开题报告--Quora 句子相似度匹配

领域背景

本项目为自然语言处理的句子相似度匹配问题,句子相似度是指两个句子在表达意思上的相似程度。句子相似度匹配直接影响了其他领域的发展。比如信息检索领域,自动问答系统等等,都以句子相似度匹配作为基础。

历史上关于句子相似度匹配的研究包括最长公共子序列的 LCS 算法,字符串快速比较的 MCWPA 算法等。国内李素提出过语句相关的定量计算模型等等。2003 年秦兵等在"基于常问问题集的中文问答系统研究"一文中使用了 TFIDF 方法和基于语义的橘子相似度方法来计算句子相似度,并将这一结果运用到问答系统中。

● 问题描述

这是一个二分类的监督学习任务,需要建立一个模型,这个模型可以判断给出的两个句子的含义是否相同,相同输出 1,不同输出 0。运用 train dataset 训练这个模型,最终这个模型在 test 数据集上的 logloss 要小于 0.18267。Datasets and Inputs

● 数据集和输入

数据为 Kaggle 端的 Quora 数据集[参考 1]。

训练集包含 404290 行数据,每行数据包含 6 列,分别为 id:编号,qid1 和 qid2 为每个问题的独特编码,question1 和 question2:每个问题的文本,is_duplicata:表示question1 和 question2 是否是相同的问题,为 1 表示两者为相同的问题,为 0 表示两者为不同的问题。

题目中明确说明测试集数量为 2345796, kaggle 网站上下载的数据有误,如下图 image1,有重复现象,做了截断处理。数据有三列,为 test_id:编号,question1 和 question2:两个问题的文本。

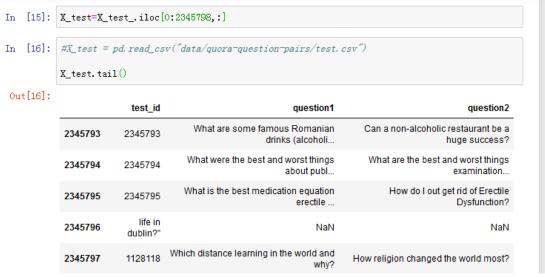


Image1

对于训练集,有 255027 条数据为相同句意,占比为 0.37,代码如下:

```
isdup=0
notdup=0
for i in X_train['is_duplicate']:
    if i==0:
        notdup+=1
    if i==1:
        isdup+=1
print(notdup)
print(isdup)
255027
149263
```

解决方案陈述

在训练模型之前需要明确两个句子的特征是什么即首先需要 feature engineering。在 自然语言处理领域有几种基本的 feature engineering 方法, 比如 TF-IDF, 用词频和逆向 文件频率表示一个词对一句话来说有多重要。比如 Word2Vec, 可以使用预先训练过的模 型将句子中的词转换为向量来表示两个句子的接近程度。Feature engineering 的方法有很 多,选择合适的 feature 在某种程度上决定了模型的好坏。

然后选择一个分类算法,比如 adaboost、随机森林,支持向量机,神经网路等。选择 模型后利用训练数据和 feature 训练模型。

● 对标模型

由 kaggle 网站可知目前 kaggle 使用随机森林模型,使用 log loss 作为评价指标。 Quora Question Pairs 比赛第一名的队伍使用 Siamese 神经网络和 Attention 神经网 络.[参考 3]

● 评价指标

使用 logloss 分数来评价模型[参考 2],下图 image2 中 y 为真实标签,p 为模型判断 为 1 的概率。Sklearn 中有 logloss 计算工具。此项目需要达到 logloss 得分小于 0.18267.

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

Image2

项目设计

首先先做的应该是特征工程,这一步的目的是从数据中提取特征以供算法和模型使用, 对于自然语言处理问题,可以选择 TF-IDF、Word2Vec、fuzzy features 等。

特征选择之后开始建立模型,模型初步认为应该选择集成树,比如 adaboost 或者 xgboost,微软最近的 LightGBM 据说速度要比 xgboost 快很多[参考 5],可以先尝试。随 着训练需要逐渐调整参数或者调整特征。

Image3[参考 4]中可以看到使用 XGboost 结合合适的特征已经可以达到较高的准确 率,可以尝试多加入一些特征,并使用网格搜索等方法调优模型参数。并且可以尝试使用 LightGBM 加快训练速度。

<u>Feature Set</u>	Logistic Regression Accuracy	Xgboost Accuracy
Basic features (fs1)	0.658	0.721
Basic features + fuzzy features (fs1 + fs2)	0.660	0.738
Basic features + fuzzy features + w2v features (fs1 + fs2 + fs4)	0.676	0.766
W2v vector features (fs5)	x	0.78
Basic features + fuzzy features + w2v features + w2v vector features (fs1 + fs2 + fs4 + fs5)	x	0.814

Image3

References:

- 1. https://www.kaggle.com/c/quora-question-pairs/data
- 2. https://www.kaggle.com/c/quora-question-pairs#evaluation
- 3. https://www.kaggle.com/c/quora-question-pairs/discussion/34355
- 4. https://www.linkedin.com/pulse/duplicate-quora-question-abhishek-thakur/
- 5. https://blog.csdn.net/huacha_/article/details/81057150