编号：202210213010

哈尔滨工业大学
大学生创新创业训练计划项目验收书

项目名称： 基于 NLP 的智能校园通问答助手“小哈”的设计与实现

项目级别： 国家级

执行时间： 2021 年 9 月 至 2022 年 9 月

负 责 人： 敬刘畅 学 号 ： 120L021902

联系电话： 18030999650 电子邮箱： 120l021902@stu.hit.edu.cn

院 系 及 专 业： 计算机科学与技术学院人工智能专业

指导教师： 张宇 职 称 ： 教授

联系电话： 15084667669 电子邮箱： zhangyu@ir.hit.edu.cn

院 系 及 专 业： 计算机科学与技术学院

哈尔滨工业大学本科生院

填表日期：2022 年 9 月 5 日

一、课题组成员：（包括项目负责人、按顺序）

姓名	性别	所在院	年级	学号	身份证号	本人签字
敬刘畅	男	计算机科学与技术学院	2020 级	120L021902	51078120011006413X	敬刘畅
程钰莹	女	数学类	2020 级	1201230111	340204200204132621	程钰莹
刘议骏	男	未来技术学院	2020 级	7203610630	230104200111160219	刘议骏
唐浩程	男	未来技术学院	2020 级	120L010821	230103200208105113	唐浩程

二、指导教师意见：

项目按预定计划完成，实现了立项时设定的目标。在项目执行过程中，项目组成员分工明确、各个部分完成较好。结题报告撰写符合规范，同意结题！建议后续进一步完善，真正做到实用。

签　名：　　2024年　月 9 日

三、学院专家组意见：

项目技术方案合理，实现了预期功能，提供了在线答题系统。综合了较前沿新技术，难度较大。

在知识库规模上建议进一步提升。

组长签名：　　（盖　章）

2024年 9月17日

四、项目成果：

（一）申请专利情况：

序号	专利名称	发明人	专利申请号	备注

注：请将专利申请书的电子版作为附件报送。

（二）发表论文情况：

序号	论文题目	作者	刊物名及期号	备注

注：请将所发表论文及当期刊物封皮、目录的电子版作为附件报送。

（三）其它成果（软件、模型、图纸或作品等）：

序号	名称	说明
1、	小哈同学	基于项目问答系统实现的聊天窗口形式的问答网站

五、项目研究结题报告

1、 课题研究目的

针对当下校园环境中存在的校园信息及时问答、学校政策快速查询和已有分散信息资源整合等应用缺口和热点问题，项目面向新生、学生家长以及所有希望了解哈工大的社会人士开发了一款智能校园通问答助手“小哈”。“小哈”作为一款问答系统，可以满足：1、回答新生和家长因为对学校不熟悉而产生的一系列常问问题；2、回答在校生学习生活中遇到的常问问题；3、回答志愿填报过程中学生、家长针对招生政策的问题；4、回答针对教师信息的检索问题等等。

通过本项目的实现，可以让新生更快地融入到哈工大这个新环境中，方便学生更快速地找到教室、研讨室、实验室、了解优秀导师等等。同时，本项目也可以向学生家长、高三学生等社会各界想要了解哈工大相关内容的人士提供介绍哈工大的服务，通过“小哈”，大家能了解到哈工大的悠久历史、教育成果、师资力量、招生政策等等。

2、 课题背景

2.1 研究现状与趋势

问答系统是自然语言处理领域的重要技术应用输出口之一，自 1960 年左右国外的科学家提出希望计算机能运用自然语言处理来解决人们的问题开始，问答领域的技术就随着自然语言处理的技术革新而不断变化着。从 1950~1990 的小规模专家知识，到 1990~2010 的统计机器学习模型，再到 2010~2017 的深度学习模型，最后到近年的预训练模型，问答领域的相关技术一共经历了 3 次大规模的革新。在可预见的未来，像 GPT 这样的大规模预训练还将在一段时间内继续给相关领域带来更多更好的效果。同时，在应用领域，多模态场景的相关技术创新也将是研究趋势之一。

