人工智能与数据挖掘课程设计

命名实体识别

**组员信息：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 姓名 | 学号 | 成员贡献 | 实践成绩 |
| 1 | 郭嘉琪 | 2021522001 | 代码运行环境布置、报告编写 |  |
| 2 | 王雅曈 | 2021522005 | 代码编写、报告编写 |  |
| 3 | 赵欣然 | 2021522013 | 代码编写、报告编写 |  |
| 4 |  |  |  |  |

# ****一、引言****

**1、问题描述：**

命名实体：特定类型事物的名称和符号，一般是一个单词或名词短语。

命名实体识别（NER）:就是识别出文本中的命名实体，并将实体分类为给定的类别中。2、**任务分析：**

基于百度的一个深度学习框架paddlepaddle,该框架的生态环境中，有一个paddleNLP开源子项目，该项目提供了当前大部分NLP预训练模型训练好的模型参数，我们使用paddleNLP提供的预训练模型来做NER命名实体识别任务。

# ****实验数据****

# 1、数据获取或预处理

**（1）数据获取：**

本次数据是在清华大学开源的文本分类数据集THUCTC基础上，选出部分数据进行细粒度命名实体标注，原数据来源于Sina News RSS。

训练数据：文件模型训练数据集采用CLUENER2020-master数据集。

链接：[CLUEbenchmark/CLUENER2020: CLUENER2020 中文细粒度命名实体识别 Fine Grained Named Entity Recognition (github.com)](https://github.com/CLUEbenchmark/CLUENER2020?tab=readme-ov-file)

1. **预处理：**
   1. 配置环境

采用conda方式配置环境：

<https://www.paddlepaddle.org.cn/install/quick?docurl=/documentation/docs/zh/install/docker/linux-docker.html>

主要依赖包paddlepaddle-gpu==2.5.2、paddlenlp==2.6.1

首先安装 Anacodna，再安装相应第三方库bash，进入anaconda的python环境。

创建虚拟环境，在虚拟环境中安装paddlepaddle，根据情况选择cpu或gpu的版本。

* 1. 加载预训练模型

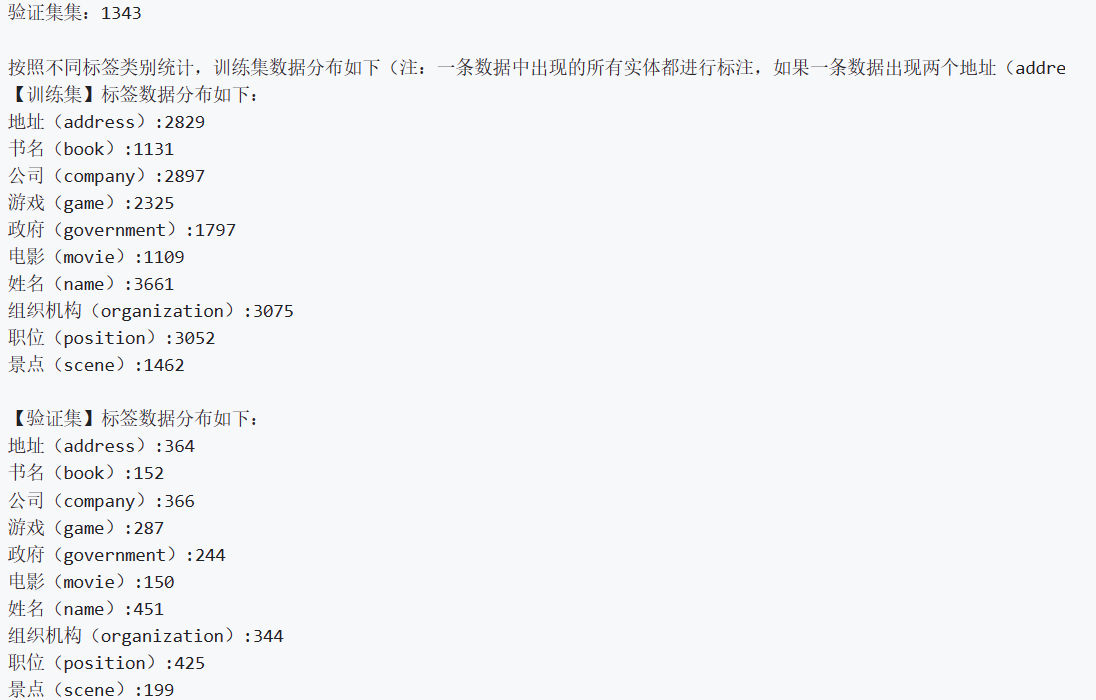
PaddleNLP的预训练模型可以很容易地通过from\_pretrained()方法加载

* 1. 生成数据集和构建模型

PaddleNLP的load\_dataset()方法会从paddle.datasets中找到msra\_ner数据集对应的数据读取脚本，并调用脚本中的DatasetBuilder类的相关方法生成数据集，以默认的MapDataset的类型返回。

