中 国 科 学 技 术 大 学软件学院

**工程实践**

详细设计



**基于知识图谱的人机智能对话系统**

项目成员：郭亚杰，臧磊光，何超

研究方向：自然语言处理，知识图谱

导 师：赵振刚老师

目录

[1 引言 3](#_Toc77975898)

[1.1编写目的 3](#_Toc77975899)

[1.2 背景 3](#_Toc77975900)

[1.3 参考资料 3](#_Toc77975901)

[2 开发环境 5](#_Toc77975902)

[2.1开发工具 5](#_Toc77975903)

[3 模型介绍 5](#_Toc77975904)

[3.1 BiLSTM + CRF 5](#_Toc77975905)

[3.2 Bert模型 7](#_Toc77975906)

[4 问答界面 8](#_Toc77975907)

[5 页面布局设计 9](#_Toc77975908)

[5.1 组件设计 10](#_Toc77975909)

[5.2自定义hooks 10](#_Toc77975910)

[5.3网络请求设计 11](#_Toc77975911)

[6程序(智能问答)设计说明 11](#_Toc77975912)

[6.1 程序描述 11](#_Toc77975913)

[6.2 功能 11](#_Toc77975914)

[6.3 性能 11](#_Toc77975915)

[6.4 输入项 12](#_Toc77975916)

[6.5 输出项 12](#_Toc77975917)

[6.6 算法 12](#_Toc77975918)

[6.7 接口 12](#_Toc77975919)

[6.8 存储分配 12](#_Toc77975920)

[6.9 注释设计 12](#_Toc77975921)

[6.10 限制条件 12](#_Toc77975922)

[6.11 测试计划 12](#_Toc77975923)

**详细设计说明书**

# 1 引言

## 1.1编写目的

此说明书针对整个人机对话工程的设计过程进行了详细的说明，对需要了解整个工程细节的读者来说，这是一个详细的参考文档。

## 1.2 背景

（1）软件系统的名称是：基于知识图谱的人机对话系统设计

（2）本项目的提出者是赵振刚老师，开发者是：郭亚杰、何超、臧磊光。

用户群体是：所有需要得到聊谈服务的用户。

## 1.3 参考资料

[**1**]  <http://bbs.talkop.com/forum.php?mod=viewthread&tid=119685>

[**2**]  <https://games.qq.com/a/20150615/057918.htm>

[**3**]  <http://kw.fudan.edu.cn/cndbpedia/intro/>

[**4**]  <http://kw.fudan.edu.cn/>

[**5**]  <http://kw.fudan.edu.cn/apis/cndbpedia/>

[**6**]  <https://zh.moegirl.org/index.php?title=Template:%E6%B5%B7%E8%B4%BC%E7%8E%8B&action=edit>

[**7**]  <http://bbs.talkop.com/forum.php?mod=forumdisplay&fid=49&filter=typeid&typeid=145>

[**8**]  <http://www.jinglingbiaozhu.com/>

[**9**]  <https://jena.apache.org/>

[**10**]  王鑫,邹磊,王朝坤,彭鹏,冯志勇.知识图谱数据管理研究综述.软件学报,2019,30(7):2139−2174. <http://www.jos.org.cn/1000-9825/5841.htm>

[**11**]  <https://www.w3.org/TR/sparql11-query/>

[**12**]  <https://neo4j.com/>

[**13**]  <https://github.com/zjunlp/deepke>

[**14**]  <https://github.com/ymcui/Chinese-BERT-wwm>

[**15**]  <https://www.macalester.edu/~abeverid/thrones.html>

[**16**]  <https://www.maa.org/sites/default/files/pdf/Mathhorizons/NetworkofThrones.pdf>

[**17**]  <http://www.openkg.cn/tool/refo-kbqa>

[**18**]  <https://github.com/SimmerChan/KG-demo-for-movie>

[**19**]  <https://github.com/machinalis/refo>

[**20**]  <https://github.com/Honlan/starwar-visualization>

[**21**]  <https://d3js.org/>

# 2 开发环境

问答系统的前端开发，即问答页面架构设计。主要是由与交互设计，视觉设计的配合，根据效果图来规划页面布局，合理部署页面代码层次，挖掘用户体验效果。用CSS，JS，antd等布局网页内容，制作静态页面，这些页面要兼容各主流浏览器，并配合程序完成静态页面与后台程序的整合工作。整个问答系统开发的流程在后面有具体介绍。

## 2.1开发工具

Visual Studio Code（以下简称vscode）是一个轻量且强大的跨平台开源代码编辑器（IDE），支持Windows，OSX和Linux。内置JavaScript、TypeScript和Node.js支持，而且拥有丰富的插件生态系统，可通过安装插件来支持C++、C#、Python、PHP等其他语言。

