NLP HW3

---

1. **Describe how you approach this problem. (Classification task, question answering task, seq2seq, etc…)**

   I format this problem as sequence classification.

   - Input:
     [CLS] + tokens in d + [SEP] tokens in q, [SEP], tokens in o , and [SEP]
     - [CLS]: classifier token
     - [SEP]: sentence separator
     - d: paragraph
     - q: question
     - o: answer
   - Output:
     We get the vector whose size is 4 and its size represent one class.

2. **How do you preprocess your data? Have you encountered the problem that the input sequence is too long to fit in your model? How did you solve this?**

   I use the following method to preprocess our data.

   - Method:
     [CLS] + tokens in d + [SEP] tokens in q, [SEP], tokens in o , and [SEP]
     - [CLS]: classifier token
     - [SEP]: sentence separator
     - d: paragraph
     - q: question
     - o: answer

   I encountered the following problems.

   - Input sequence too long:
     We truncate sequence if its sequence longer than 512
   - Input sequence have different length:
     We pad the sequence to make each sequence have equal size.

3. **Which model(s) have you tried? Did you use models that have finetuned to some NLP tasks? What's their difference in performance?**

   I have tried the following model:

   - BertForSequenceClassification:
     I utilize this model when I format this task as classification
   - BertForQuestionAnswer:
     I utilize this model when I format this task as QuestionAnswer
   - BertForMultipleChoice:
     I utilize this model when I format this task as Multiple choice.

   I also tried different model which is finetuned on difference NLP tasks:

- BERT-wwm-ext、BERT-wwm
  They finetuned Bert model on a whole word masking task.

I find that I can get better. I format this task as a classification task. I find that the model which is finetune on a whole word masking task can get the better result. The final model we use is “BERT-wwm-exttForSequenceClassification”. I think the reason why finetune on a whole word masking task can get better performance is that in traditional masked language models like BERT, a random subset of subword tokens within a word is masked during training. However, for certain applications or languages with a significant presence of out-of-vocabulary words, masking entire words can be beneficial.

4. **Have you encountered any other difficulties (besides Q2) while doing this task? How did you solve them? If none, what do you think is the hardest part in this task? (preprocessing / model implementation / model selection …)**

I think the hardest part in this task is preprocessing. In the preprocessing stage, there are several tasks to handle before deciding on the problem formulation. One of the initial considerations is determining the appropriate problem formulation for addressing the specific issue at hand. Different problem formulations necessitate distinct preprocessing steps. After fixing problem formulation of this task, we need to think how to transform the data format to the format we want. Transforming data into different formats can have a significant impact on the final performance and training time of a machine learning model.
For example:
In the context of the same classification task, I have tried with different input formats.

- Initial:
  - Input:
    [CLS] + tokens in d + tokens in q tokens in o (option A) [SEP]
    [CLS] + tokens in d + tokens in q tokens in o (option B) [SEP]
    [CLS] + tokens in d + tokens in q tokens in o (option C) [SEP]
    [CLS] + tokens in d + tokens in q tokens in o (option D) [SEP]
  - Output:
    The probability of each option.
  - Operation:
    Choose the one with the maxim probability as our answer.
- Final:
  - Input:
    [CLS] + tokens in d + [SEP] tokens in q, [SEP], tokens in o , and [SEP]
  - Output:
    The probability of each class.
  - Operation:

Choose the one with the maxim probability as its class.

After transitioning to a different input format, both training time and model performance have shown significant improvements. I believe the improved performance can be attributed to the fact that, initially, a significant portion of the text among the four options was the same. This similarity might have posed a challenge for the model to discern the correct choice.