04

实战演练之多项选择

（1）多项选择任务介绍

（2）基于Transfromers的解决方案

（3）代码实战演练

多项选择任务介绍

多项选择简介

- **什么是多项选择任务**
  - 多项选择（Multiple Choice）任务是机器阅读理解任务中的一种形式，相较之前的机器阅读理解定义而言，多项选择任务还需要提供答案的候选项，即**给定一个或者多个文档P，以及一个问题Q和对应的多个答案候选，输出问题Q的答案A，A是答案候选中的某一个选项。**
  - 当然，更广泛的来看，多项选择也不局限于一定是阅读理解，也可以理解为是一种匹配，关于文本匹配的概念，将在下次课程进行介绍

多项选择任务介绍

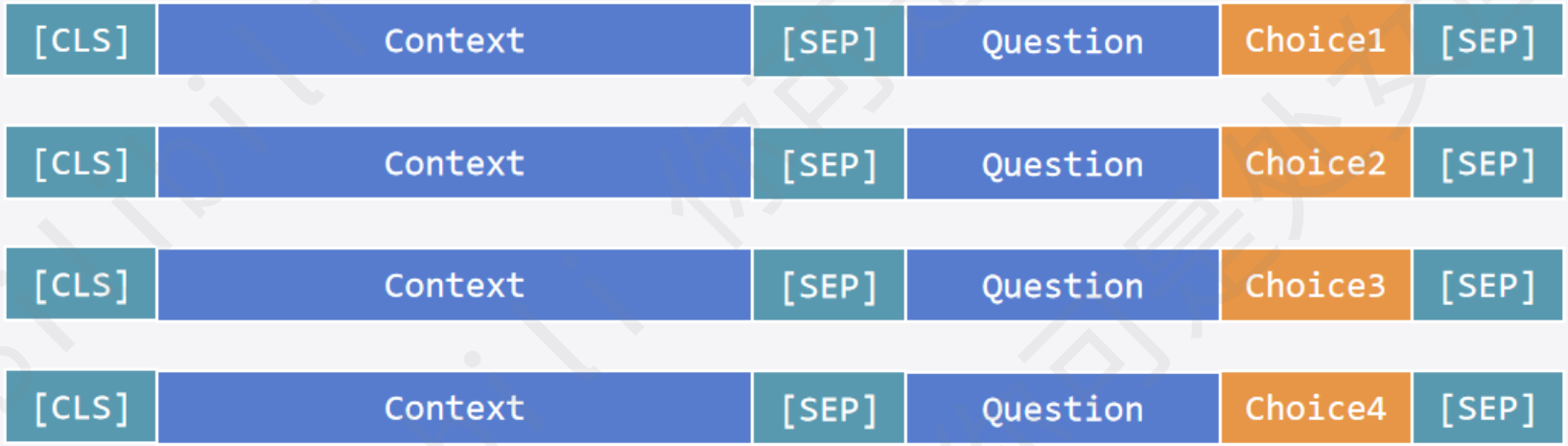
多项选择简介

- **多项选择任务示例**
  - Context：老师把一个大玻璃瓶子带到学校，瓶子里装着满满的石头、玻璃碎片和沙子。之后，老师请学生把瓶子里的东西都倒出来，然后再装进去，先从沙子开始。每个学生都试了试，最后都发现没有足够的空间装所有的石头。老师指导学生重新装这个瓶子。这次，先从石头开始，最后再装沙子。石头装进去后，沙子就沉积在石头的周围，最后，所有东西都装进瓶子里了。老师说：“如果我们先从小的东西开始，把小东西装进去之后，大的石头就放不进去了。生活也是如此，如果你的生活先被不重要的事挤满了，那你就无法再装进更大、更重要的事了。”
  - Question：正确的装法是，先装?
  - Choices / Candidates：["小东西", "大东西", "轻的东西", "软的东西"]
  - Answer：大东西

