Natural Language Processing

Tutorial 10 (Week 13): Pretraining / Prompting

Question 1

The Masked Language Model (MLM) pretraining objective is central to the functionality of models like BERT. MLM randomly masks out tokens in the input data and trains the model to predict these masked tokens. Given the sentence "The quick brown fox jumps over the lazy dog", assume that during MLM pretraining, **2** tokens are being masked.

1) Construct an example input sequence for the MLM pretraining given the sentence "The quick brown fox jumps over the lazy dog".

Solution 1 (1)

1) Given the sentence "The quick brown fox jumps over the lazy dog", two tokens could be masked, e.g., “brown”, “over”. Then the input sentence becomes: "The quick [MASK] fox jumps [MASK] the lazy dog".

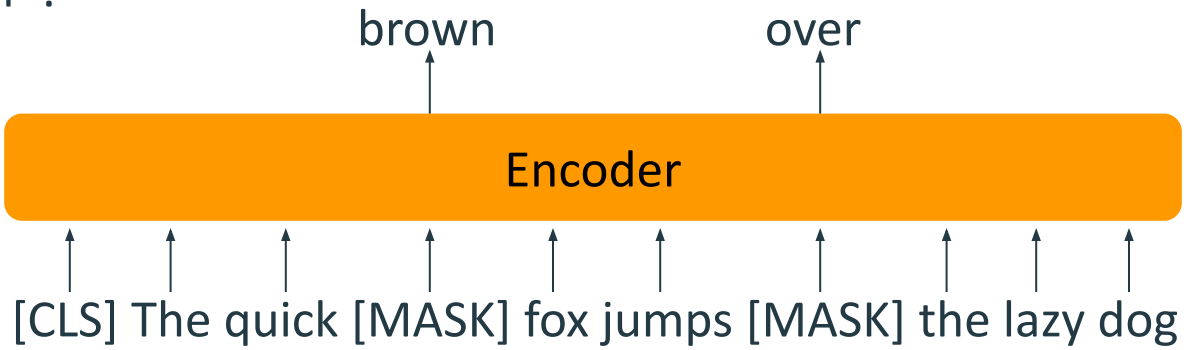
Question 1

The Masked Language Model (MLM) pretraining objective is central to the functionality of models like BERT. MLM randomly masks out tokens in the input data and trains the model to predict these masked tokens. Given the sentence "The quick brown fox jumps over the lazy dog", assume that during MLM pretraining, **2** tokens are being masked.

2) For the input sequence you have created, specify what the expected output labels would be during the training. What does the objective function look like?

Solution 1 (2)

- For the input sequence with masked tokens: "The quick [MASK] fox jumps [MASK] the lazy dog", the expected output labels during training would be the original tokens that were masked. So for the above sentence, the expected labels would be "brown" and "over".



Solution 1 (2)

- The objective function during training is to minimize the cross-entropy loss between the predicted probabilities of the masked tokens and the actual tokens. The objective function for the two masked tokens can be represented as:

$$L(\theta) = -\log y_{brown}^{(3)} - \log y_{over}^{(6)}$$

- $y^{(3)}$ and $y^{(6)}$ are the predicted probability distributions over the entire vocabulary at position 3 and 6.

Question 1

The Masked Language Model (MLM) pretraining objective is central to the functionality of models like BERT. MLM randomly masks out tokens in the input data and trains the model to predict these masked tokens. Given the sentence "The quick brown fox jumps over the lazy dog", assume that during MLM pretraining, **2** tokens are being masked.

3) Identify two downstream tasks where a pretrained MLM model like BERT would be expected to excel and provide reasons based on the features of the MLM pretraining process.

Solution 1 (3)

- **Named Entity Recognition (NER)**: BERT would excel at NER because MLM pretraining helps it learn rich contextual representations for each token in a sentence via a bidirectional encoder. BERT's understanding of context allows it to effectively distinguish between entities and non-entities.
- **Sentiment Analysis**: The bidirectional training strategy, which means each word is learned based on all other words, not just the words before it leads to a deeper understanding of the sentence which is beneficial for sentiment analysis.

