## 《基于深度学习的情感分类和智能客服研究与实现》

1. **实验目的**
2. 训练学生进行科研文献查询和进行文献分析的能力；
3. 掌握文献综述的写作方式方法；
4. 熟悉深度学习的相关知识和自然语言处理的方法；
5. 采用LSTM网络和相关的深度学习平台如TensorFlow，Caff深度学习软件实现情感分类；
6. 培养从事科研工作的基本能力和软件开发解决实际问题的能力。
7. **实验内容**

本次科研训练要求每个学生独立完成项目的原型开发。

项目方向：

基于深度学习的情感分类和智能客服研究与实现；

实训会提供训练素材。

1. 情感分类中主要提供10000条以上的已标记正负类的文本用于训练。

数据主要是购物评价。

1. 会给出回复意图标准和文本的参考模板。

测试要求：

学生设计原型系统后，将输入测试语句，由原型系统首先给出情感分类结果（正类积极，负类消极），然后根据分类结果给出智能回复。

人机接口可采用Web界面方式（选做）。

1. **实验方法**
2. 利用中国知网、万方等搜索渠道查询相关的深度学习资料学习。
3. 利用网易云课堂、B站等网站进行相关的深度学习、自然语言处理的视频学习。
4. 参考github、CSDN等相关源码和资料进行学习。
5. 利用PyCharm、Anaconda和jupyter开发工具进行实验开发。
6. 使用TensorFlow、Keras、Flask和Python技术实现项目目标。
7. **实验步骤**
8. 搭建实验开发的环境。

Python 3.7.3，

TensorFlow 1.14.0，

Flask 1.1.1，

Anaconda3，

Jupyter notebook，

PyCharm 2019.1.3 (Professional Edition)，

Windows 10。

1. 利用网易云课堂、B站等网站进行视频学习，学会TensorFlow、Keras的简单使用。
2. 阅读深度学习、自然语言处理的相关文档，了解情感分析和智能客服实现的常见方式。
3. 准备实验数据。

4.1 本次实验数据为老师提供，所以只需要对数据作进一步的确认处理即可。

4.2 本次实验数据为酒店和书籍的评论，分为积极评论和消极评论两类，消极评论中又根据评论的内容分为书籍内容、书籍服务、书籍质量、书籍物流、书籍价格、酒店服务、酒店设施、酒店周边环境和酒店价格共计9个类别。

4.3 对积极评论和消极评论进行人工筛选，去除脏数据（即不符合该类别的数据）、过长句子删减、去除不相关语句（比如宾馆反馈的回复）、增加常见的短句（比如酒店价格很便宜等）。

1. 根据课堂讲解的“中文文本分类基础知识”和CNN、LSTM模型的了解，开始进行项目的开发。
2. 实现评论的二分类（判断评论为积极还是消极）算法。

6.1 加载数据。

利用pandas模块的read\_excel（）函数对数据（pos.xls、neg.xlsx）进行读取并保存到一个评论list中。

6.2 数据上标签。

二分类算法中积极评论的标签为“1”，消极评论的标签为“0”。

6.3 中文分词。

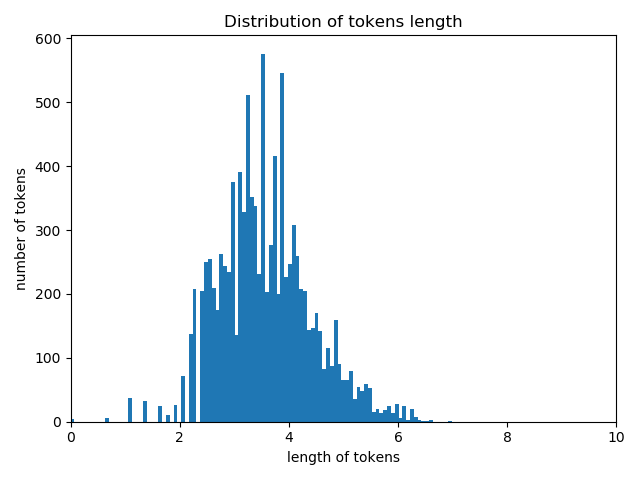
加载分词向量，使用gensim加载预训练中文分词embedding，本次实验采用的是知乎的分词向量sgns.zhihu.bigram。

