自然语言通顺与否判定

任务描述

本次任务为自然语言通顺与否的判定，即给定一个句子，要求判定所给的句子是否通顺。

问题分析

首先判断语句通顺是一个二分类任务，可以使用某些分类器实现。针对任务特点和任务场景的分析，认为与机器翻译的任务可能有较大关联，或许可以借鉴机器翻译的相关工作。在本学期课程中，曾经讲到了机器翻译的实现方法，包括了一个翻译模型和一个语言模型，其中语言模型的作用包含了确定词的出现位置和顺序。因此第一个尝试是使用语言模型进行判定。

当然可以预见仅简单使用语言模型，是很难识别出自然语言句子复杂的特征的。由于需要对序列进行处理，判断某个词语出现是否合法需要用到之前词汇的信息，因此想到了可以使用 RNN 进行学习。

另外通过观察语料库，可以发现一句话错误的原因，经常是某个词与其之后的部分发生了语法错误，而单看句子的前半部分没有错误。因此可以猜想，句子中某个词的出现，不仅依赖于这个词之前的序列，还与这个词之后的序列有关。因此希望引入双向 RNN 来提高网络提取特征的能力。当前在 RNN 相关的任务中，LSTM 的表现非常出色，所以在具体实现中使用 LSTM 或许能取得更好的效果。

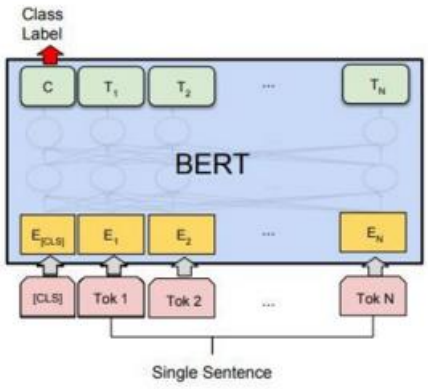
另外，句子通顺与否可以看作为判断语法是否正确，如果能够引入词性标签，语法解析树等特征或许能有所帮助。

技术方案

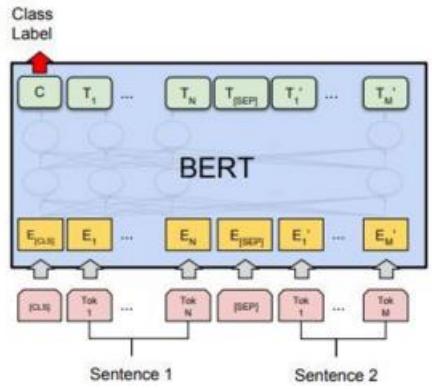
初步尝试：使用 BiGram 的方法，借助了 sklearn 包实现了语言模型。首先得到了二元模型下训练集的统计信息。之后计算出测试集中每句话生成的概率。将概率大于阈值的句子标记为正，否则标记为负。但显然训练集样本仍然不足，未登录词较多，效果不好仅有 51.6%的准确率。因此 LM 只是作为一个基准尝试。

正式设计：经过前述问题分析，选定的技术路线是使用双向 LSTM 训练。需要更大量的训练数据才能有好的效果。不久前，由 Google 提出了预训练模型 Bert，经过了大量语料训练后，效果非常优越，在 nlp 的多项任务中都取得了非常不错的结果。Bert 的模型也是基于双向 LSTM 的思想，因此在本次实验中，可以基于 Bert 的预训练模型进行微调。

下图所示的是本次任务使用的微调模型。输入的是整个句子，Bert 本身网络的输出是一个池化的结果，在此之上再接一层输出层网络即可进行微调训练。网络输出的是两个概率，句子是通顺的概率和句子是不通顺的概率。但该概率偏向于句子是正确的，因此需要重新设置一个阈值进行调整，当句子通顺的概率大于该阈值时，才正式标记为通顺。实验中设置阈值为 0.98。仅经过一个 epoch 的训练，分类效果就明显提高，得分达到了 0.8 以上。



此外，前述分析提到了有必要加入语法分析的特征，bert 中也缺乏词性标注结果的知识，因此又尝试了引入词性标记的效果。实验使用 jieba 对句子进行词性标记，按词的顺序拼接成完整句子，按如下图方式输入到网络中。经过一个 epoch 的训练，最终得分为 0.807585，略微提高但不明显。



遇到的问题及解决方案

训练集中存在有部分标记错误的情况，影响模型训练效果。

用原始训练出的模型对训练集分类，将预测结果不一致且概率大于阈值的部分删去，得到新的训练集后重新训练模型。效果有略微上升。

用到的数据

使用任务提供的训练集 train.txt，测试集 test.content.txt，未使用其他外部数据。

性能评价

使用 CPU 训练，花费的时间较大，设置 batch_size 为 32，训练一个 epoch 大约花费了 10 个小时。可用 GPU 加速，经了解大概可缩短至一小时。

结果分析

受限于计算资源，只训练了一个 epoch，最终在 B 榜的得分为 0.807585。可以预见当加大训练量后，能有更显著的提高。

将来可能的改进

在使用 bert 时引入外部知识，需要更好的设计。比如说可在 bert 的输出层后，将句子的 embedding 向量连同语法的向量再接上某种网络联合提起特征。本次实验中将词性标注信息如同自然语言句子一样直接输入到 bert 中，或许并不是很适合从词性中提取特征。

另外还查阅到有比如 TreeRnn 等网络，便于表示训练语法树，或许能够帮助改进本任务。

程序运行

实验使用的库有：tensorflow 1.12.0，sklearn,pandas,jieba 0.39。
实验环境为 Windows，Python 3.6
尝试性设计的语言模型代码为 LM.py。

基于 Bert 实现的文件为 run_classify.py。如果需要训练模型，则将参数“do_train”设置为 True，如果需要预测则将参数“do_predict”设置为 True。

将 Bert 输出的概率转变为符合提交格式代码文件为 filerw.py。