CS6493: Natural Language Processing - Assignment 2

Instructions

1. Due at 6 pm, March 18, 2025;
2. You can submit your answers by **a single PDF with the code package** or **a single Jupiter notebook** containing both the answers and the code;
3. For the coding questions, besides the code, you are encouraged to additionally give some descriptions of your code design and its workflow. Detailed analysis of the experimental results is also preferred;
4. Total marks are 100;
5. If you have any questions, please post your questions on the Canvas-Discussion forum or contact Mr. Jilin CAO (jilincao2-c@my.cityu.edu.hk) or Mr. Weichuan WANG (weicwang2-c@my.cityu.edu.hk).

Question 1

(**20 marks**) Machine translation in natural language processing (NLP) refers to the task of automatically translating text or speech from one language to another using computational methods. It involves developing algorithms and models that can understand the meaning of the source language and generate an equivalent translation in the target language.There are several approaches to machine translation in NLP, including rule-based methods, statistical models, and neural machine translation (NMT) models.

1. The model needs to use matrices to accelerate operations, but training requires data to be input into the model in batches. The sentences in a batch need to be of equal length, so padding is required. In fact, Padding is a process of filling 0, which can be padded from the left (left padding) or right padding (right padding). The following Figure 1 is the padding process of a batch:

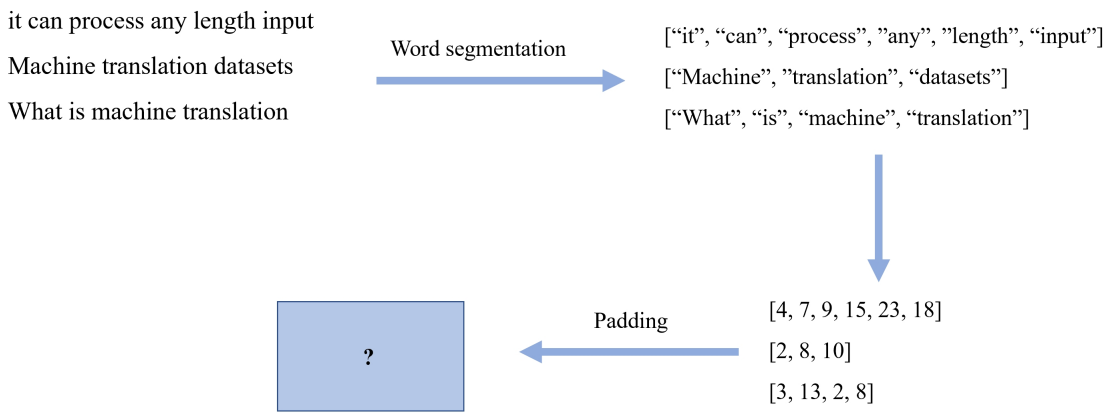

Figure 1: Padding process of a batch

   Please write down the batch after padding is completed. (**4 marks**)

2. Considering that Out Of Memory (OOM) often occurs, how to select the batch size at a small cost to ensure that no memory leak occurs during a complete epoch of model training. (**4 marks**)

3. Beam search and greedy search are two common decoding algorithms used in sequence generation tasks, such as machine translation or text generation. They are used to generate the most likely output sequence given a trained sequence-to-sequence model. Here's a possible search tree, where edges indicate probabilities of generating the next token, estimated by our language model on all the previous tokens. Please use greedy search and beam search decoding to search through this tree to find the most probable sequence. Suppose we perform a beam search with k=2. (**6 marks**)