Question 1

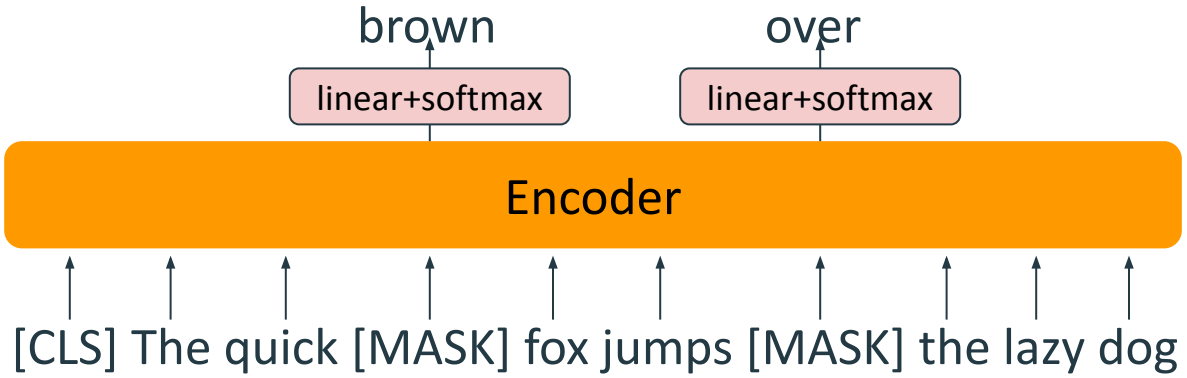
The Masked Language Model (MLM) pretraining objective is central to the functionality of models like BERT. MLM randomly masks out tokens in the input data and trains the model to predict these masked tokens. Given the sentence "The quick brown fox jumps over the lazy dog", assume that during MLM pretraining, **2** tokens are being masked.

4) Describe the steps involved in fine-tuning an MLM-pretrained model for the downstream tasks you have identified. What modifications, if any, are typically made to the model architecture during fine-tuning?

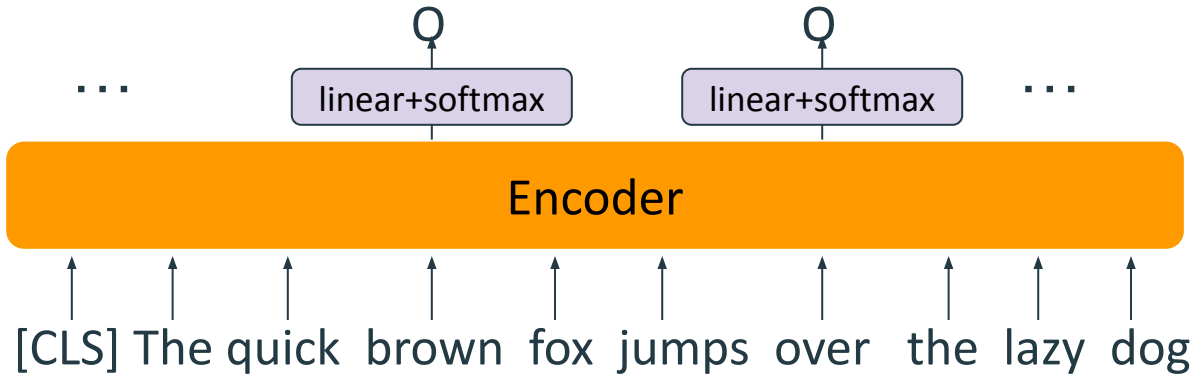
Solution 1 (4)

- The pretrained MLM typically has a final linear layer followed by a softmax that predicts the probability of each token in the vocabulary for the masked positions.
- For NER, you remove this layer and replace it with a new linear layer that has an output size equal to the number of classes for entity prediction. Then a softmax function is applied on top.

