机器学习工程师纳米学位

句子相似度匹配开题报告

Stephen 2019年4月9日

主要背景

除了Quora，物理学家可以帮助厨师解决数学问题并获得烹饪技巧吗？ Quora是一个获取和分享知识的地方。这是一个提出问题并与提供独特见解和质量答案的人联系的平台。这使人们能够相互学习，更好地了解世界。

每个月有超过1亿人访问Quora，因此很多人提出类似措辞的问题也就不足为奇了。具有相同意图的多个问题可能会导致寻求者花更多时间找到问题的最佳答案，并使作者觉得他们需要回答同一问题的多个版本。Quora重视规范性问题，因为它们为活跃的求职者和作家提供了更好的体验，并且从长远来看为这两个群体提供了更多价值。

现在我们需要通过应用先进技术来分类问题对是否重复来解决这种自然语言处理问题的挑战。这样做可以更容易地找到问题的高质量答案，从而改善Quora作家，求职者和读者的体验。

问题概述

这是一个文本匹配的任务，一般可以用到一些相似度算法，如：

1. 余弦相似度

   > 余弦（余弦函数），三角函数的一种。在Rt△ABC（直角三角形）中，∠C=90°，角A的余弦是它的邻边比三角形的斜边，
   > 即cosA=b/c，也可写为cosA=AC/AB。余弦函数：f（x）=cosx（x∈R）
   > 简单来说这个算法就是通过计算两个向量的夹角余弦值来评估他们的相似度

2. 简单共有词

   > 通过计算两篇文档共有的词的总字符数除以最长文档字符数来评估他们的相似度。
   > 假设有A、B两句话，先取出这两句话的共同都有的词的字数然后看哪句话更长就除以哪句话的字数。
   > 同样是A、B两句话，共有词的字符长度为4，最长句子长度为6，那么4/6，≈0.667。

3. 编辑距离

> 编辑距离（Edit Distance），又称Levenshtein距离，是指两个字串之间，由一个转成另一个所需的最少编辑操作次数。
> 许可的编辑操作包括将一个字符替换成另一个字符，插入一个字符，删除一个字符。一般来说，编辑距离越小，两个串的相似度越大。
> 由俄罗斯科学家Vladimir Levenshtein（找不到他照片233333）在1965年提出这个概念。

4. Jaccard相似性系数

> Jaccard 系数，又叫Jaccard相似性系数，用来比较样本集中的相似性和分散性的一个概率。
> Jaccard系数等于样本集交集与样本集合集的比值，即J = |A∩B| ÷ |A∪B|。
> 说白了就是交集除以并集，两个文档的共同都有的词除以两个文档所有的词。

这里我们需要用机器学习的方法，对已有带是否重复的标签的数据，进行训练，得到一个分类器，用于预测判断给定的两句话是否代表同一个含义。同时，test.csv中提供的200万条，要将这200万条数据加以预测，并提交最终结果。

数据集

1. 训练集 train.csv

字段名	描述
id	一对句子的唯一标识符
qid1	第1句句子的唯一标识符
qid2	第2句句子的唯一标识符
question1	第1句句子
question2	第2句句子
is_duplicate	是否重复 （0为不重复，1为重复）

1. 测试集 test.csv

字段名	描述
test_id	一对句子的唯一标识符
question1	第1句句子
question2	第2句句子

解决方案

由于是个分类的案例，初步的想法，可以找一些分类模型，比如XGBoost，决策树等等，也可以构建神经网络，用sigmod函数来对结果进行二分类。

同时这也是一个NLP的项目，有个很重要的一点是，需要对文本进行嵌入操作（sentence embedding），可以找一些文本数据作为语料， 比如GoogleNews, glove等对现有的句子进行Tokenizer化，转成向量（word2vec）。句子是由词语组成的，而词语的顺序和语义有很大的关联，前一个词语的状态需要带入到后一次训练的过程，所以这里需要用到LSTM（长短期记忆）进行训练。

另外，谷歌最近发布的Bert, 我觉得在特征工程上可以发挥一些作用。

评估指标

根据机器学习工程师纳米学位的要求，最优提交分数需要达到kaggle private leaderboard 的top 20%，对于该题目的就是660th/3307,对应logloss得分为0.18267。

logloss在这里是对数损失，既binary_crossentropy.

> 对数损失，即对数似然损失(Log-likelihood Loss)，也称逻辑斯谛回归损失(Logistic Loss)或交叉熵损失(cross-entropy Loss)，是在概率估计上定义的.它常用于(multi-nominal，多项)逻辑斯谛回归和神经网络,以及一些期望极大算法的变体．可用于评估分类器的概率输出．
>
> 对数损失通过惩罚错误的分类,实现对分类器的准确度(Accuracy)的量化．最小化对数损失基本等价于最大化分类器的准确度.为了计算对数损失，分类器必须提供对输入的所属的每个类别的概率值，
> 不只是最可能的类别．对数损失函数的计算公式如下：

$$L(Y,P(Y|X))=-\log P(Y|X)=-\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M}y_{ij}\log(p_{ij})$$

> 其中，Y 为输出变量，X为输入变量，L 为损失函数．N为输入样本量，M为可能的类别数，yij 是一个二值指标，
> 表示类别 j 是否是输入实例 xi 的真实类别．pij 为模型或分类器预测输入实例 xi 属于类别 j 的概率．
>
> 如果只有两类 {0，1}，则对数损失函数的公式简化为

$$-\frac{1}{N}\sum_{i=1}^{N}(y_i\log p_i+(1-y_i)\log(1-p_i))$$

> 这时，yi 为输入实例 xi 的真实类别，pi 为预测输入实例 xi 属于类别 1 的概率．
> 对所有样本的对数损失表示对每个样本的对数损失的平均值，对于完美的分类器，对数损失为 0 ．

特征工程

1. 检查训练集和测试集文件，解决有些由于“回车”出现语法错误
2. 将空字段的记录移除

3. 整合基本的语法，如isn't 全部改为 is not

项目设计一: LSTM

1. 准备语料库
2. 生成词语嵌入(word embedding)

- 方案一：利用google或者glove语料库，通过word2vec生成词向量
- 方案二：利用bert已经预训练好的模型，生成token embeddings

1. 将两句话的词语嵌入，各自加到LSTM中
2. concat后，添加Dense层
3. 训练模型，标签为is_duplicate字段
4. 用sigmoid收尾，算出概率值

项目设计二: Bert + DNN

1. 准备语料库
2. 生成词语嵌入(word embedding)

- 方案一：利用google或者glove语料库，通过word2vec生成词向量
- 方案二：利用bert已经预训练好的模型，生成token embeddings

1. 生成句子嵌入

- 方案一：利用bert word embedding中的cls字段
- 方案二：将所有词向量求一个平均值作为句子向量

1. 直接加入神经网络，并添加Dense层
2. 训练模型，标签为is_duplicate字段
3. 用sigmoid收尾，算出概率值

项目设计三: Bert Fine Tune

利用bert自带的fine tune微调方式 参考利用bert自带的fine tune微调方式

需要注意的地方

1. Bert训练量非常大，一般的机器估计跑不起来，需要批处理 `train_on_batch` 或者 `fit_generator`

参考资料

- bert论文

- bert项目
- 利用bert自带的fine tune微调方式
- 如何简单的理解LSTM