《计算两个句子的相似度》
项目报告

团队名称：NLP_newbee

团队成员 1：105_李成飞

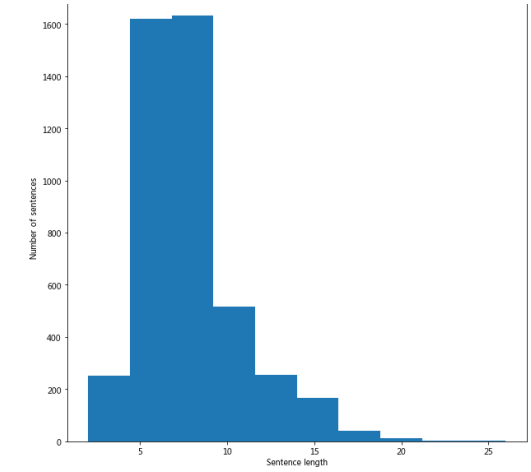
团队成员 2：110_常永炷

自然语言处理（NLP）与计算机视觉（CV）一样，是目前人工智能领域里最为重要的两个方向。如何让机器学习方法从文字中理解人类语言内含的思想？这是一个很广阔的问题。参加了好未来的 AI lab 训练营，我们迎来了第一周的任务《计算两个句子的相似度》。写过一篇关于这个题目的一些想法的帖子：https://www.kesci.com/apps/home/competition/forum/5a7024470fd6d04e2be93dcb 。在这个任务中，每个人都面临着从两个角度思考。第一：利用统计特征和传统机器学习模型来给出解决方案；第二：利用深度学习给出解决方案。我们团队也不例外，我和我的小伙伴分工合作，从两个角度来给出了解决方案，最终将两个人的结果做加权平均，形成了该任务的第一周的解决方案。接下来就是说明一下两个角度的具体步骤。最终的线上评测结果如下：

排名	浮动	团队	分数	提交次数	最佳成绩提交时间	最后提交时间
1	-	biorad	0.874353	16	18-02-02 21:07	18-02-02 21:09

一、 利用统计特征和传统机器学习模型

1） 我们首先采用了基本的数据探索，下图表示句子长度的分布：



2） 进行了如下的数据清洗方式

- 删除所有不相关的字符，如任何非字母数字字符
- 把文字分成单独的单词来标记解析
- 将所有字符转换为小写字母，使「hello」,「Hello」和「HELLO」等单词统一
- 考虑将拼写错误和重复拼写的单词归为一类（例如「cool」/「kewl」/「cooool」）
- 考虑词性还原（将「am」「are」「is」等词语统一为常见形式「be」）

3） 特征统计

最常见的统计特征主要包括 Bag of Words、TF-IDF、直接索引方式

（word2index）、各种相似度计算。在这个任务中，我们尝试了多种特征，比如直接使用索引方式、Bag of words、tf-idf 等，然后使用传统机器学习模型进行学习。经过了多次的探索，我们生成的特征主要包括：

1. n-gram @ word and char level

使用了 1-3 的字符和单词的 gram，计算了各种距离，其中包括 simhash ,还有 Ochiai，Dice，Jaccarc 在集合层次计算相似度的一些衡量手段。

2. 基本的句子（字符和单词）统计特征

分词性统计了名词，动词相同的个数，基本的句子长度，词的数量，相同词的匹配率，tfidf 的求和，平均，长度（后来思考一下这些特征做一些多项式组合可能会更好。)针对有无停用词的相同比率，相同词的数量不同词的数量等。还有针对字符级别的异同数量统计。

3. 求句子的表示，计算向量的距离和分布特征

句子的表示采用了两种方法，一种是 Bag of words 的直接平均，一种是是 Bag of words 的 tf-idf 加权平均。拿到句子表示之后，计算了向量的距离：cosine, manhatton, euclidean。以及数据特征：pearson，spearman，kendall。

4. doc2vec 产生词向量

PV-DBOW，PV-DM w/average，PV-DM w/concatenation - window=5 得到向量之后衡量相似度。

5. （1-3gram）tfidf 的向量

拿到向量后计算距离。其中包括 cosine、manhatton、euclidean。

4） 模型使用

单模型：SVR ，RF(随机森林)， Gboost，XGBoost，LightGBM，

多模型：1、Stacking： 第一层模型：SVR 、RF(随机森林)、 Gboost、 XGBoost、LightGBM；第二层模型： XGBoost；2、Aver

age：RF(随机森林) 、SVR、Stacking、XGBoost

实验结果中均表示线下结果，将训练数据随机抽取 20%进行测试

最终的实验结果

Text2vec	SVR	RF	Gboost	XGB	LGB	Stacking	Average	线上
W2I	Nan	0.3908	0.3758	0.3914	0.3782	0.4240	**0.4302**	0.3243
BOW	0.5842	0.5419	0.5393	0.5941	0.5265	0.6465	**0.6489**	0.5916
TF-IDF	0.4939	0.5679	0.5572	0.6273	0.5813	0.6413	**0.6454**	0.5885
BOW-TI	Nan	0.5630	0.5967	0.6340	0.5933	**0.6540**	0.6461	0.5795
统计特征	0.8103	0.8283	0.8303	0.8309	0.8198	0.8292	**0.8348**	0.8110

备注：W2I：利用字典索引进行向量化。BOW: bag of words。BOW-TI：bag of words 和 tf-idf、出现 nan 之后线上提交已经把这个模型给去掉了，线上提交结果均为表中加粗的结果。

