Bert中文文本分类

1 实验描述

项目来源：https://github.com/649453932/Bert-Chinese-Text-Classification-Pytorch

任务：利用Bert模型完成中文文本分类。类别：财经、房产、股票、教育、科技、社会、时政、体育、游戏、娱乐。

数据集：从THUCNews中抽取了20万条新闻标题，文本长度在20到30之间。一共10个类别，每类2万条。数据以字为单位输入模型。

数据集划分：

数据集	数据量
训练集	18万
验证集	1万
测试集	1万

2 模型介绍

BERT（Bidirectional Encoder Representations from Transformers）是由Google提出的一种预训练语言模型，首次实现了在多个自然语言处理任务上的新性能突破。BERT模型的核心思想是通过双向Transformer架构捕获文本的上下文双向依赖关系，从而生成丰富的词向量表示。

BERT的核心论文是 *"BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding"*，发表在2018年，作者为Jacob Devlin等。

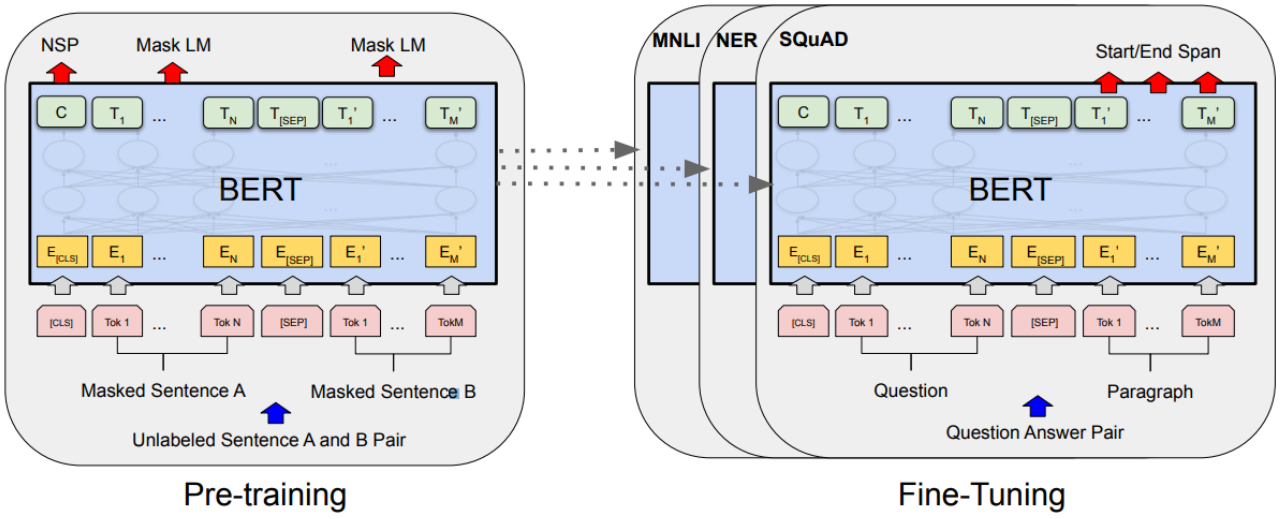

Figure 1: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are fine-tuned. `[CLS]` is a special symbol added in front of every input example, and `[SEP]` is a special separator token (e.g. separating questions/answers).

2.1 BERT模型结构

BERT的主要组成部分是多层堆叠的Transformer编码器，其结构可以概述为：

- 输入层：输入是一个由词嵌入、位置嵌入和分段嵌入组成的向量序列。
- Transformer编码器

  ：采用标准的多头自注意力机制和前馈神经网络。BERT通常提供两种版本：

  - BERT-Base：12层Transformer，隐藏层大小为768，注意力头数为12，总参数量为1.1亿。
  - BERT-Large：24层Transformer，隐藏层大小为1024，注意力头数为16，总参数量为3.4亿。

- 输出层：在预训练任务中输出特定任务所需的向量表示。

2. 2 BERT的预训练与微调

2.2.1 第一阶段：预训练（基础训练）

> *核心目标：学会通用的语言规则*

任务本质：BERT在预训练阶段并不知道它最终会用来干什么（分类、问答等任务）。它唯一的目标是：在不依赖具体任务的情况下，理解语言的规律和上下文关系。

1.1 Masked Language Model (MLM)

- 目标：随机掩码输入句子中15%的单词，让模型预测这些被掩码的词。
- 过程
    a. 随机选择15%的词：
        - 80%的概率将其替换为特殊标记 `[MASK]`。
        - 10%的概率随机替换为其他词。
        - 10%的概率保持原词。
    b. 模型根据上下文预测被掩码的词。
- 举例：给模型一段句子，比如：“我今天吃了一个[MASK]，感觉很好吃。”，模型的任务是根据句子上下文预测 `[MASK]` 这个位置可能是什么词，比如“苹果”。
- 意义：MLM训练让模型能够学习双向上下文信息，克服传统单向语言模型的限制（如GPT）。

