# 实验一：中文情感分析

20123101 李昀哲

一、**实验题目**

基于PaddleNLP利用ERNIE 3.0预训练模型微调并进行中文情感分析预测。

二、**实验内容**

首先阐述什么是情感分析任务，再使用预训练的ERNIE 3.0模型进行调优，最后进行训练并利用模型进行预测。

1. 情感分析任务

情感分析是一种自然语言处理 (NLP) 技术，用于确定数据情感是正面的、负面的还是中性的。简而言之，说一句话，判断其情感，正向、负向还是中性。

一般应用为：帮助企业监控客户反馈中的品牌和产品情感，了解客户需求；有助于企业分析商业伙伴们的态度，以便更好地进行商业决策。

1. ERNIE 3.0 模型

近一年来，以GPT-3、Switch-Transformer为代表的大规模预训练模型，带来了人工智能领域新的突破，由于其强大的通用性和卓越的迁移能力，掀起了预训练模型往大规模参数化发展的浪潮。然而，现有的大规模预训练模型，主要依赖纯文本学习，缺乏大规模知识指导学习，模型能力存在局限。

ERNIE 3.0的研究者进一步挖掘大规模预训练模型的潜力，基于深度学习平台飞桨的分布式训练技术优势，首次在百亿级预训练模型中引入大规模知识图谱，提出了海量无监督文本与大规模知识图谱的平行预训练方法（Universal Knowledge-Text Prediction）。

通过将大规模知识图谱的实体关系与大规模文本数据同时输入到预训练模型中进行联合掩码训练，促进了结构化知识和无结构文本之间的信息共享，大幅提升了模型对于知识的记忆和推理能力，如图1所示。

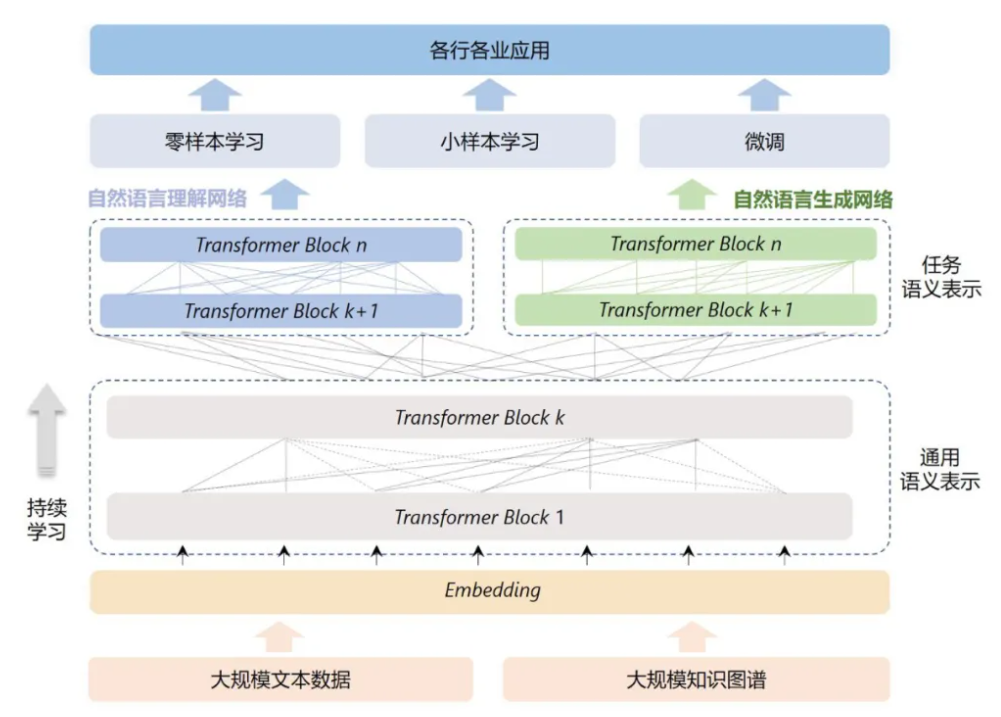


图1 ERNIE3.0模型示意图

1. 实验流程
2. 环境准备

AI Studio平台默认安装了Paddle和PaddleNLP。需要使用前upgrade

1. 加载数据集

ChnSentiCorp数据集包含酒店、笔记本电脑和书籍的网购评论。

数据集示例：

其中1表示正向情感，0表示负向情感.

|  |  |  |
| --- | --- | --- |
| **Qid** | **Label** | **Text** |
| 0 | 1 | <荐书> 推荐所有喜欢<红楼>的红迷们一定要收藏这本书,要知道当年我听说这本书的时候花很长时间去图书馆找和借都没能如愿,所以这次一看到当当有,马上买了,红迷们也要记得备货哦! |
| 1 | 0 | 商品的不足暂时还没发现，京东的订单处理速度实在.......周二就打包完成，周五才发货... |

1. 加载中文ERNIE3.0预训练模型和分词器

PaddleNLP中Auto模块（包括AutoModel, AutoTokenizer及各种下游任务类）提供了方便易用的接口，无需指定模型类别，即可调用不同网络结构的预训练模型。PaddleNLP的预训练模型可以很容易地通过from\_pretrained()方法加载，Transformer预训练模型包含了40多个主流预训练模型，500多个模型权重。

AutoModelForSequenceClassification可用于句子级情感分析和目标级情感分析任务，通过预训练模型获取输入文本的表示，之后将文本表示进行分类。PaddleNLP已经实现了ERNIE 3.0预训练模型，可以通过一行代码实现ERNIE 3.0预训练模型和分词器的加载。

1. 基于预训练模型进行数据分析

Dataset中通常为原始数据，需要经过一定的数据处理并进行采样组batch。通过Dataset的map函数，使用分词器将数据集从原始文本处理成模型的输入。定义paddle.io.BatchSampler和collate\_fn构建 paddle.io.DataLoader。

实际训练中，根据显存大小调整批大小batch\_size和文本最大长度max\_seq\_length。

1. 数据训练和评估

定义训练所需的优化器、损失函数、评价指标等，就可以开始进行预模型微调任务。

|  |
| --- |
| # Adam优化器、交叉熵损失函数、accuracy评价指标  optimizer = paddle.optimizer.AdamW(learning\_rate=2e-5, parameters=model.parameters())  criterion = paddle.nn.loss.CrossEntropyLoss()  metric = paddle.metric.Accuracy() |

迭代100次就评估当前训练的模型，将最优参数和分词器词表保存

|  |
| --- |
| # 每迭代100次，评估当前训练的模型、保存当前模型参数和分词器的词表等  if global\_step % 100 == 0:       save\_dir = ckpt\_dir       if not os.path.exists(save\_dir):           os.makedirs(save\_dir)       print(global\_step, end=' ')       acc\_eval = evaluate(model, criterion, metric, dev\_data\_loader)       if acc\_eval > best\_acc:           best\_acc = acc\_eval           best\_step = global\_step           model.save\_pretrained(save\_dir)           tokenizer.save\_pretrained(save\_dir) |

1. **实验结果及分析**

1.实验结果

通过迭代次数的变化，训练模型的超参数和分词器词表会呈现不同的效果，因此需要在迭代中得到最佳参数和最优分词表，保存至特定文件夹中。不同迭代次数下的模型的准确度如图3所示。

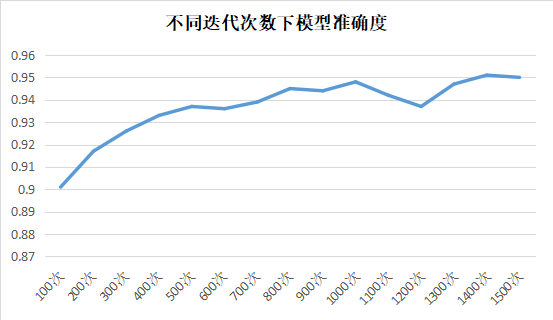


图3 不同迭代次数下模型准确度

测试集的结果如图4所示，第二列为测试文本，第三列为预测结果，从人类角度来看，预测的结果基本正确。



图2 模型预测结果

1. Epoch的选择分析

Epoch训练轮次，定义了学习算法在整个训练数据集中的工作次数。一个Epoch意味着训练数据集中的每个样本都有机会更新内部模型参数。

Epoch由一个或多个Batch组成，具有一批的Epoch称为批量梯度下降学习算法，当一个完整的数据集通过神经网络一次并且返回了一次，这个过程称为一次Epoch。

当一个Epoch对于计算机而言太过庞大的时候，需要将其拆分成多个小块；

Epoch增加一次，神经网络的权重更新一次，随着Epoch数量增加，曲线会从训练集合中的欠拟合变为过拟合。