Applied NLP - Project - [Marks 10]

Note: Each Question carries 0.25 marks including all sections.

I. True or False

1. True
2. False
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. False

II. Fill in the Blank

1. Adjudicator
2. Stop words
3. 6
4. Bigram
5. 0.8
6. 0.55
7. 28
8. 1.0
9. Self-attention
10. 0.628

III. Multiple Choice

1. a

2. b

3. c

4. a

5. a

6. c

7. a

8. d

9. c

10. d

IV. Short Answer

1. Training 'TfidfVectorizer' on training data helps it acquire vocabulary and IDF weights specific to the corpus. This ensures that training and test data are consistently transformed, successfully capturing term frequencies and informativeness. By fitting to training data, 'TfidfVectorizer' creates a foundation for feature extraction that ideally represents textual data for machine learning models, improving performance and interpretability.

2. Ensure annotation rules include explicit criteria and examples for each category. They should provide thorough guidance about annotation techniques and tools. Guidelines must specify how to handle confusing instances in order to ensure consistency. They should also include processes for feedback and resolving annotation disputes among annotators in order to ensure the quality and dependability of annotated data.

3. Beam Search may be made to function similarly to Greedy Search by setting the beam width parameter ('k') to one. This basically limits Beam Search to selecting only the highest-scoring candidate at each decoding stage, similar to Greedy Search. By restricting the beam width to one, the search algorithm favors the most likely sequence continuation at each step among other options, emulating Greedy Search's characteristic of making just the locally optimum decision without looking ahead.

4. The '[CLS]' token in BERT is intended for classification tasks. It holds the aggregated representation of the entire input sequence after it has been processed through BERT's layers. When fine-tuning for text categorization, the final hidden state of the '[CLS]' token is used.

5. The representation corresponding to `[CLS]` is frequently used as input to a classifier to predict the label or category of the input text. This representation, learned during pre-training, encodes contextual information and semantics from the sequence, making it a powerful feature for downstream tasks. Its use ensures that BERT can effectively adapt to various classification tasks by leveraging the encoded understanding of the input text provided by the `[CLS]` token.

6. To differentiate training examples from various tasks during the pre-training of a multitask sequence-to-sequence model, several strategies can be utilized:
   1. Task Embeddings: Integrate task-specific embeddings or indicators appended to input sequences, enabling the model to learn task-specific nuances.
   2. Task Identifiers: Prefix each input sequence with a task identifier or label to explicitly indicate the task associated with each example.
   3. Task-Aware Loss Functions: Use task-specific loss functions or weighting schemes during training to appropriately prioritize learning from each task.
   4. Task-Specific Parameters: Implement separate parameters or layers in the model dedicated to each task, allowing the model to specialize in different tasks while sharing common knowledge.
   5. Data Segregation: Train the model on batches containing examples from a single task at a time to facilitate task-specific learning without interference from other tasks.