Applied NLP - Project - [Marks 10]

Note: Each Question carries 0.25 marks including all sections.

I. True or False

1. __True__
2. __True__
3. __True__
4. __True__
5. __True__
6. __False__
7. __False__
8. __False__
9. __True__
10. __False__
11. __True__
12. __False__
13. __False__
14. __False__
15. __True__

II. Fill in the Blank

1. **_adjudicator_**
2. **_Stop words_**
3. **_6_**
4. **_bigram_**
5. **_0.5_**
6. **_0.55_**
7. **__38_**
8. **__1.0___**
9. **_Attention mechanism _**
10. **0.609_**

III. Multiple Choice

1. **a. AutoModelForTokenClassification**
2. **b. Word tokenization**
3. **b. High tf and low idf.**
4. **a. w1 and w2 are closer together than w1 and w3.**
5. **a. [0, ∞)**
6. **c. 0.5**
7. **a. GPT uses the decoder and BERT uses the encoder.**
8. **d. It can be a square matrix.**
9. **c. Sequence-to-Sequence**
10. **d. "SGD. an optimizer. It learns. errors."**

IV. Short Answer

1. Training the TfidfVectorizer on the training data is essential to ensure that the vectorization process accurately reflects the characteristics of the specific training dataset. It allows the vectorizer to learn the vocabulary from the training data, calculate the inverse document frequency (IDF) based on the entire training corpus, and establish a consistent transformation for both the training and test datasets. By fitting the vectorizer to the training data, data leakage from the test set is prevented, and realistic representations of documents are obtained, making it a fundamental step in building machine learning models for text data.

2. The annotation guidelines should include essential information such as the task's description, the annotation schema or labelling rules, exemplar annotations, instructions for addressing ambiguity, and quality control measures. These elements ensure clarity and consistency in the annotation process.

3. Yes, it is possible to make Beam Search behave as Greedy Search. One way to do this is to set the beam size to 1. This means that at each time step, only the single best word will be kept. This is equivalent to the Greedy Search algorithm, which always selects the word with the highest probability at each time step.

4. During BERT pre-training, the [CLS] token has a dual role: it is used for sentence classification in the "Next Sentence Prediction" task, where BERT learns to predict whether two input sentences are consecutive or not, and it also provides an aggregated representation of the entire input sequence. This aggregated representation is valuable for various downstream tasks and can be fine-tuned for text classification, named entity recognition, and other NLP tasks, making the [CLS] token a central element in BERT's pre-training and its versatility in handling various language understanding tasks.

5. To help distinguish training examples from different tasks when pre-training a multitask sequence-to-sequence model, several approaches can be employed. One effective strategy is to incorporate task-specific tokens or prefixes in the input sequences, indicating the associated task for each example. Additionally, using separate training datasets for distinct tasks, introducing auxiliary objectives that predict task identification, implementing task dropout, and designing task-specific architectural components, such as multi-head models, all contribute to task differentiation. Including domain or metadata tags and organizing mini-batches by task further aids in emphasizing the task-specific learning, ensuring that the model can effectively handle a range of sequence-to-sequence tasks in a multitask setting.

Happy Learning