6.4 提取文本关键词。

6.5 建立tokens字典。

利用先对所得的评论list去除标点，再利用结巴分词对结果进行处理。

6.6 使用tokens字典将“文本”转化为“数字列表”，对tokens的分布进行可视化处理，利用matplotlib.pyplot可以实现。



6.7 截长补短让所有“数字列表”长度都是一致，取tokens平均值并加上两个tokens的标准差作为最终长度，保证每个文本都是同样的长度可以避免不必要的错误。

6.8 Embedding层将“数字列表”转化为向量列表。

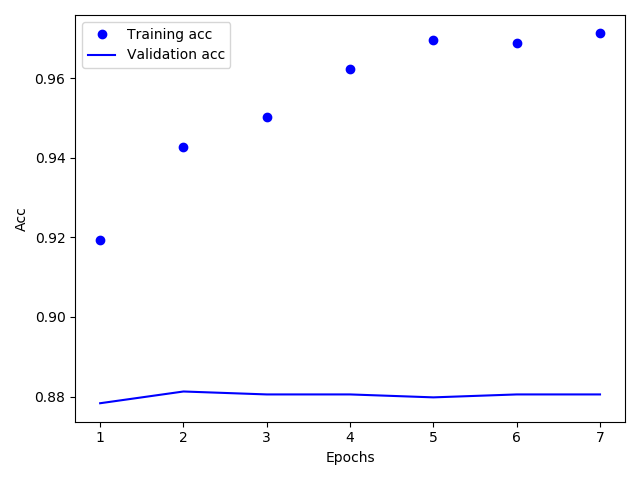
6.9 将向量列表送入深度学习模型进行训练。

主要是利用双向和单向LSTM、Dropout、中间层数、注意力机制、激活函数、loss函数、学习率等对模型进行实现和不断优化。

6.10 保存模型与模型可视化。

利用model的save函数将已经训练好的模型进行保存，同时在模型训练中加入Tensorboard的相关函数对过程可视化处理。

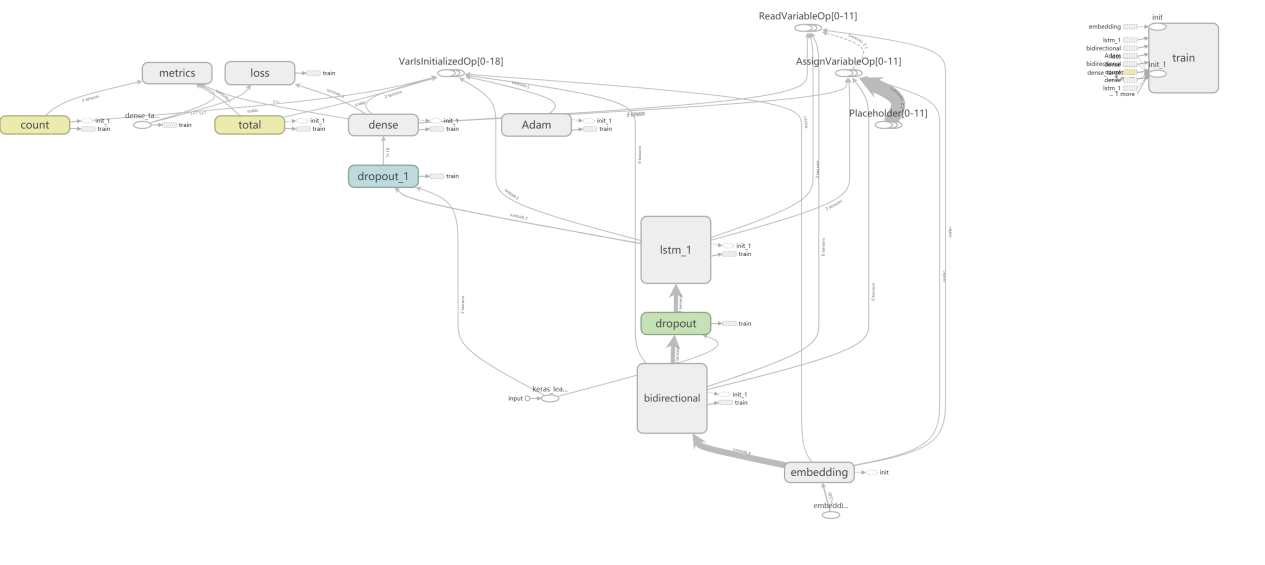
利用matplotlib.pyplot模块可以查看训练过程每一次epoch后的训练精确度和验证精确度的变化过程。