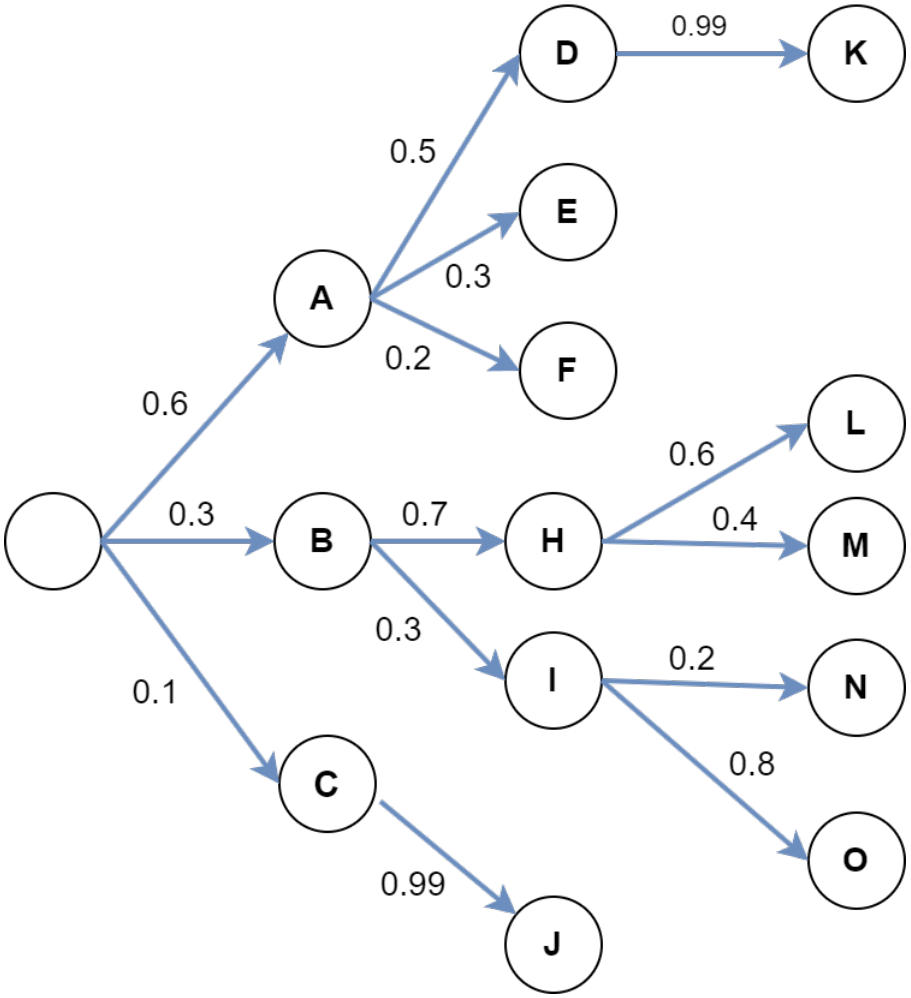

Figure 2: Search tree

4. BLEU (Bilingual Evaluation Understudy) is a popular evaluation metric used in machine translation to assess the quality of translated text. It was proposed by Papineni et al. in 2002 as a way to compare machine-generated translations with human translations.

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right). \quad (n\text{-gram precision } p_n, \text{ with weights } w_n)$$

Now, here's a set of translations:

Candidate sentence: The quick brown fox jumps over the lazy dog in the park.

Reference sentence 1: The fast brown fox jumps over the lazy dog in the park.

Reference sentence 2: The speedy brown fox jumps over the lazy dog in the park.

Please calculate the BLEU Score when N=4. Provide your calculation process. (**6 marks**)

Question 2

(20 marks) Traditional machine translation is mainly based on the Seq2Seq model, which consists of RNNs. To address the issues in Seq2Seq models, the **Attention mechanism** was introduced. Given the query matrix Q, and a set of key-value pairs $\{K, V\}$, the scaled dot-product attention is formulated as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V,$$

where d_k is the hidden size.

1. What are the limitations of the Seq2Seq model, and how does the attention mechanism address these issues? What is the role of the query (Q), key (K), and value (V) matrices in the attention mechanism? Why does the Transformer use separate weight matrices for Q and K, and why can't it use the same matrix for both? (3 marks)

2. Explain the necessity of the scaling factor $\frac{1}{\sqrt{d_k}}$ in the attention mechanism. What is the role of the softmax function in this context, and why is normalization important? (3 marks)

3. Another way to implement attention is by utilizing linear attention. Describe the linear attention mechanism and its relevance to optimizing the Transformer model for efficiency. Explain why linear attention is particularly useful for long sequences. What limitations does it still have? (3 marks)

4. Group attention is a recent optimization introduced to enhance the Transformer architecture by dividing the input into multiple groups and applying attention within each group. Write down the equation for group attention and explain it in detail. How does it improve the scalability and performance of the Transformer model? Discuss how group attention can be applied to multi-modal tasks. (6 marks)

5. Decoder-only language models have revolutionized NLP and have become increasingly influential in our everyday lives. Provide an example of a decoder-only model you like, include an architecture figure, and describe how it processes NLU and NLG tasks. For each task category, provide a specific example (e.g., Named Entity Recognition (NER) for NLU and Machine Translation (MT) for NLG). Finally, describe how you would use this model in your daily life. (For example, with DeepSeek-R1, you can try them freely on the CityU Portal under IT and Library Services[1].) (5 marks)