$$y_i = softmax(Wh_i + b), W \in \mathbb{R}^{|V| \times d}$$



$$y'_i = softmax(W'h_i + b'), W' \in \mathbb{R}^{|C| \times d}$$



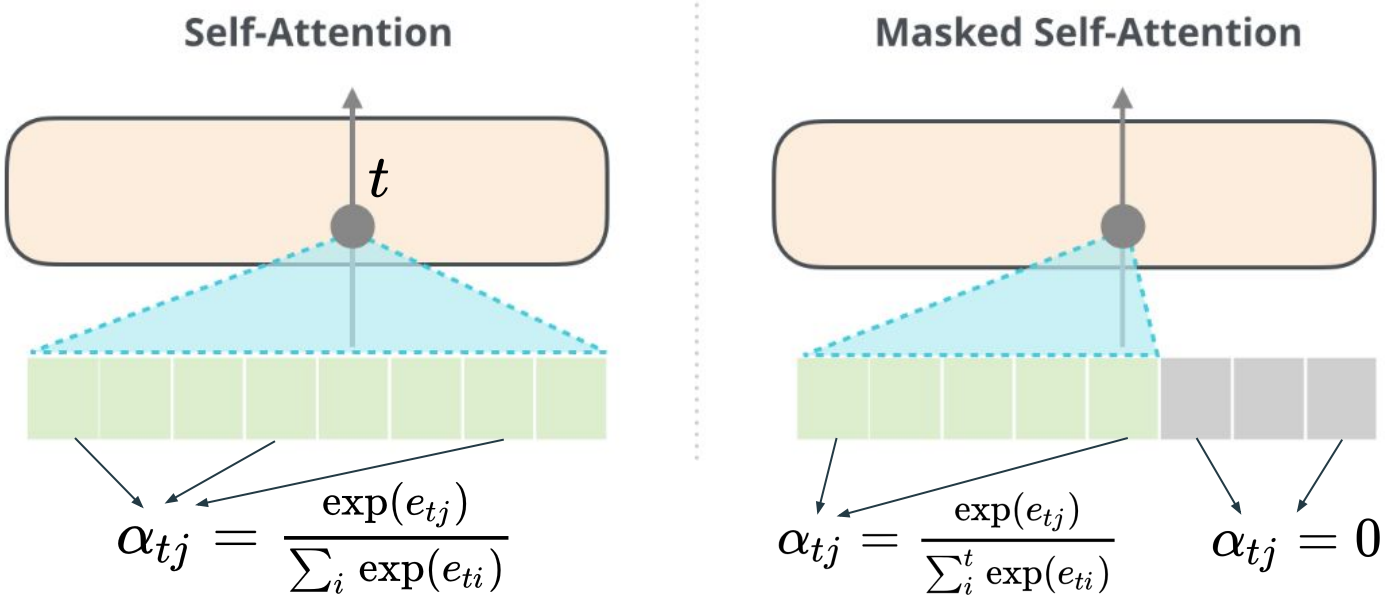
Question 2

Decoder-only models like GPT (Generative Pretrained Transformer) are designed to predict the next word in a sequence, making them well-suited for generative tasks.

1) Explain the concept of masked attention in a decoder-only model. How does it ensure that the model predicts each subsequent word based only on the words before it?

Solution 2 (1)

This is implemented by masking out future tokens in the attention scores.



Question 2

Decoder-only models like GPT (Generative Pretrained Transformer) are designed to predict the next word in a sequence, making them well-suited for generative tasks.

2) Suppose the decoder generates a partial sequence: “<START> I love”. Describe the functions used in masked self-attentions when predicting the next token (assuming a single head without position encodings).

Solution 2 (2)

- Build queries, keys and values

$$q_{love} = Q \cdot x_{love}$$

$$k_{<START>,I,love} = K \cdot x_{<START>}, K \cdot x_I, K \cdot x_{love}$$

$$v_{<START>,I,love} = V \cdot x_{<START>}, V \cdot x_I, V \cdot x_{love}$$

- Compute attention scores $n, n' \in \{< START >, I, love\}$

$$e_{love,n} = q_{love}^T \cdot k_n \quad \alpha_{love,n} = \frac{\exp(e_{love,n})}{\sum_{n'} \exp(e_{love,n'})}$$