在命令行输入如下命令可以查看训练过程中各项数据的更加详细变化过程。

|  |
| --- |
| tensorboard --logdir=F:\Desktop\sentimentAnalysis\static\logs\class2 |

算法的模型结构图：



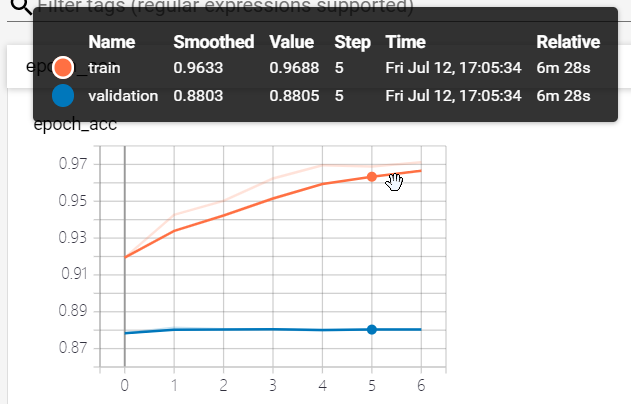
6.11 模型的预测功能。

对输入的文本先进行去除标点符号操作，再进行分词操作、tokenize、padding等，最后调用model.predict（）进行结果预测，得到一个在0-1范围内的浮点数即精确度，大于0.5即偏向积极情感，反之则是偏向消极情感。