2.2 研究价值

当下，信息检索技术已经十分成熟，例如百度和 Google 搜索引擎，但是传统的信息检索会返回大量信息，造成信息过载，很多时候用户都会对返回的大量信息感到迷茫。同时在新生入学的情境下，由于学校建设、网络上信息滞后等多方面因素影响， 针对学校进行检索所返回的很多信息并不是最新的，有的还可能是错误的。而在这样的场景下，问答系统恰恰能够解决以上问题，人们可以用自然语言向问答系统提交问题，与传统检索不同的是，问答系统能够返回一个精准、精炼的回答而非整个网页集合。因此，本项目面向这些应用缺口具有应用意义和研究价值。

3、 课题研究主要内容

项目的整体框架由 3 个部分构成，分别是 FAQ 模块、抽取式问答模块和知识图谱问答模块，分别针对性解决校园常问问题问答、招生政策文档集问答和教师信息检索问题。在进行问答前，首先对问题经过分词、去停用词、同义词拓展等预处理后由文本分类模块将用户意图进行分类，根据分类将问题输入不同的模块中。系统框图如图 3.1 所示。

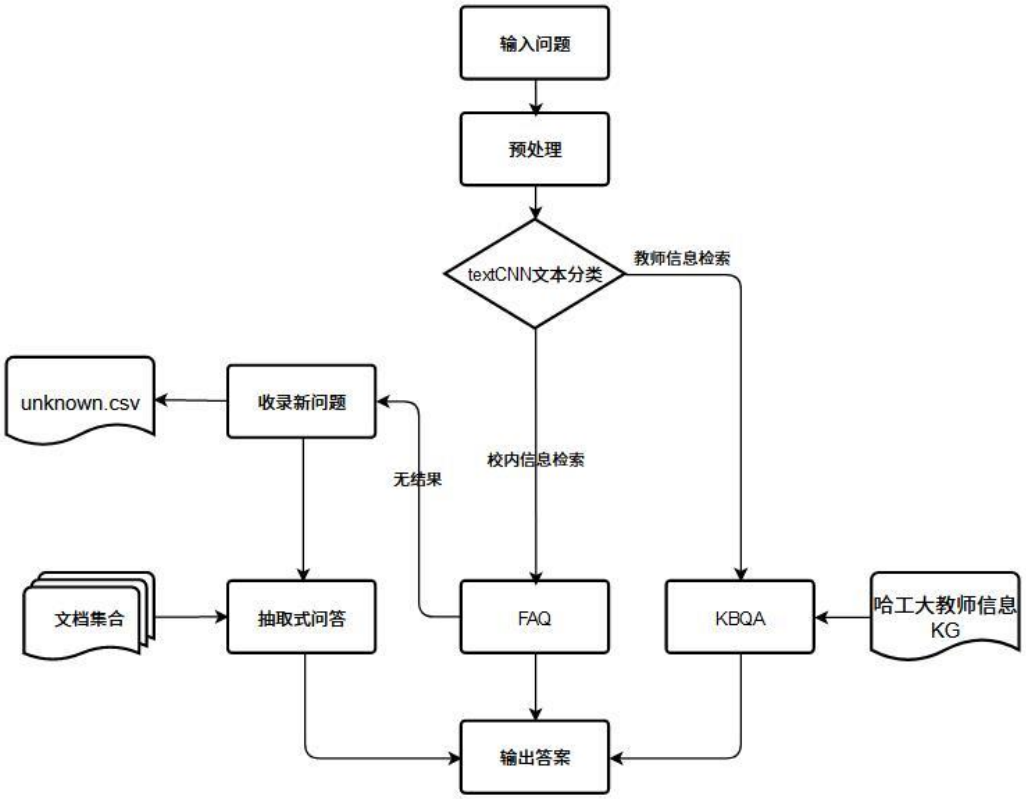

图 3.1 系统框架图

3.1FAQ 模块

FAQ 模块的方案为检索式问答，语料库来自我们根据《新生手册》、《哈尔滨工业大学报考指南》、网络资源以及“小哈”在 2022 年 7、8 月迎新活动中收集、整合、提取的 700 余条问答对，方案框架如图 3.2 所示。对用户输入的问题，首先做预处理，包括分词和去停用词。接着将预处理后的结果作为 key 值进入倒排检索系统中进行粗排序，其中倒