[1] https://www.cityu.edu.hk/portal/dashboard

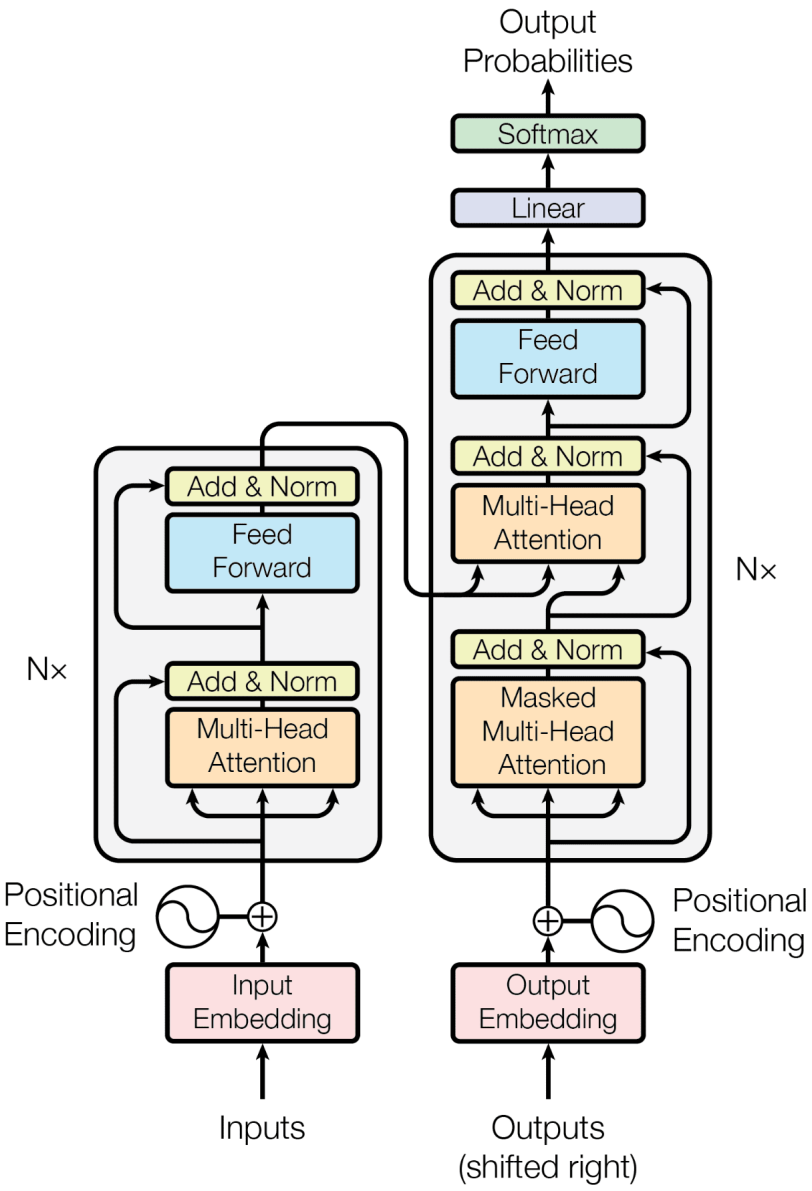

Figure 3: Overview of transformer

Question 3

(**30 marks**) We can generally divide existing pre-trained models into three categories, namely **encoder-based model**, **decoder-based model** and **encoder-decoder model** according to the model architecture. Encoder-based models, such as BERTs, employ a bidirectional Transformer encoder to predict the masked words. These models obtain all input tokens once a time and simultaneously encode them. Therefore, encoder-based models are also called **auto-encoding** models. Decoder-based models, such as GPTs, use a left-to-right Transformer to predict a text sequence word-by-word. At each stage, a token can only access to its leftward tokens. Therefore, these kinds of models are called **auto-regressive** models. Encoder-decode models, such as T5 and BART, combining the features of encoder-based models and decoder-based models, consisting of a bidirectional Transformer encoder and a unidirectional decoder.