- Compute attention output $h_{love} = \sum_n \alpha_{love,n} \times v_n$

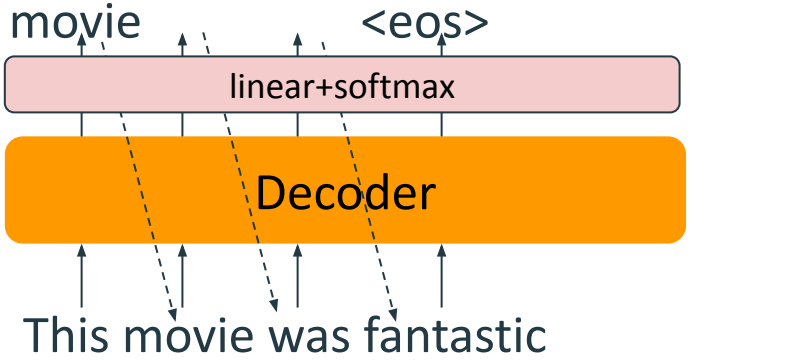
Question 2

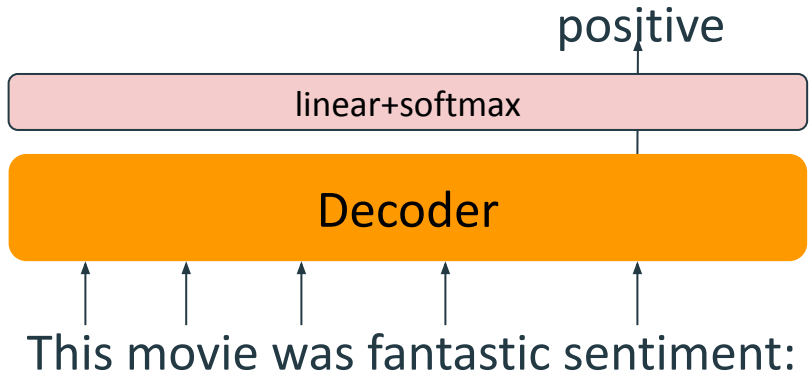
Decoder-only models like GPT (Generative Pretrained Transformer) are designed to predict the next word in a sequence, making them well-suited for generative tasks.

3) Illustrate how a decoder-only model, traditionally used for generative tasks, can be adapted for a classification task such as sentiment analysis. Discuss the modifications needed to convert the model's output from sequence generation to a classification format using this example: “This movie was fantastic.”

Solution 2 (3)

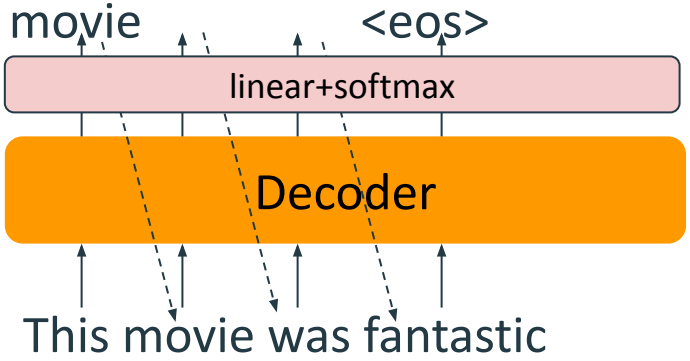
Option 1



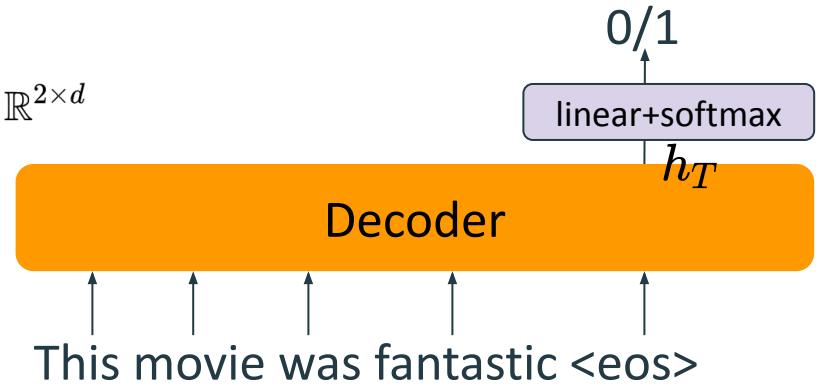


Solution 2 (3)

Option 2



$$y = softmax(Wh_T + b), W \in \mathbb{R}^{2 \times d}$$



Question 3

Prompting is a technique used to guide language models, particularly those based on the Transformer architecture, to perform specific tasks by feeding them with a carefully crafted input.