# 3 模型介绍

output，最可能的标注序列

CRF层

Bilstm层

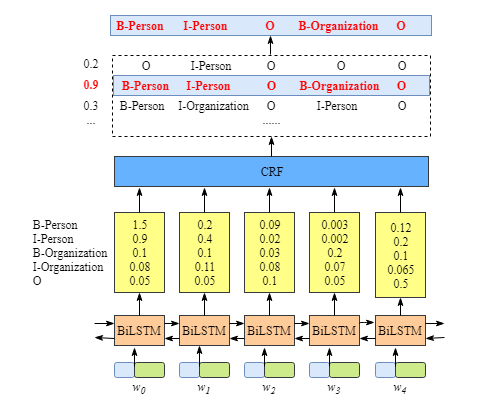
字向量

bert模型

question

## 3.1 BiLSTM + CRF

大家都知道BiLSTM可以通过cell保留上文的记忆，学习到上下文的信息，但是其输出互相之间并没有影响，它只是在每一步挑选出一个最大概率值的label输出。这样就会导致如B-person后再接一个B-person的问题。而crf中有转移特征，即它会考虑输出label之间的顺序性。对于每一个输入X=(x\_1,x\_2,…,x\_n),我们得到一个预测Label序列Y=(y\_1,y\_2,…,y\_n), 定义一个得分函数S(X,Y) 。得分函数S就很好地弥补了传统BiLSTM的不足，因为我们当一个预测序列得分很高时，并不是各个位置都是softmax输出最大概率值对应的label，还要考虑前面转移概率相加最大，即还要符合输出规则（B后面不能再跟B），比如假设BiLSTM输出的最有可能序列为BBIBIOOO，那么因为我们的转移概率矩阵中B->B的概率很小甚至为负，那么根据s得分，这种序列不会得到最高的分数，即就不是我们想要的序列。

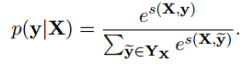


所以在BiLSTM+CRF中需要训练的参数为：

1. BiLSTM中的参数

2. 转移概率矩阵A

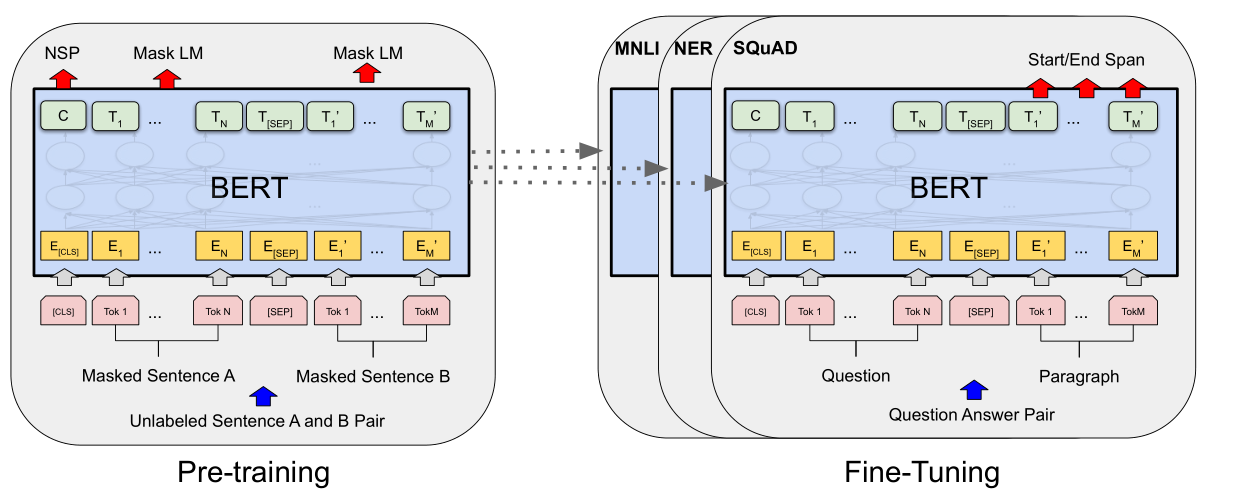
对于每个训练样本Y，求出所有可能的标注序列y的得分S(X,Y),对得分进行归一化。



那么我们的目标就是最大化上式（即真实标记应该对应最大概率值），因为叫损失函数，所以我们也可以对上式取负并去对数然后最小化之，这样我们就可以使用梯度下降等优化方法来求解参数。在这个过程中，我们要最大化真实标记序列的概率，也就训练了转移概率矩阵A和BiLSTM中的参数。