排检索的倒排索引表由已知的常问问题-答案对提前建立。然后将倒排索引结果中匹配度得分前 5 的 QA 对作为候选答案。最后，将候选 QA 对中的“问题”做预处理后借助 word2vec 训练得到的词嵌入向量经过 SIF 算法得到“问题”句向量，同样方法得到用户问题的句向量，计算两者的余弦相似度后进行精排序输出得分最高的“问题”对应的“答案”作为最终答案，同时将得分第二、三高的答案作为参考，输出其编号，允许用户直接点击或者输入其编号得到答案。

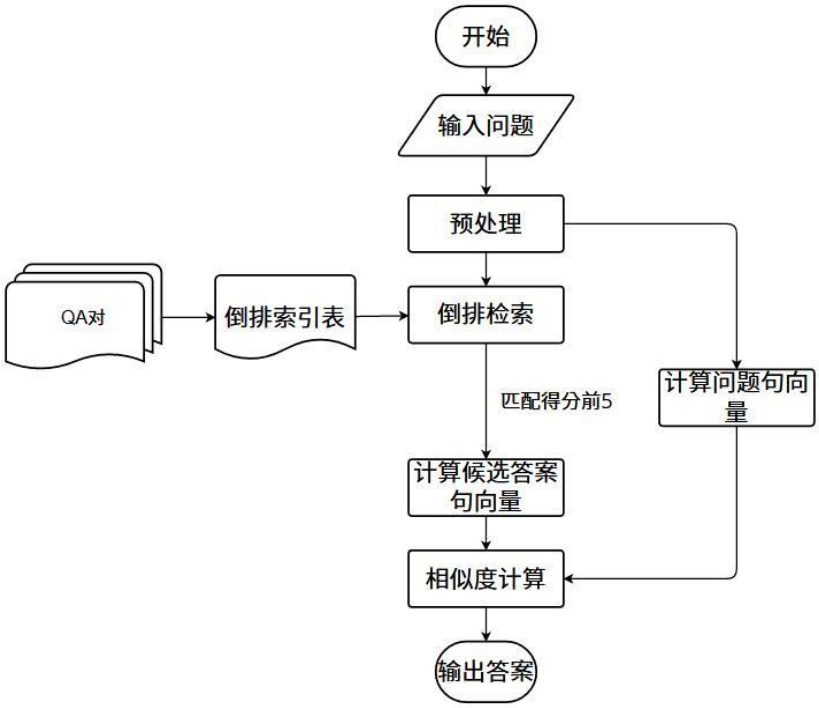

图 3.2 FAQ 模块流程图

3.1.1 文本预处理

文本预处理步骤包括分词、去停用词、同义词拓展三个部分。

1.分词

分词部分使用了 jieba 分词库来完成。jieba 库是一个简单实用的中文自然语言处理分词库，属于概率语言模型分词，其分词任务是：在全切分所得的所有结果中求某个切分方案 S，使得 P(S)最大。

Jieba 分词的原理为：

- 从自带的统计词典中构建前缀词典。例如：“哈工大”的前缀为“哈”、“哈工”、“哈工大”。
- 构建有向无环图。基于前缀词典实现高效的词图扫描，生成句子中汉字所有可能成词情况所构成的有向无环图 (DAG)。
- 计算最大概率路径。DAG 中的不同路径代表不同的分词方式，因此需要计算最大概率路径。在根据词频对 DAG 中节点进行赋权后，采用动态规划计算最大概率路径。
- 使用 Viterbi 算法解基于汉字成词能力的 HMM 模型处理未登录词：HMM 模型将中文词汇按照 BEMS 四个状态来标记，B 是开始 begin 位置，E 是 end 结束位置，M 是 middle 中间位置，S 是 singgle 单独成词的位置，jieba 采用(B,E,M,S)这四种状态来标记中文词语。HMM 模型将分词问题视为一个序列标注问题，其中，句子为观测序列，分词结果为状态序列。首先通过语料训练出 HMM 相关的模型，然后利用 Viterbi 算法进行求解，最终得到最优的状态序列，然后再根据状态序列，输出分词结果。

分词模块对“南京市长江大桥”的分词结果如图 3.3 所示：

```
Loading model from cache /tmp/jieba.cache
Loading model cost 1.141 seconds.
Prefix dict has been built successfully.
['南京市', '长江大桥']
<class 'list'>
```

图 3.3 分词效果图

2.去停用词

去停用词部分使用了“哈工大停用词表”，将停用词表保存为词典形式，然后将分词结果到词典中搜索，将匹配的词从分词结果中删除。

3.同义词拓展

与去停用词相似，同义词拓展部分使用了“哈工大信息检索中心同义词词林拓展版”，将同义词表保存为词典形式，然后将分词结果到词典中搜索，将匹配的词添加到分词结果中。

```python
def synonym_extend(cuted):
    extend_cut=[]
    for word in cuted:
        if word in synonym_dictionary.keys():
            extend_cut+=synonym_dictionary[word]
    extend_cut+=cuted
    return extend_cut
```

图 3.4 同义词拓展

3.1.2 倒排检索

倒排索引源于实际应用中需要根据属性的值来查找记录。这种索引表中的每一项都包括一个属性值和具有该属性值的各记录的地址。由于不是由记录来确定属性值，而是由属性值来确定记录的位置，因而称为倒排索引。

倒排检索的原理分为两个部分。第一部分为构建倒排索引表：对原文档进行编号给出每个文档的文档 ID，然后对文档中数据进行分词，得到词条。对词条进行编号，以词条创建索引，然后记录下包含该词条的所有文档编号以及词频等其他信息。第二部分为依据索引表进行检索：对待检索数据进行分词，根据得到的词条到索引表中获得每个词条对应的文档编号，对文档编号取交集完成检索。

单词ID	单词	倒排列表（包含该单词的文档ID，DocID）
1	谷歌	0, 1, 2, 3, 4
2	地图	0, 1, 2, 3, 4
3	之父	0, 1, 3, 4
4	跳槽	0, 3
5	Facebook	0, 1, 2, 3, 4
6	加盟	1, 2, 4
7	创始人	2
8	拉斯	2, 4
9	离开	2
10	与	3
11	wave	3
12	项目	3
13	取消	3
14	有关	3
15	社交	4
16	网站	4

图 3.5 倒排索引

如图 3.6 所示，本项目利用倒排检索原理首先为项目语料库建立了附带词频信息的倒排索引表，然后在检索时利用“哈工大信息检索研究中心同义词词林扩展版”对用户问题分词结果进行同义词扩充，将扩充后的结果进行倒排检索，对检索后的结果利用基于词频统计的 IFIDF 算法选取出得分最高的前五个结果。

```python
candidate={}
cuted_query=cut.seq_seperate(query)
cuted_query=list(set(cuted_query))
cuted_query=synonym_extend(cuted_query)
for word in cuted_query:
    if(word in index_tf.keys()):
        for key in index_tf[word]:
            if not candidate.__contains__(key):
                candidate[key]=get_score(cuted_query,key)
print('第一次倒排召回',candidate)
if not candidate:
    ans=[]
    return ans
candidate=get_order_dict_N(candidate,5)
print('倒排前五: ',candidate)
```

图 3.6 倒排检索粗排序

3.1.3 句子相似度

句子相似度计算使用句向量模型、句向量模型将句子特征映射为一个高维向量，句子的相似程度越高则在句向量空间中两个句子的句向量越接近，这种“接近”可以用两向量的夹角的余弦值量化计算。句向量获取分为基于 word2vec 的词嵌入模型和 SIF 句向量算法两部分。首先将 wiki 中文语料库基于 word2vec 模型训练出中文词向量模型，然后将句子分词结果的各个词的词向量输入 SIF 算法中获取句子的句向量。

1.Word2vec 模型

Word2vec 模型来自于 NNLM 模型的训练副产品词嵌入模型。在 word2vec 模型中使用的训练模型分别是连续词袋模型 CBOW 和 Skip-gram 模型。其中，CBOW 模型是在已知词语 W(t)上下文 2n 个词语的基础上预测当前词 W(t)，训练目标是：

$$maximize\ P\ （w_t|w_{t-n},\dots,w_{t+n}）$$

而 Skip-gram 模型是根据词语 W(t)预测上下文 2n 个词语训练目标是：

$$maximize\ P\ （w_{t-n},\dots,w_{t+n}|w_t）$$

两种训练模型的体系结构如图 3.7 所示:

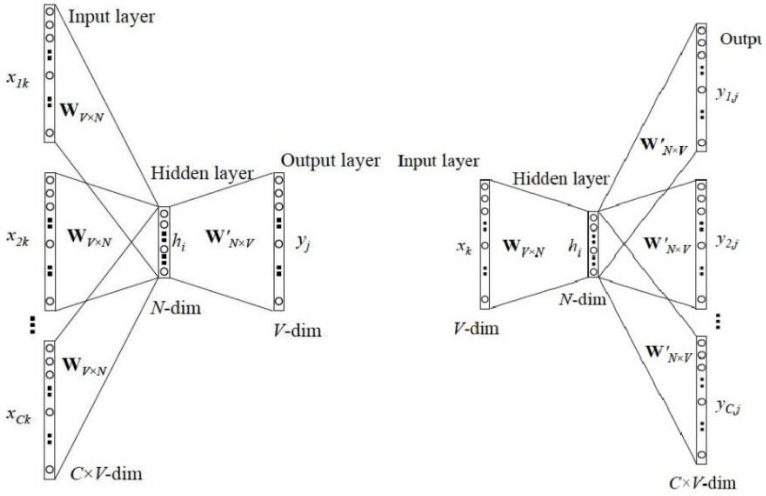

图 3.7 CBOW(左)Skip-gram(右)

项目中使用了 wiki 中文语料库和项目问答语料库作为 word2vec 词向量训练的数据来源，并将数据进行繁简转换、分词和去停用词等语料清洗。清洗完成后可以进行 word2vec 训练：

```
model=Word2Vec(LineSentence(infile),size=300,window=5,min_count=5,workers=multiprocessing.cpu_count(), iter=100)
```

2.SIF 句向量算法

SIF 算法通过词向量的加权平均来代表一个句向量，然后再用主成分分析的方法修改这些句向量。这种加权方法在文本相似度任务中的表现提高了 10%到 30%，并击败了包括 RNN 和 LSTM 的复杂的监督方法。

SIF 为句子中每一词的权重计算方法为：

$$Weight = \frac{a}{a + p(w)}$$

其中 a 是常数，p(w)代表词频。算得词权重后，对 word2vec 训练得到的词向量求加权平均后作为初步的句向量。然后将语料库中全部的初步句向量构成矩阵 X，最终的句向量为初步句向量减去初步句向量在矩阵 X 的第一主向量上的投影，即：

$$V = V - uu^TV$$

项目中，对于倒排检索返回的 5 个句子使用 SIF 算法中的最终句向量作为句向量表示，对于用户问题则使用 SIF 算法中的初步句向量作为句向量表示。

3.2 抽取式问答模块

抽取式问答可以根据用户提问在文本中抽取出对应用户问题的答案区间作为回答，项目以常见的学校政策文件集超过 3 万余字作为抽取式问答的文本库。在机器问答与训练模型中对于每一个问答数据样本都会有“context”，"question"和“answers”三个 key，模型通常将 question 和 context 拼接之后作为输入，然后让模型从 context 里寻找答案。项目分别尝试了 BiDAF 机器阅读理解模型和微调 RoBERTa 中文预训练模型的方法，并对两种方法的检索速度和检索效果进行了对比。结果表明，在速度近似的情况下，微调预训练模型返回的答案更加精确，同时，抽取式问答返回答案所需要的时间随着文档大小的增大而增长，因此项目将文本库拆分为相关度较高的 200 条小段落，在抽取式问答之前通过快速的倒排检索迅速缩小需要进行抽取的段落，并根据抽取后的评分来决定最终答案。同时将抽取式问答模块作为 FAQ 模块的补充模块，当 FAQ 无法回答用户提问时将询问用户是否花更长的时间使用抽取式问答模块进行问答。

3.2.1 微调预训练模型——预处理数据

微调预训练模型首先需要预处理数据，使用 tokenizer 对数据进行预处理，使得数据满足模型需要的输入格式。项目使用 AutoTokenizer.from_pretrained 实例化 tokenizer，它首先对输入进行 tokenize，然后将 tokens 转化为预模型中需要对应的 token ID，同时对于超出模型要求的最大输入长度的输入，还需要对输入进行切片操作。