本质上就是一项分类任务

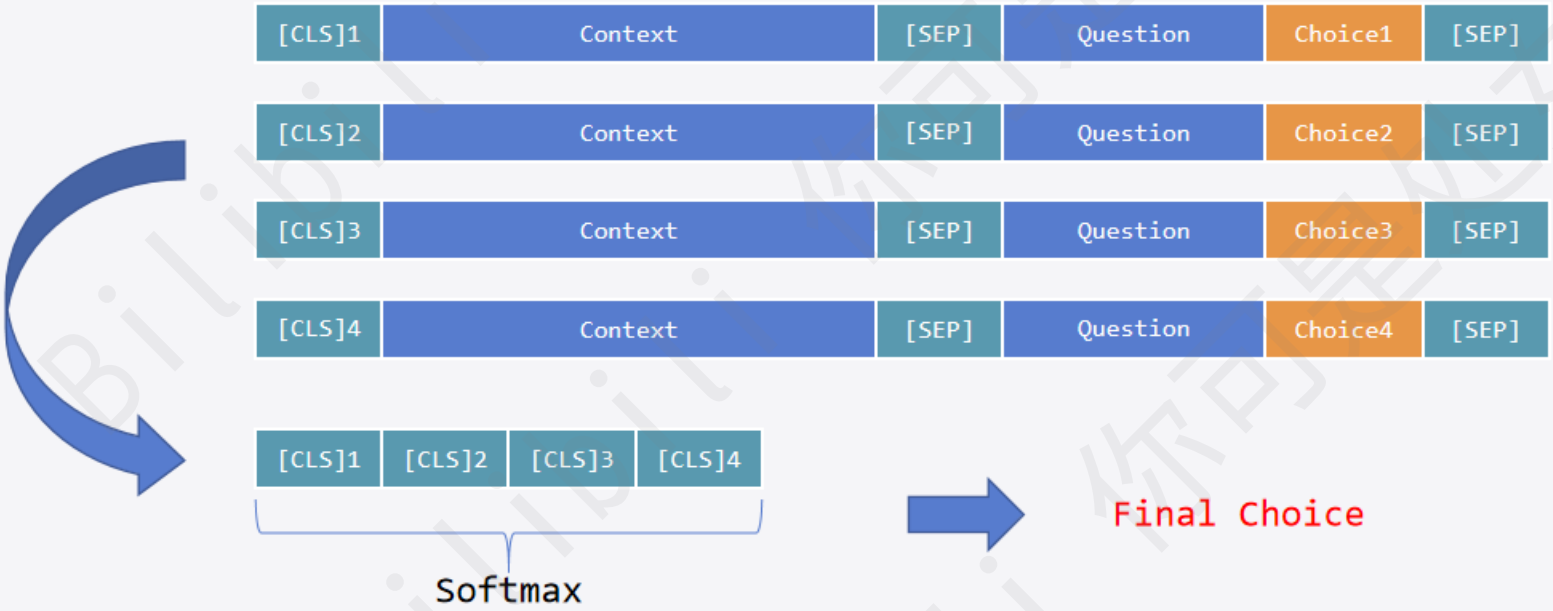
基于Transformers的解决方案

- **数据预处理**
  - 数据处理格式



基于Transformers的解决方案

- 训练与推理
  - 基本原理



基于Transformers的解决方案

- **模型结构**
  - *ModelForMultipleChoice

```python
class BertForMultipleChoice(BertPreTrainedModel):
    def __init__(self, config):
        super().__init__(config)

        self.bert = BertModel(config)
        classifier_dropout = (
            config.classifier_dropout if config.classifier_dropout is not None else config.hidden_dropout_prob
        )
        self.dropout = nn.Dropout(classifier_dropout)
        self.classifier = nn.Linear(config.hidden_size, 1)

        # Initialize weights and apply final processing
        self.post_init()
```

基于Transformers的解决方案

- **模型结构**
  - *ModelForMultipleChoice

```
num_choices = input_ids.shape[1] if input_ids is not None else inputs_embeds.shape[1]

# [batch * num_choices, seq_len]
input_ids = input_ids.view(-1, input_ids.size(-1)) if input_ids is not None else None
attention_mask = attention_mask.view(-1, attention_mask.size(-1)) if attention_mask is not None else None
token_type_ids = token_type_ids.view(-1, token_type_ids.size(-1)) if token_type_ids is not None else None
position_ids = position_ids.view(-1, position_ids.size(-1)) if position_ids is not None else None
inputs_embeds = (
    inputs_embeds.view(-1, inputs_embeds.size(-2), inputs_embeds.size(-1))
    if inputs_embeds is not None
    else None
)
```

基于Transformers的解决方案

- **模型结构**
  - *ModelForMultipleChoice

```
outputs = self.bert(
    input_ids,
    attention_mask=attention_mask,
    token_type_ids=token_type_ids,
    position_ids=position_ids,
    head_mask=head_mask,
    inputs_embeds=inputs_embeds,
    output_attentions=output_attentions,
    output_hidden_states=output_hidden_states,
    return_dict=return_dict,
)

pooled_output = outputs[1]  # [batch * num_choices, hidden_dim]

pooled_output = self.dropout(pooled_output) # [batch * num_choices, hidden_dim]
logits = self.classifier(pooled_output)     # [batch * num_choices, 1]
reshaped_logits = logits.view(-1, num_choices)  # [batch, num_choices]
```

代码实战演练

- **数据集**
  - C3
- **预训练模型**
  - hfl/chinese-macbert-base