## 3.2 Bert模型

下面简要介绍一下BERT模型，本质可以把其看做是新的word2Vec，对于现有的任务，只需把BERT的输出看做是word2vec，在其之上建立自己的模型即可了。



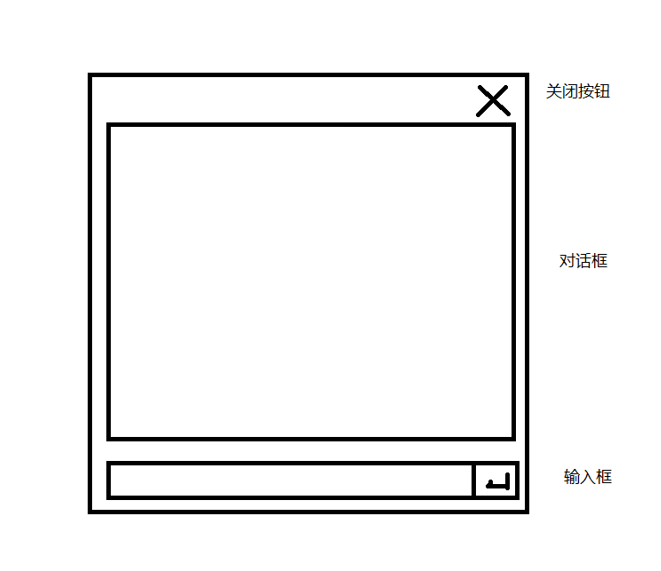
我们使用的是2018年Google开源发布的bert中文词向量预训练模型，然后在此基础上接入LSTM+CRF进行进行Fine-Tuning 。 question通过bert源码中提供的模块将句子拆分成字，并且将字映射成id，并通过bert模型把字映射成词向量，向量的维度为768.

在单纯使用BiLSTM + CRF对命名实体识别的效果就已经不错，在此基础上融合上Bert模型简直就是如虎添翼。

# 4 问答界面

由于问答机器人场景使用很多，大部分都需要前后端通信，前端使用了react框架， ui 使用了 antd组件库，这样很多人能看得懂代码并且方便定制修改。几乎所有样式都暴露出来，可以直接进行修改样式。antd 配置项也几乎全部暴露出来，满足各种特殊需要。

页面主要有两个面部分组成，首先是服务按钮。点击服务按钮，会出现对话框，对话框有输入框和对应的发送按钮，展示框和关闭按钮。



功能模块：

服务：显示对话框

输入框：输入文本，点击发送，文字显示在对话框，并且向后端发送网络请求，将问句发到后端。

对话框：展示输入框的内容，展示后端返回的请求中的答案。

关闭按钮：关闭对话框

# 5 页面布局设计

服务按钮的位置：位置用fixed布局（元素会被移出正常文档流，并不为元素预留空间，而是通过指定元素相对于屏幕视口（viewport）的位置来指定元素位置。元素的位置在屏幕滚动时不会改变。打印时，元素会出现在的每页的固定位置。fixed 属性会创建新的层叠上下文。），right设置为10px，top为40%。

对话框的位置：位置为水平垂直居中。这里用flex布局实现，设置元素的display属性为flex，justify-content的值为center; align-items的值为center。



## 5.1 组件设计

组件采用antd组件库的组件

服务按钮用一个button组件，设置点击事件，将对话框显示出来。

对话框用Modal（需要用户处理事务，又不希望跳转页面以致打断工作流程时，可以使用 Modal 在当前页面正中打开一个浮层，承载相应的操作。）

输入框用Input（）负责处理文文本的输入。

对话内容框为Avatar+reactNode的文本。

## 5.2自定义hooks

这个是自定义钩子，也是主体部分，传入参数，以及返回格式：

export function useRegister(

    //modal状态,只有开启状态才能开启Modal

    state: boolean,

    //获取用户回话的回调，用户输入会通过callback传回

    callback?: (v: RenderList) => void,

    //这个是antd的modal属性，参考antd官网

    modalOption?: ModalProps,

    //这个是input属性，参考antd官网

    inputOption?: InputProps,

    //这个是机器人语句，就是第一次打开后机器人发的语句

    initWelcome?: ReactNode,

    //这个是初始值，如果需要持久化可以考虑使用

    initState?: RenderList[],

): [ReactNode, React.Dispatch<React.SetStateAction<RenderList[]>>];

返回里可以拿到 render 渲染结果以及 setList 来设置对话框中的聊天语句。

值得注意的是 callback 设置语句请使用 useCallback 将函数作为常驻变量，否则用户每次输入都会触发。

## 5.3网络请求设计

前端使用axios（Axios 是一个基于 promise 的 HTTP 库，可以用在浏览器和 node.js 中）来发送网络请求，发送请求完成后，从接受到的报文中显示相应的答案。

# 6程序(智能问答)设计说明

## 6.1 程序描述

主要流程为：使用Bert + BiLSTM + CRF模型解析输入的自然语言问句生成 SPARQL 查询，在Neo4j图数据库中进行检索, 得到结果并返回。

## 6.2 功能

对医疗问题进行回答。

## 6.3 性能

机器人回复需要2秒时间

## 6.4 输入项

常规问句

## 6.5 输出项

知识图谱问题答案

## 6.6 算法

无

## 6.7 接口

## 6.8 存储分配

## 6.9 注释设计

对每一次判断都写出了详细的注释。

## 6.10 限制条件

该程序没有限制条件。

## 6.11 测试计划

根据不同的输入，将机器人的回复通过bleu来判断是否性能如何