1. There are two types of NLP tasks, namely natural language understanding (NLU) tasks, and natural language generation (NLG) tasks. What is the difference between them? (**2 marks**)

2. Please respectively answer and explain whether encoder-based/decoder-based/encoder-decoder pre-trained models are a good choice for NLU and NLG tasks. (**4 marks**)

3. The bidirectionality nature makes BERT difficult to be applied to NLG tasks. To address this problem, Dong et al., proposed a unified pre-trained language model (UNILM) which consists of a multi-layer Transformer encoder, but it can be applied to both NLG and NLU tasks. In this work, they introduced four pre-training objectives, namely unidirectional LM, bidirectional LM, sequence-to-sequence LM, and next-sentence prediction. In specific,

   • Unidirectional LM: Predict the randomly masked token only using the leftward context and itself;

   • Bidirectional LM: Predict the randomly masked token using the whole context;

   • Sequence-to-sequence LM: In this mode, we generally need to distinguish two segments, i.e., source segment and target segment. For the randomly masked tokens in the source segment, we predict them using the whole source segment. For the randomly masked token in the target segment, we predict it using the whole source segment and its leftward tokens in the target segment.

   • Next sentence prediction: It is the same as the NSP in BERT.

   These pre-training objectives can be simply implemented by different mask strategies. Given two masked segments - "Jill Birt [MASK] outdoor activities" (source segment) and "He was [MASK] hiking" (target segment), please fill in the value in the following table to show how the masks should be designed for the three language modeling pre-training objectives. (**15 marks**)

	Jill	Birt	[MASK]	outdoor	activities	He	was	[MASK]	hiking
Jill	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
Birt	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
[MASK]	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
outdoor	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
activities	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
He	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
was	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
[MASK]	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
hiking	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

Table 1: Illustration of the mask matrix. 0 for unmasked. 1 for masked.

4. Implement three functions for these masking strategies, namely mask for unidirectional lm(source seg, target seg), mask for bidirectional lm(source seg, target seg), mask for seq2seq lm(source seg, target seg). (**9 marks**)

Reference

[1] Dong, Li, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and HsiaoWuen Hon. "Unified language model pre-training for natural language understanding and generation." Advances in Neural Information Processing Systems 32 (2019).

Question 4

(**30 marks**) There are two famous and popular public datasets for QA tasks, i.e., **SQuAD v1.1** [1] and **SQuAD v2.0** [2]. A major difference between these two datasets is that the SQuAD v2.0 extends the SQuAD v1.1 problem definition by allowing for the possibility that no answer exists in the provided paragraph, making the problem more realistic. For the QA task proposed in SQuAD v1.1, we represent the input question and the passage as a single-packed sequence. Then, we introduce a start vector $S \in \mathbb{R}^H$ and an end vector $E \in \mathbb{R}^H$ during fine-tuning where H is the hidden size. The probability of word i being the start of the answer span is computed as a dot product between T_i and S followed by a softmax over all of the words in the paragraph:

$$P_i = \frac{e^{S \cdot T_i}}{\sum_j e^{S \cdot T_j}}$$

where T_i is the embedding of $i-th$ word. The analogous formula is used for the end of the answer span. The score of a candidate spans from position i to position j is defined as $S \cdot T_i + E \cdot T_j$, and the maximum scoring span where $j \geq i$ is used as a prediction.

1. For the QA task defined in SQuAD v2.0, it additionally introduces a set of questions that have no answers. However, the approach to SQuAD v1.1 must predict an answer to each question, which conflicts with the task in SQuAD v2.0. Please extend the BERT-based approach to SQuAD v2.0 to address this problem. Elaborate on your design and the workflow of your proposed approach. (**15 marks**)

2. The approaches to QA tasks, such as BiDAF and BERT, are essentially to find the boundaries of the answer spans. However, BERT, which is pre-trained with masked language modeling (MLM) and next sentence prediction (NSP), just learns token-level and sentence-level information, instead of involving any span information during the pre-training stage. Therefore, we probably can say that the pre-training objectives of the original BERT are not very good choices for QA tasks. Please design your pre-training objectives for BERT to fit in QA tasks, and other span-based tasks, like coreference resolution[3]. (**15 marks**)

Reference

[1] Rajpurkar, P., Zhang, J., Lopyrev, K. and Liang, P., 2016, November. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing.

[2] Rajpurkar, P., Jia, R. and Liang, P., 2018, July. Know What You Don't Know: Unanswerable Questions for SQuAD. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics.

[3] Lee, K., He, L., Lewis, M. and Zettlemoyer, L., 2017, September. End-to-end Neural Coreference Resolution. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing.