate but related task, change it (optionally), and train it for the target task using a limited number of instances.

Question 8:

Triplet loss is a loss function that is defined by triplets of embeddings:

* Anchor data point
* Positive data point
* Negative data point

In order to minimize triplet loss, we need to minimize the distance between the anchor data point and positive data point and maximize the distance between the anchor data point and the negative data point. These actions are done simultaneously.

In the case of training a bi-encoder model for semantic similarity, the data is defined as triplet of an anchor sentence, a sentence that has the similar meaning as the anchor sentence, which will be the positive data point, and a sentence that no similarity to the anchor sentence, which will be the negative data point.

When training starts, the model will tune the sentences embeddings in a way such that the anchor sentences is as close as possible to the positive data points and as far as possible to the negative data points. Therefore, after training the embeddings of sentences that are similar will be closer to each other and further from dissimilar sentence embeddings.

Question 9:

Approximate Nearest Neighbor (ANN) can be useful in the situation. In this case, the data is divided into smaller groups of related embeddings. Even if you have millions of vectors, this index can be effectively searched, and the embeddings with the highest similarity (the closest neighbors) may be recovered in milliseconds. The outcomes, however, are not always precise. Some vectors with a high degree of similarity might be overlooked. It is known as the approximate nearest neighbor for this reason.  The recall-speed trade-off is typically controlled by one or more parameters that can be tuned for all ANN approaches. You run a considerable risk of missing hits if you desire the fastest speed. Search speed slows down if good recall is what you're after.

**ANNOTATION GUIDELINE**

Our target class is Hate tweets. Hate speech in tweets are defined by tweet that expresses hate or encourages violence towards a person or group based on things such as races, religions, sex, or genders. Some examples of hate speech in tweets are:

* 'I ACTUALLY HATE YOU! YOU WOMEN ARE LAZY, SELF ABSORBED DISGUSTING PEOPLE. I NEVER CELEBRATE ANYTHING BUT I WILL CELEBRATE THE DAY YOU DIE'
* 'Kamala is useless. Tits on a Bull are more productive than Kamala'

Different from Offensive tweets, hate tweets target a particular person or group. Tweets that are considered offensive cause someone to feel insulted, annoyed or resentful. Some examples of tweets that are considered offensive instead of hate:

* 'also east asian girls who preach inclusivity in public but use the n word in private are so fucking weird like i'm ashamed!'
* 'idfk most girls confuse me😭'

Tweets that are considered offensive are not considered as hate tweets, therefore they will be classified as non-hate.

It is difficult to specify which tweet is a hate tweet or an offensive tweet. We need to be able to understand the context of the tweet, to see if the tweet is targeting a particular person or a group of people, and does the tweet offend, insult or encourage violence. Therefore, there are some cases that are ambiguous and difficult to classify:

* 'Lock her up!!'
* 'the ultimate girl boss'

We classify tweets like these as Can’t tell/ not annotable. This class contains tweets that are too ambiguous to classify since we do not have enough context or a tweet that is neither hate nor offensive.