本次作业使用的cluener数据集，使用load\_dataset()方法中提供的data\_files参数实现某些数据格式与内置数据集相同的本地数据替换某些内置数据集的数据，也可以用这个方法在本地创建数据集。由于只有Paddle内置的数据集才具有将数据中的label自动转为id的功能（如DatasetBuilder），所以还需要自定义中添加label转id的功能。此外，我们还采用了\_\_getitem\_\_()方法和\_\_len\_\_()方法的python对象创建MapDataset。

## 2.2数据统计



# ****方法/模型****

方法基于paddlepaddle深度学习架构，采用bert-wwm-chinese预训练模型，结合交叉熵损失函数，实现命名实体标签预测。

1. **方法：**

Cluener数据集中信息是一整个短语，所以我们采取的序列标注方法是BIO标注。B为名词短语的开头，I为名词短语的中间，O不属于名词短语。

cluener\_label\_list = ['B-address', 'I-address', 'B-book', 'I-book', 'B-company', 'I-company', 'B-game', 'I-game', 'B-government', 'I-government', 'B-movie', 'I-movie', 'B-name', 'I-name', 'B-organization', 'I-organization', 'B-position', 'I-position', 'B-scene', 'I-scene', 'O']

msra\_label\_list = ['B-PER', 'I-PER', 'B-ORG', 'I-ORG', 'B-LOC', 'I-LOC', 'O']

peoples\_daily\_label\_list = ['B-PER', 'I-PER', 'B-ORG', 'I-ORG', 'B-LOC', 'I-LOC', 'O']

针对cluener中文数据集，我们采用了基于语言模型词向量的动态表征方法：Goole的BERT（预训练语言模型的应用）。

1. **模型：**

Word-level embedding（词嵌入）是自然语言处理（NLP）中语言模型与表征学习技术的统称，NLP中的早期预训练技术。

选择了BERT中基于全词掩码（wwm）中文的预训练模型：bert-wwm-chinese。

我们使用BertForTokenClassification来制作NER的模型。通过from\_pretrianed方法载入tokenizer得到input\_ids（输入的id）。

# ****结果评测/展示****

cluener                     #cluener数据集

│   ├── README.md

│   ├── cluener\_predict.json

│   ├── dev.json

│   ├── test.json

│   └── train.json

├── main.py                        #训练代码

└── src

    ├── \_\_init\_\_.py

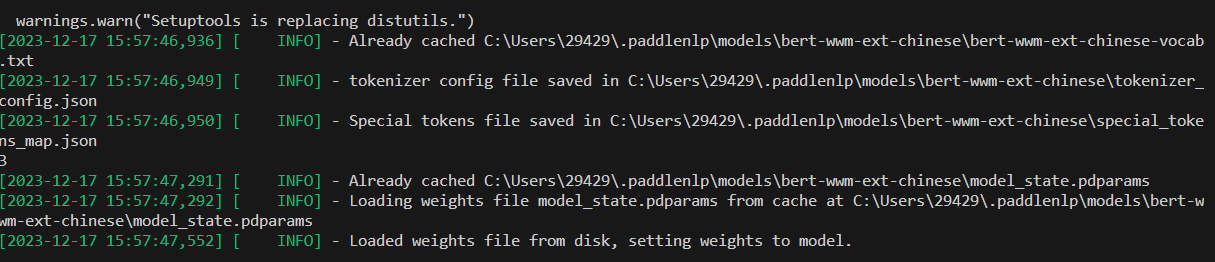
    ├── cluener\_dataset.py        #cluener数据集处理代码

    ├── dataset.py                #数据集相应的处理代码

    ├── eval.py                   #模型评估代码

    ├── parameter.py              #各种超参数的设置，包括训练的各种参数，数据集，预训练模型等

    └── predict.py                #预测代码





****五、总结****

通过这次的实践我们不仅能够了解到命名实体识别的基本概念和原理，还能通过实践将课堂知识发挥作用，并且学习到如何使用各种工具和技术来解决实际问题。深入了解NER算法，在开始使用Github上的NER项目之前，建议先学习NER算法的基本原理和常用模型，这样能够更好地理解实现细节并进行必要的调整。同时，实践项目可以让我们更好地理解理论知识，并且提升了我们运用编程技能和解决实际问题的能力。

尽管命名实体识别在许多领域中发挥着重要作用，但它仍然存在一些局限性。

关于领域限制：命名实体识别往往只能在有限的领域和有限的上下文环境中工作。

关于开放性问题：命名实体的开放性是指命名实体内容和类型并非永久不变，会随着时间变化发生变化。

关于复杂性问题：命名实体识别是未登录词中数量最多、识别难度最大、对分词效果影响最大的问题，同时它也是信息抽取、信息检索、机器翻译、问答系统等多种自然语言处理技术必不可少的一部分。

关于依赖外部资源：命名实体识别过程常常要与中文分词、浅层语法分析等过程相结合，分词、语法分析系统的可靠性也直接决定命名实体识别的有效性，使得中文命名实体识别更加困难。

关于训练数据的局限性：命名实体识别模型的训练数据往往来自特定的领域或主题，如果训练数据不足或者不涵盖所有可能的实体类型，那么模型的泛化能力就会受到影响。