二、 利用深度学习进行实验

1. 任务实现的基本思路

在卷积神经网络和循环神经网络模型上的调参和改进，发现在卷积神经网络和循环神经网络的混合提取特征，模型会取得不错的结果。

2. 任务实现的基本思路

Step1：首先是对原始文本数据进行预处理，通过 Keras 深度学习框架对其进行 sequence 的预处理，和填充（选择每个句子处理后的长度，和词嵌入的维度）。

Step2：通过 Keras 对输入的句子进行分词，并通过 Keras 的 Embedding 层对其进行词嵌入

Step3：使用 Keras 的 Function API 建立两个输入（sentence1 和 sentence2）的模型。

Step4：对输入进来的两个句子分别使用卷积神经网络（一维）和 GRU 处理，其中包括（卷积+GRU，卷积，GRU），不同的方法进行特征提取。

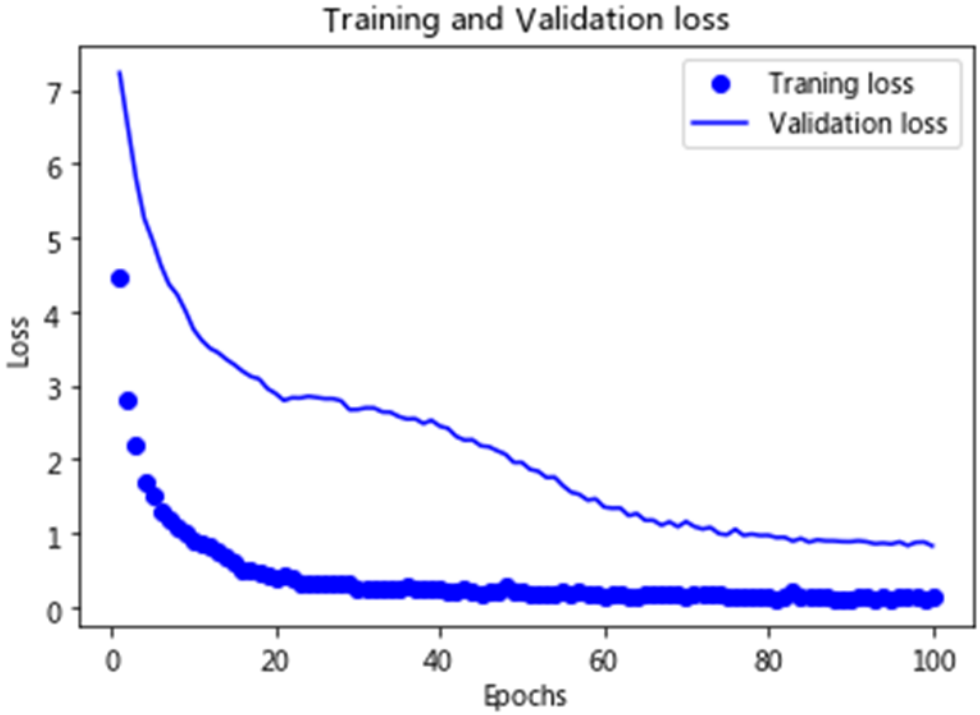
Step5：对每个句子处理后的特征 vector 进行拼接。

Step6：对两个句子的特征 vector 进行拼接。

Step7：对拼接后的 vector 输入到全连接层（dense）中。

Step8：最后只有一个输出的节点，使用 mse 损失函数进行优化和训练。

其中损失函数训练图如下所示：



三、 总结

将深度学习模型结果与机器学习结果进行加权平均得到了最终的结果。

四、 遇到的问题

在进行实验过程中遇到了许许多多的问题。比如在进行特征工程时，如何构造新的特征？以及怎么确定新特征是否有效等等。还有在进行深度学习模型训练过程中如何解决深度网络容易过拟合的问题。当碰到这些问题首先想到的解决办法就是去查找资料、文献等。比如在解决新特征是否有效时候，我们采取了直接线上验证的方式，在解决深度网络过拟合问题采用正则化。

五、 希望与导师、同学交流的地方

1. 如何根据自己的数据集确定神经网络的规模？（网络的层数，神经元的个数）
2. 对于卷积+循环神经网络对于文本的分析，是否有更好的解释？
3. 神经网络的特征可解释性一直是一个诟病，那反问一句，可解释真的有必要吗？（在文本方面）
4. 在特征工程上如何进行特征选择？
5. 一些魔法特征到底该不该使用呢？所谓魔法特征就是效果很好，但是特征难以解释。

六、 未尝试的想法

在此次任务中，由于时间紧任务强度较大，因此有许许多多的为改进的地方，因此这个只是一个很粗糙的解决方案。其中未尝试的地方包括：

1、 机器学习中特征选特，在此次任务中批量生成特征，未考虑特征间的相互影响，也未做特征选择等工作。

2、 机器学习模型的调参，在此任务中的模型参数均为由较丰富的使用模型的经验所来，没有根据此次任务进行专门的调参工作。

3、 没有实验过多的神经网络模型，此次任务的神经网络模型为经典论文的结构进行微调所得。没有进行神经网络模型的融合。

4、 没有尝试将机器学习与深度学习结合。比如将深度学习所训练倒数第二层的权重进行机器学习训练。

5、 未尝试设计一个多模态输入的网络。多模态输入是利用统计特征和原始数据形成的多源数据输入到神经网络中进行训练。

希望在完成此次训练营所有任务之后，对没有探索过的想法进行充分的实验，进行更进一步的学习。