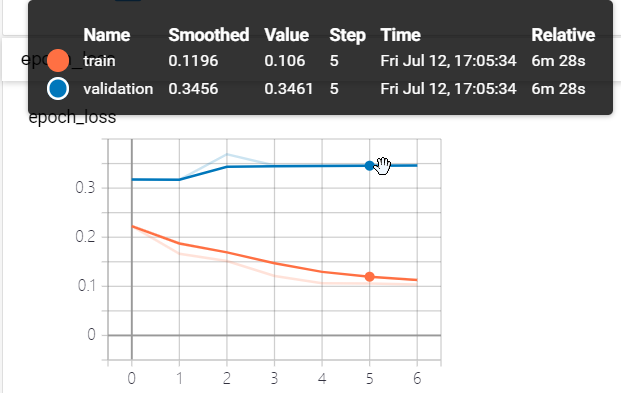
6.12 训练过程可视化。

在训练过程中加上Tensorboard相关操作即可记录实时的精确度、loss、learn rate的变化过程。

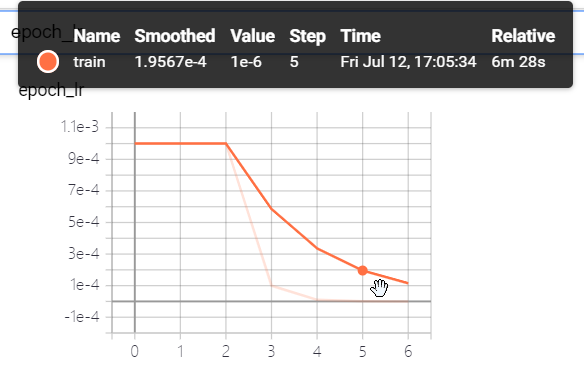
训练过程中的train和validation和accuracy的变化如下所示：



训练过程中的loss的对比变化过程：



训练过程中的学习率learn rate的变化过程：



1. 实现对消极评论的九分类即进一步判断消极评论是哪一类的算法。

消极评论的九分类和评论的二分类算法基本相似，主要是对标签编码进行独热编码处理和预测结果的调整即可，所以这里主要讲述九分类算法与二分类算法的区别。

7.1 数据读取时只读取消极数据，标签根据九个类别分为0-8。

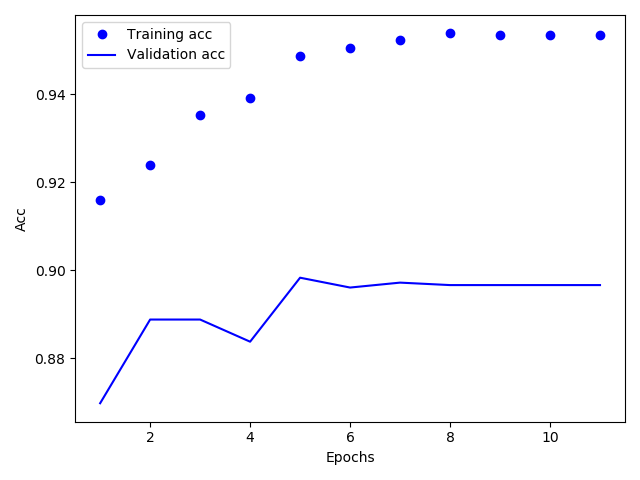
7.2 利用tensorflow.python.keras.utils的to\_categorical（）函数对标签进行one-hot编码处理。

7.3 训练模型对比二分类删去了双向的LSTM，因为根据测试结果对比，本次项目更适合仅使单向的LSTM，其他参数基本不变。

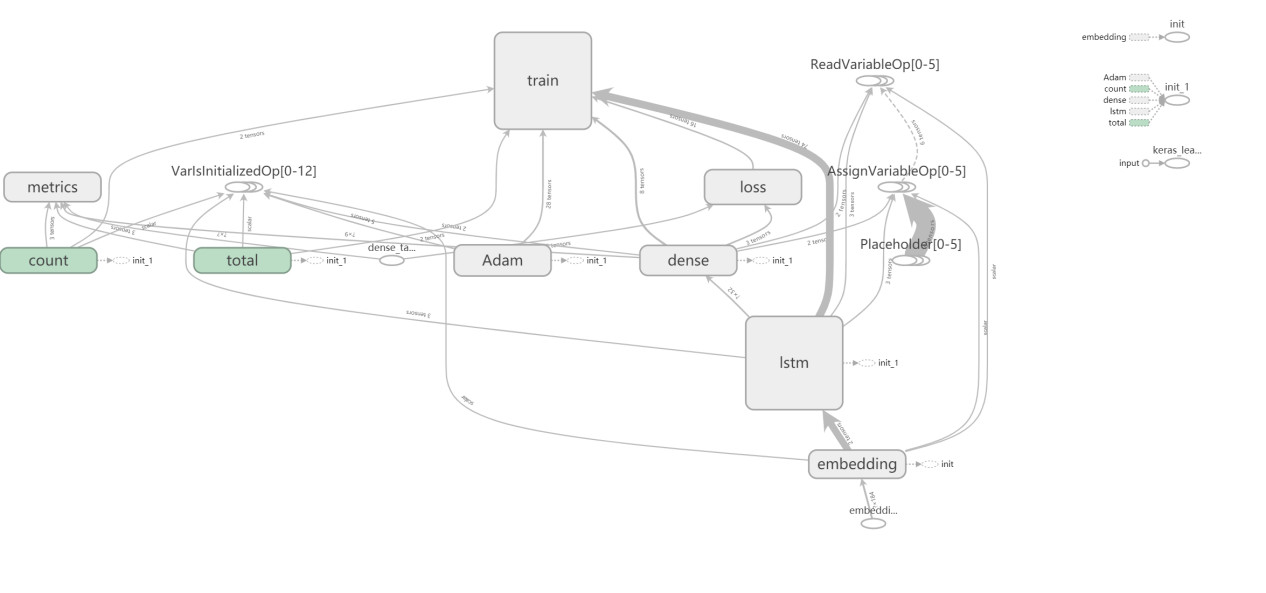
7.4 预测结果的是一个一维向量，包含九个比率，利用numpy的argmx（）函数选择其中最大的比率作为最终预测的类别。

