TAIL CAMP-AI实战训练营 NLP第一周项目报告

## 项目成绩

1. 项目目的

给定一系列的英文句子对，每个句子对的两个句子，在语义上具有一定的相似性；每个句子对，获得一个在0-5之间的分值来衡量两个句子的语义相似性，打分越高说明两者的语义越相近。（句子语义相似度预测）

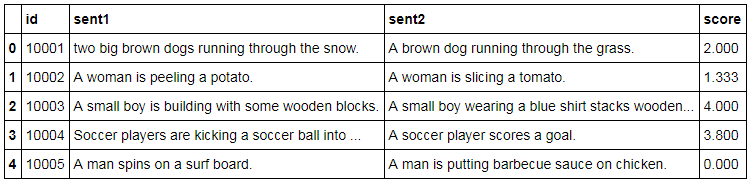
2. 数据集

**数据集**：英文语句文本数据集（txt文件，字段之间以tab分割）

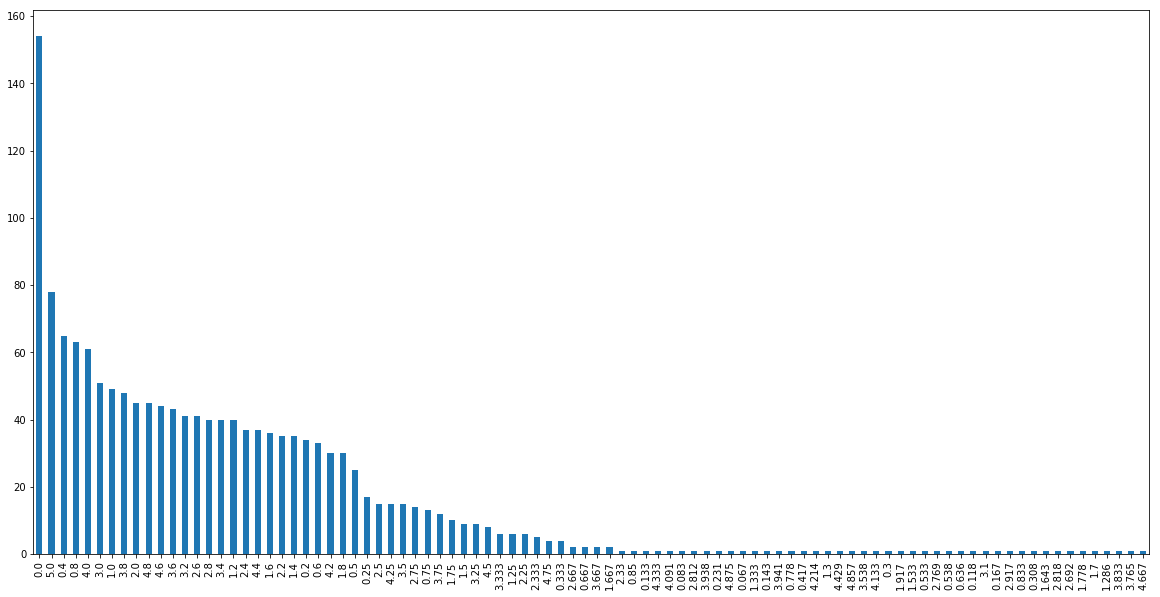
3. 基本思路

3.1 数据分析

数据集预览前几行



Score分布



【数据分析】

分值为0和分值为5的数据较多，模型预测后可能会出现超出 0~5 范围的异常值。

3.2 数据预处理

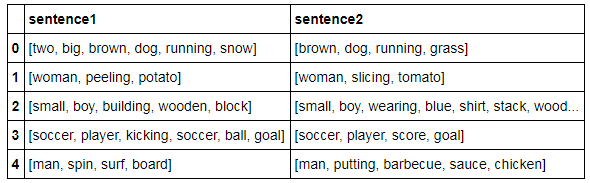
1. 分词

一般而言，相比处理整个句子，我们更愿意去处理一个词汇组成的列表。那么，就需要对句子进行分词。对数据集中的英文文本，可以使用nltk模块下的tokenize进行分词。

1. 去停用词：stopwords.words('english')
2. 转成小写
3. 词变体还原

词变体还原(Lemmatize)类似于词干，但不同的是，变体还原的结果是一个真实的单词。不同于词干，当你试图提取某些词时，它会产生类似的词：increases → increas；而用nltk的WordNet来对同一个单词进行变体还原，才是正确的结果：increases → increase。

预处理后数据集形式如下：



3.3 特征提取

1. 字面特征

common\_words：两个句子共同单词个数 / 两个句子最大单词总数

1. 模糊特征

此处使用了 fuzzywuzzy的库，主要用于字符串模糊匹配，使用Levenshtein距离计算句子间的差异，提取的特征包括：fuzz\_Qratio, fuzz\_Wratio, fuzz\_partial\_ratio, fuzz\_partial\_token\_set\_ratio, fuzz\_partial\_token\_sort\_ratio, fuzz\_partial\_token\_sort\_ratio和fuzz\_token\_sort\_ratio。

1. 基于 word vector 的特征

首先，通过所有句子抽取一个词袋(BOW)，将文档的token映射成id；然后将用字符串表示的文档转换为用id表示的文档向量。此处的BOW仅考虑了词频，下面基于此计算一个TFIDF模型，弥补词频向量化的不足。最后，计算TIDF训练后的语句的余弦相似度tfidf\_features。

Glove语料库+ word vector distance

1. 主题相似度特征

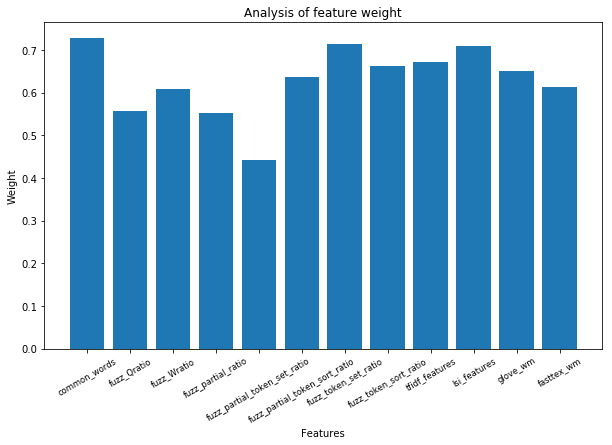
在上述TFIDF的基础上，再次训练一个LSI模型，进行主题相似度分析，然后求解相似度。LSI的物理意义就是将训练文档向量组成的矩阵SVD分解，并做了一个秩为2的近似SVD分解。

3.4 模型分析

此处采用传统的回归模型，挑选了三个：XGB、RF和GB，然后利用Stacking对三个模型进行融合。

3.5 特征权重分析

仅考虑单个特征，利用3.4节的Stacking模型默认参数，求解pearsonr系数。下图为各特征权重分布：



4. 基本实现步骤

5. 评测结果展示

6. 遇到的问题和思考