1) Discuss how prompting (or in-context learning) is different from traditional learning paradigms such as finetuning.

Solution 3 (1)

Traditional fine-tuning involves adjusting the weights of a pre-trained model across all layers on a specific task with labeled data. The model's parameters are updated to minimize the loss on this task-specific dataset. This approach requires a substantial amount of labeled data for each task the model is fine-tuned on.

Prompting, or in-context learning, on the other hand, does not alter the model's weights. Instead, it leverages the knowledge of the pre-trained model by crafting input prompts that guide the model to generate the desired output.

Question 3

Prompting is a technique used to guide language models, particularly those based on the Transformer architecture, to perform specific tasks by feeding them with a carefully crafted input.

2) Using the example of sentiment analysis, compare zero-shot and few-shot learning approaches in the context of prompting. How would the prompts differ in each case, and what are the expected outcomes?

Solution 3 (2)

In zero-shot learning, the model is provided with a task description or the input of the test example. For sentiment analysis, a zero-shot prompt might be: “Sentiment analysis: 'I loved the friendly staff and the clean rooms.' Sentiment: ”. The model is expected to generate “positive/negative”.

Few-shot learning, meanwhile, gives the model a few examples to illustrate what is expected. Each example includes a sample input (a review) and the corresponding label (sentiment).

Input: 'What a fantastic meal!' Sentiment: Positive
Input: 'Terribly rude service.' Sentiment: Negative
Input: 'I loved the friendly staff and the clean rooms.' Sentiment:

Question 3

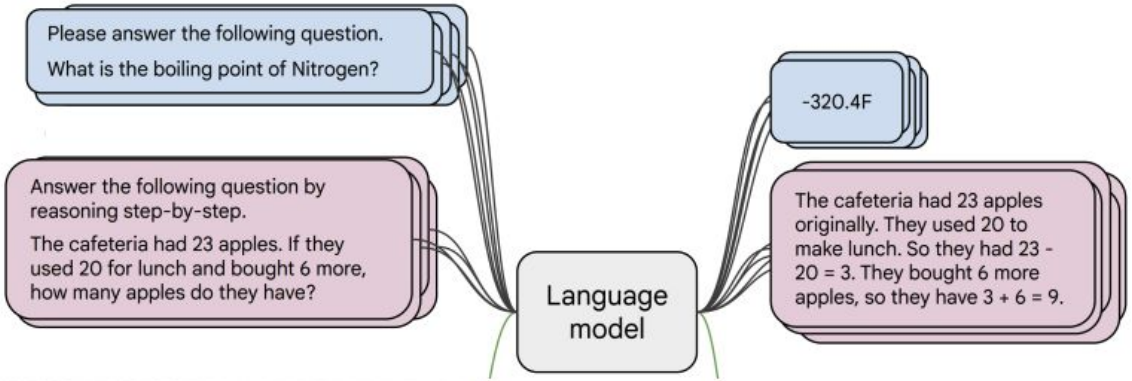
Prompting is a technique used to guide language models, particularly those based on the Transformer architecture, to perform specific tasks by feeding them with a carefully crafted input.

3) What does “instruction finetuning” refer to? Discuss how instruction finetuning is used to train and improve a language model.

Solution 3 (3)

Instruction fine-tuning refers to further training a pre-trained language generation model on a dataset where the inputs are instructions describing a wide variety of tasks, and the outputs are the desired responses or actions. This type of fine-tuning aims to improve the model's ability to follow instructions and perform tasks described in natural language.

- **Collect examples** of (instruction, output) pairs across many tasks and finetune an LM



Code

https://colab.research.google.com/drive/1L_hwnQISoIBrH7W_r83I62hJ4FBlfNsz?usp=sharing