7.5 训练过程的模型及相关的数据的可视化展示。

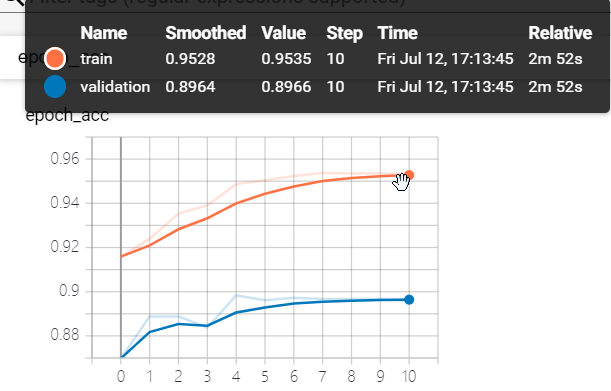
每次epoch后的训练精确度和验证精确度的变化：



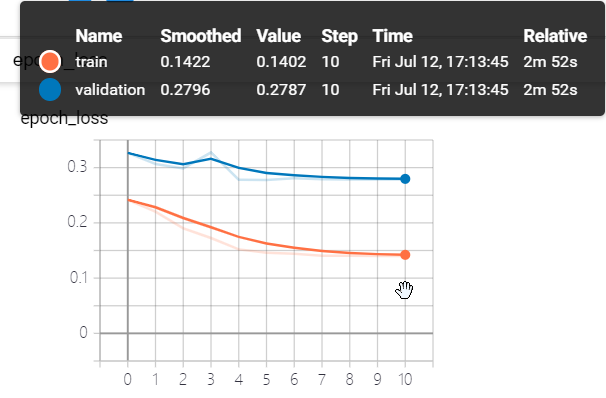
算法的模型结构图：



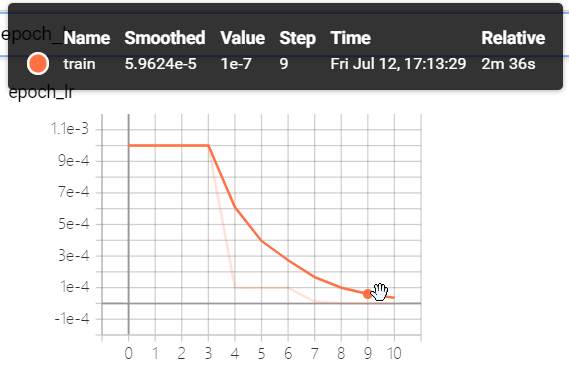
训练过程中的train和validation和accuracy的变化如下所示：



训练过程中的loss的对比变化过程：



训练过程中的学习率learn rate的变化过程：



1. 结合Flask框架实现用户交互界面，便于成果展示与操作。

8.1 编写用户交互的html页面，一个是进入的欢迎页面，一个是用于用户输入评论及系统进行回复的操作页面。

8.2 编写模型调用函数useModel.py，对一串文本进行预处理后进行二分类预测，如果为积极，则显示情感倾向，类别，并通过对积极回复词典的随机选取进行回复，如果为消极则进行九分类预测，判断究竟为哪一类别的消极情感，之后显示情感倾向、类别、吐槽类型，再根据吐槽类型进行有选择性的回复。

8.3 编写flask的页面跳转函数，获取url中的用户输入评论并调用预测函数，获取预测结果再将数据传递到前端页面中显示给客户。

8.4 对于用户的评论回复是根据系统既定的N条模板和用户评论的类型（如积极或消极、消极的类别）进行不同模板选择，同时同一类型有多条回复语句，利用随机数可以随机选择一条即可以做到同一条评论在不同次提交时回复的评论并不相同，这样就可以显得更加像是人工回复。

8.5 用户交互的界面如下所示：

程序在运行后会生成一个网址，用户在浏览器输入该网址即进入到系统的首页。



点击【点击进入】后即可来到用户操作界面，在【用户评论】下的文本框中可以输入想要预测的语句（仅限书籍和酒店相关评论），之后点击【提交】，即可看到系统的回复信息，比如情感倾向、评论情感是积极还是消极等等。



1. 对项目的有关参数进行调整，优化模型，提高精确度。

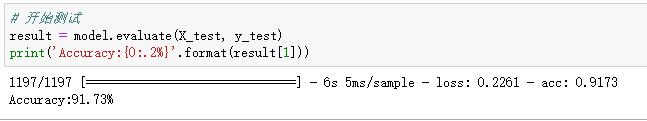
一方面是对数据模型的更换，比如CNN、LSTM、BERT模型的比较，然后根据精确度和loss等数据进行综合评价择优选取，对于本实验而言因为数据量并不多，所以LSTM模型是最合适的。