1.2 Next Sentence Prediction (NSP)

- 目标：判断两个句子是否具有逻辑上的连续关系。
- 过程
    a. 输入由两个句子对构成：句子A和句子B。
    b. 50%的情况下，句子B是句子A的真实连续句；50%的情况下，句子B是从语料库中随机抽取的句子。

c. 模型通过输出层预测这两个句子是否连续。

- 举例：给模型两句话，比如“我今天吃了一个苹果。”和“这是一个晴朗的早晨。”，模型需要判断这两句话是不是连贯的，或者是否是随机拼接的。
- 意义：NSP任务帮助模型学习句子间的语义关系，是对许多自然语言理解任务（如问答系统、文本分类）的重要补充。

预训练完成后，BERT会输出每个单词的上下文嵌入向量，这些向量包含了丰富的语义信息，可以理解为模型已经“懂得语言了”。

2.2.2 第二阶段：微调（专项训练）

核心目标：针对特定任务训练语言技能

任务本质：微调是让模型专注于某个具体的任务，比如情感分类、问答、翻译等。我们不需要从头开始训练模型，而是利用BERT已经掌握的语言知识快速适配特定任务。

2.1 任务适配机制

- **输入保持一致**：无论是预训练还是微调，BERT的输入格式（Token Embeddings、Segment Embeddings、Position Embeddings）相同。
- 输出调整：将BERT的输出与特定任务的目标结合。例如：
    - **分类任务**：使用 `[CLS]` 标记的输出作为句子级表示，接一个全连接层进行分类。
    - **序列标注任务**：直接对每个词的输出向量进行标注。
    - **问答任务**：用两个标记（`start` 和 `end`）预测答案的起始和结束位置。

2.2 微调过程

1. **初始化参数**：将BERT预训练的权重作为初始参数。
2. **添加任务特定层**：如分类头或序列标注层。
3. **训练**：在小规模标注数据集上通过梯度下降更新模型参数，BERT的所有层均可微调。

2.3 举例

（以本次实验中的文本分类任务为例）

1. **输入准备**：
    - 将具体任务的数据转化为BERT需要的输入格式。
    - 比如，在文本分类任务中，把句子转化为带有 `[CLS]` 和 `[SEP]` 特殊标记的序列，BERT会用 `[CLS]` 的输出表示整个句子的特征。
2. **添加任务特定层**：
    - 根据任务要求，往BERT的输出后面加一个简单的任务特定层，对于分类任务，加一个全连接层输出类别。
3. **训练模型**：
    - 使用下游任务的数据，通过反向传播调整模型参数。
    - 微调的重点是让BERT的语言表示和特定任务的目标对齐。

经过微调后，BERT不仅理解语言，还能根据你的要求解决具体问题。例如，给它一句话，它可以判断对应的类别。

3 实验环境

python 3.7
pytorch 1.1
tqdm
sklearn
tensorboardX

4 运行说明

（1）下载Bert预训练模型

预训练模型下载地址：

1、`bert_pretrain`模型：

bert_Chinese: 模型 https://s3.amazonaws.com/models.huggingface.co/bert/bert-base-chinese.tar.gz
词表 https://s3.amazonaws.com/models.huggingface.co/bert/bert-base-chinese-vocab.txt
模型来源：这里
备用：模型的网盘地址：https://pan.baidu.com/s/1qSAD5gwClq7xlgzl_4W3Pw

将下面的三个文件放到`bert_pretrain`文件夹下：

pytorch_model.bin
bert_config.json
vocab.txt

2、ERNIE预训练模型：

ERNIE_Chinese: http://image.nghuyong.top/ERNIE.zip
来源：这里
备用：网盘地址：https://pan.baidu.com/s/1lEPdDN1-YQJmKEd_g9rLgw

将下面的三个文件放到`ERNIE_pretrain`文件夹下：

pytorch_model.bin
bert_config.json
vocab.txt

（2）运行（训练+预测）

```
# 训练并测试：
# bert
python run.py --model bert

# bert + 其它
python run.py --model bert_CNN

# ERNIE
python run.py --model ERNIE
```

5 效果

模型	ACC	备注
bert	94.83%	bert
ERNIE	94.61%	-
bert_CNN	94.44%	bert + CNN
bert_RNN	94.57%	bert + RNN
bert_RCNN	94.51%	bert + RCNN
bert_DPCNN	94.47%	bert + DPCNN

原始的bert效果就已经很好了。

6 拓展任务

以下拓展任务根据自己的能力和时间选择性完成即可。

- 更换其他数据集，使用Bert模型进行文本分类任务。（提示：使用新的数据集可以通过修改代码来使用其他数据集，也可以直接按照本实验中数据集的格式来格式化其他的数据集。其他的数据集文本分类任务也有如情感分析等，可以自主选择。）
- 微调Bert模型完成其他预训练任务。（如命名实体识别 NER等）