*建议本文档使用markdown编辑器打开

介绍

本项目的核心部分是基于tf-idf检索的召回模型，构建召回+排序的客服聊天机器人。系统支持FAQ问答模式的客服机器人，采取的数据集是小鸡孵化器相关垂直领域的FAQ问答数据集。

目前该系统的优点在于：
一、 召回+排序 2个模块互不干扰，便于自定义修改以及维护；
二、系统采取了排序规则优化，提升了检索速度。
三、加入了简单的倒排索引，优化了检索流程。

环境配置

Python版本为3.6

需要创建Python=3.6的虚拟环境

详细配置见requirements.txt或者qa.yaml

这里提供两种配置项目运行环境的方法

1.使用pip安装

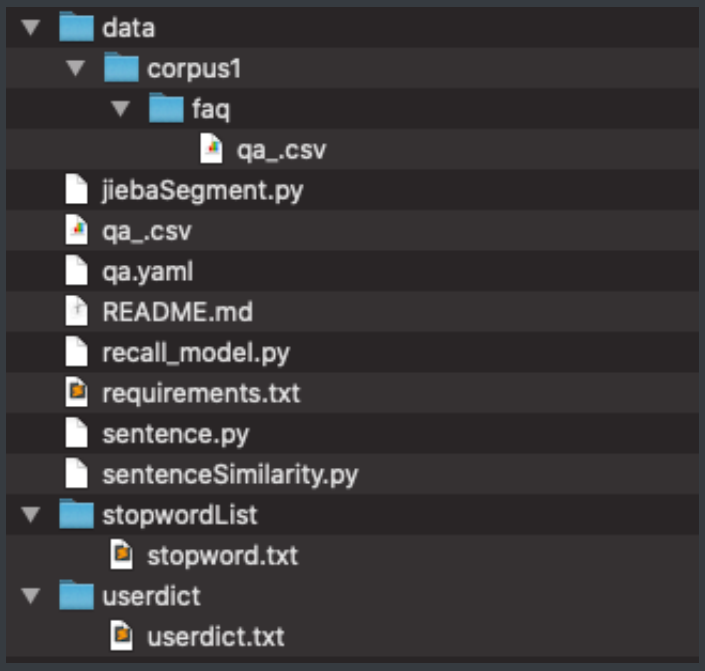
```
conda cerate -n qa python=3.6 #创建名为qa，Python版本为3.6的conda虚拟环境

conda activate qa # 激活虚拟环境

pip install -r requirements.txt # 安装环境依赖库
```

2.使用conda安装

```
conda env create -f env.yaml
```

个人建议：更换清华源后，使用第一种方法安装

文件结构说明



qa_.csv文件为数据存放文件

即孵化器问答对

工程师如需添加问答对，直接在文件末尾添加即可

jiebaSegment.py、**sentence.py**、**sentenceSimilarity.py**均为辅助文件

里面包含了主文件运行需要调用的、且开源库没有的自定义函数

stopword.txt文件为项目停用词一般不需要工程师修改

userdict.txt文件为用户定义词典

为了预防数据集中出现生僻词，jieba分词不可识别，这时需要工程师自行添加

里面已经包含了示例

项目运行

进入项目文件夹顶层

激活conda环境

```
conda activate qa
```

运行主函数文件

```
python recall_model.py
```

部分代码说明

```
24  def read_corpus1():
25      qList = []
26      # 问题的关键词列表
27      qList_kw = []
28      aList = []
29      data = pd.read_csv('./data/corpus1/faq/qa_.csv', header=None)#读取CSV文件
30      data_ls = np.array(data).tolist()
31      for t in data_ls:
32          qList.append(t[0])
33          qList_kw.append(seg.cut(t[0]))
34          aList.append(t[1])
35      return qList_kw, qList, aList
```

recall_model.py文件第29行：读取问答对

```
113 if __name__ == '__main__':
114     # 设置外部词
115     seg = Seg()
116     seg.load_userdict('./userdict/userdict.txt')
```

recall_model.py文件第116行：读取外部词

```
128     while True:
129         question = input("请输入问题(q退出): ")
130         if question == 'q':
131             break
132         time1 = time.time()
133         question_k = ss.similarity_k(question, 5)
134         print("您好，我给您找到的答案是：  {}".format(answerList[question_k[0][0]]))
135         for idx, score in zip(*question_k):
136             print("same questions:  {},                score:  {}".format(questionList[idx], score))
137         time2 = time.time()
```

recall_model.py文件第129行：输入问题

recall_model.py文件第129行：输出匹配的问题和回答

```
123     ss.set_sentences(questionList)
124     ss.TfidfModel()          # tfidf模型
125     # ss.LsiModel()          # lsi模型
126     # ss.LdaModel()          # lda模型
```

recall_model.py文件第124行：选择调用核心函数

三个召回模型已经在函数文件写好，在主文件直接调用即可

tfidf模型，精度最高，但是语义映射到了高维空间，速度相对其他两个较慢

建议问答对5位数以内使用该模型，5位数以上使用其他两个

运行Demo

可以看到最后的语义理解还是非常喜人的