另一个方面则是对参数的调整，如模型的层数、训练集和验证集的比例划分、训练集和测试集的比例划分、模型参数batchsize调整等等。

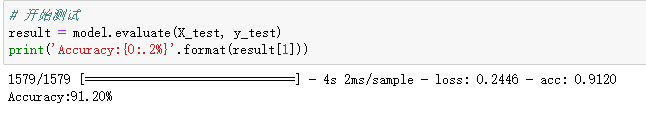
1. 总结实验过程并进行思考汇总，完成实验报告编写。
2. **实验结果**
3. 模型的精确度

根据实验的多次模型调整、参数修改与结果优化，二分类和九分类模型的精确度都达到了90%以上，效果相对而言是比较好的。

二分类算法模型的精确度可以达到91.73%，



而九分类模型的精确度可以达到91.20%，



1. 模型的实际评论测试效果

2.1 测试1：积极数据——《平凡的世界》书如其名，没有任何传奇色彩的跌宕起伏，也没有武侠小说的那种刀光剑影血雨腥风，也没有时下一些爱情小说的缠绵悱恻深刻动人。第一次在图书馆读到就被深刻的吸引了，多年后重新阅读，依旧动人心弦。

测试结果：结果显示评论情感为【积极】，情感倾向达【93%】，客服回复为【感谢您的光临，您的满意，就是我们最大的安慰，最大的回报，我们加倍努力做得更好！(\*^\_\_^\*) 嘻嘻……】，即能够正确预测并回复。



2.2 测试2：消极数据+酒店价格——涨价了，不爽………之前价格很给力，很有性价比，现在感觉这价格有点鸡肋了

测试结果：预测的评论情感为【消极】，情感倾向为【44%】，吐槽类型为【酒店：价格】，客服回复为【尊敬的宾客， 感谢您抽出宝贵的时间给我们点评，您提出的建议我将会转达给相关部门，并尽最大努力做出改善，期待与您的下一次见面。】，是对消极评价的正确处理，即测试结果正确。



2.3 测试3：消极评论+书籍内容——让我吐槽一下这个纸质啊，为什么永远都这么粗糙呢？再好的书被这种纸一印就不好了！

测试结果：预测的评论情感为【消极】，情感倾向为【0%】，吐槽类型为【书籍：内容】，客服回复为【亲，对于您提到的书籍内容问题我们深感抱歉，我们一定会再接再厉不断改进！】，能够识别为书籍内容且对内容的消极评论给予对应的回复，即测试结果正确。



2.4 测试4：无关评论测试（评论不在书籍和酒店评论范围）——这个学期的课程很多，真的忙不过来。

测试结果：预测的评论情感为【消极】，情感倾向为【1%】，吐槽类型为【书籍：物流】，客服回复为【感谢反馈支持，非常抱歉物流问题带给您不便~小店一定全方位改善！重新测试包装，甄选包装，优化配送！对您的不快再次表示抱歉，您的反馈使小店进步，祝您生活愉快！】。

本次测试结果预测的情感倾向正确即消极评论，但是情感分类出现错误，判断为了【书籍：物流】，然后并就书籍物流问题给予了对应的回复。

测试结果分析，从本次测试结果可初步判断本模型对情感的二分类即积极或者消极的判断具有一定的迁移能力，但是对于情感类别判断由于程序本身限制是不能够正确识别的，不过客服回复依旧可以根据系统判断的吐槽类型进行回复，所以模型依旧有待完善。



1. **实验结论**

根据实验的最终模型的测试精确度和测试结果来看，本项目——【基于深度学习的情感分类和智能客服研究与实现】已经成功完成，符合项目需求，且正确率较高。

对于用户输入的有关书籍和酒店的相关的评论基本上是可以正确判断，可以分别判定为积极抑或消极，对于消极的评论还能进一步做类别判断。对于积极评论可以从积极回复模板中随机选取一条作为回复，消极评论则是先选择对应问题的回复类型，然后再从该类别的多条回复语句中随机选取一条作为回复。这样的回复设置可以给人更多的真实感，让用户感觉是真正的人工回复而不是机器回复。

