02

实战演练之命名实体识别

（1）命名实体识别任务介绍

（2）基于Transfromers的解决方案

（3）代码实战演练

命名实体识别任务介绍

命名实体识别简介

- **什么是命名实体识别任务**
  - 命名实体识别（Named Entity Recognition，简称NER）是指识别文本中具有特定意义的实体，主要包括人名、地名、机构名、专有名词等。通常包括两部分：（1）实体边界识别；（2）确定实体类别（人名、地名、机构名或其他）。
  - 例：

“小明在北京上班”

实体类别	实体
地点	北京
人物	小明

命名实体识别任务介绍

命名实体识别简介

- **数据标注体系**
  - IOB1、IOB2、IOE1、IOE2、IOBES、BILOU
- **IOB2标注**
  - I表示实体内部，O表示实体外部，B表示实体开始
  - B/I-XXX，XXX表示具体的类别

标记	说明
B-Person	人名开始
I- Person	人名中间
B-Organization	组织名开始
I-Organization	组织名中间
O	非命名实体

- **IOBES标注**
  - I表示实体内部，O表示实体外部，B表示实体开始，E表示实体结束，S表示一个词单独形成一个命名实体
  - 有时也会使用M代替I，但本质是同一含义

命名实体识别任务介绍

命名实体识别简介

- **评估指标**
  - Precision、Recall、F1
- **简单示例**

Sen:	The	Hospital	said	it	would	probably	know	by	Tuesday	whether	its	patients
Gold:	**b-AGENT**	**e-AGENT**	o	o	**b-DSE**	**m-DSE**	**e-DSE**	o	o	**b-TARGET**	**m-TARGET**	**m-TARGET**
Predict:	o	e-AGENT	b-DSE	o	b-DSE	m-DSE	e-DSE	o	o	b-AGENT	b-DSE	b-TARGET

Sen:	had	Congo	Fever	.
Gold:	**m-TARGET**	**m-TARGET**	**e-TARGET**	o
Predict:	m-TARGET	e-TARGET	o	o

predict_num=2
gold_num=3
correct_num=1

$P=\frac{1}{2}$ $R=\frac{1}{3}$ $F1=\frac{2*\frac{1}{2}*\frac{1}{3}}{\frac{1}{2}+\frac{1}{3}}$

基于Transformers的解决方案

- **模型结构**
  - *ModelForTokenClassification

```python
class BertForTokenClassification(BertPreTrainedModel):
    _keys_to_ignore_on_load_unexpected = [r"pooler"]

    def __init__(self, config):
        super().__init__(config)
        self.num_labels = config.num_labels

        self.bert = BertModel(config, add_pooling_layer=False)
        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, config.num_labels)

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

基于Transformers的解决方案

- **模型结构**
  - *ModelForTokenClassification

```
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,
)

sequence_output = outputs[0]

sequence_output = self.dropout(sequence_output)
logits = self.classifier(sequence_output)

loss = None
if labels is not None:
    loss_fct = CrossEntropyLoss()
    loss = loss_fct(logits.view(-1, self.num_labels), labels.view(-1))
```

基于Transformers的解决方案

- **评估函数**
  - seqeval
  - 需要额外安装 seqeval
    - pip install seqeval
    - 安装过程中报错：Microsoft Visual C++ 14.0 or greater is required. Get it with "Microsoft C++ Build Tools
    - 进入https://my.visualstudio.com，下载C++ build tools，安装
  - evaluate.load（“seqeval”）

代码实战演练

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