同时，相对的本模型对于书籍和酒店之外的评论具有一定的迁移能力，但是也不能完全判别出来。积极和消极的预测是较为准确，但是消极类别则是无法判断，这是因为在深度学习训练时并未作相关处理造成的。

除此之外，由于训练数据集数量的限制和测试的问题集限制本模型可能还有部分隐藏问题未被发现，这些都是有待完善改进的。对于客服回复，也可以做进一步的模板优化，比如增加数量和类别，对问题一方面利用深度学习的进行类别判定，另一方面可以进行关键字提取判定类型，从而提高回复的精确度和灵活度。

1. **实验小结**

通过本次实训，我主要是了解了深度学习的一些基本知识，学会了简单使用TensorFlow和Keras进行中文文本自然语言情感分类处理，了解了多个深度学习和文本处理的模型，比如CNN、LSTM、BERT等，扩大了知识面。还有经过课堂讲解，了解到深度学习还可以应用于目标检测、风格迁移、自动作曲等各个方面，也增加对深度学习的兴趣。

在实验中也是锻炼了个人学习能力与动手能力，对于新兴技术的学习问题作了实际的了解。从查阅资料，观看学习视频，自己动手操作，直至最后的模型调整和参数修改等等都是很考验人的。

在实验过程中也是遇到过各种各样的bug，最揪心的一个是在调用训练好的模型用于Flask网页前后端交互时一直报错【ValueError: Tensor Tensor("dense\_1/Sigmoid:0", shape=(?, 1), dtype=float32) is not an element of this graph.】，这个bug真的是调了前前后后8h+。这个bug主要是model加载和使用的问题，最后解决能用时是每次预测文本时都重新加载model。可是这样却也导致程序运行速度过慢，因为每次输入评论就要加载模型。最终发现这个问题的原因是models.load\_model(……)后必须使用一次model.predict()方法，这样就不会出现该问题了，同时也只需要加载一次model就行了，完全不影响使用速率，所以咎其原因还是语法学习的不够深入。

同时，在模型选择和参数调整时，也出现过越调整精确度越低的情况，还有过拟合、欠拟合等问题，这些通过查阅相关资料、与同学讨论或者询问老师都是得到了解决。

本次实验在刚开始学习时方向不明确甚至走错了，浪费了很多时间，也做了很多无用功，这也给了我一个教训，做正确的事胜过正确的做事。同时实验的模型选择方面还有待提升，部分模型也还未作完整的对比。参数选取方面的理论学习有待加强，不能只是随机的进行调整。模型的迁移能力以及对非书籍和酒店评论的处理问题也有待完善。

学无止境，在不断的尝试失败中也是在不断的进步。所以，此次实训还是收获良多，综合素质能力也是得到了不少提升。最后也衷心感谢同学、学长学姐和老师的帮助指导，谢谢。

1. **参考文献**

【1】杜永萍,赵晓铮,裴兵兵. 基于CNN-LSTM模型的短文本情感分类[J/OL]. 北京工业大学学报. 2019（04）

【2】邢长征,李珊. 文本情感分析的深度学习方法. 计算机应用与软件[J/OL]. 2018（08）

【3】aespresso. chinese\_sentiment[OL]. github. 2019（06）

【4】莫烦Python. Tensorflow 搭建自己的神经网络[OL]. github. 2018（05）

【5】庄宁. 基于Tensorflow的Android端本地图像分类的应用设计与实现[D]. 浙江工业大学. 2019

【6】一种基于TensorFlow结合多层CNN网络的交通灯识别方法[P]. 中国. G06K9/00.CN109508635A. 2019-03-22

【7】Kashfia Sailunaz,Reda Alhajj. Emotion and Sentiment Analysis from Twitter Text. Journal of Computational Science[J/OL]. 2019（07）

【8】keras-team. Keras中文文档[OL]. github. 2019

【9】widiot0x. TensorFlow实现CNN用于中文文本分类[OL] . CSDN. 2018（02）

【10】涤生（bluez）. 文本挖掘&情感分析[OL]. CSDN. 2019（04）

【11】王学贺,赵华. 基于Word2vec和多分类器的影评情感分类方法[N]. 宁夏大学学报(自然科学版). 2019（06）

【12】张闯. 基于深度学习的知乎标题的多标签文本分类[D]. 北京交通大学. 2018（06）