3.2.2 微调预训练模型——微调模型

由于任务的不同，预训练的语言模型神经网络参数中会有一些被重新随机初始化，这些就是我们需要在新数据集上微调的参数。

```
def train(args, model, optimizer, scheduler, src_batch, seg_batch, start_position_batch, end_position_batch):
model.zero_grad()

src_batch = src_batch.to(args.device)
seg_batch = seg_batch.to(args.device)
start_position_batch = start_position_batch.to(args.device)
end_position_batch = end_position_batch.to(args.device)

loss, _, _ = model(src_batch, seg_batch, start_position_batch, end_position_batch)
if torch.cuda.device_count() > 1:
    loss = torch.mean(loss)

if args.fp16:
    with amp.scale_loss(loss, optimizer) as scaled_loss:
        scaled_loss.backward()
else:
    loss.backward()

optimizer.step()
scheduler.step()

return
```

图 3.8 微调模型

3.2.3BiDAF

BiDAF 模型共包含三大部分：嵌入层、注意力层和输出层，其结构如图所示。

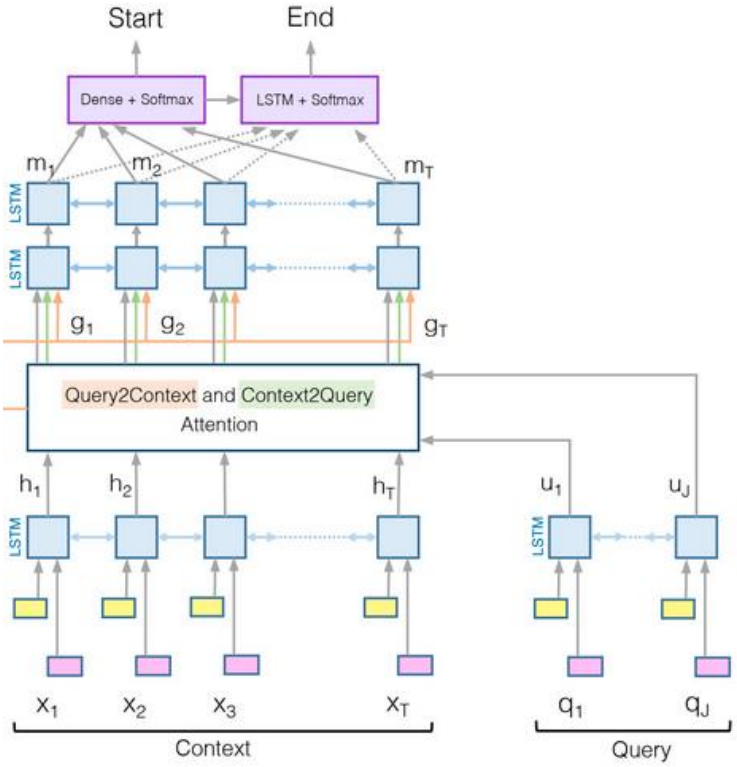

图 3.9 BiDAF

在嵌入层中共包含 Character Embedding Layer、Word Embedding Layer 和 Contextual Embedding Layer，前两层（Char-CNN 字符嵌入和 Glove 词嵌入）共同作为第三层（上下文嵌入）的输入，使用 BiLSTM 来生成上下文词向量，从中提取出序列信息，并将双向的词向量最终拼接到一起，以获取双向词向量信息，最后的词嵌入结果不仅包含语义信息还包含上下文信息。在注意力层中，根据嵌入层的结果矩阵 H 和 U 生成相似矩阵 S，矩阵 S 的 t 行和 j 列中的值表示示第 t 个上下文词(Context)和第 j 个查询词(Query)的相似性，其计算方式如下：

$$S_{tj} = \alpha\left(H_{:t}, U_{:j}\right) = \alpha(h,u) = W_{(S)}^{T}[h;u;h\circ u]$$

利用相似矩阵，分别做 Context-to-Query Attention 和 Query-to-Context Attention。注意力层后再经历 BiLSTM 层捕捉文本单词之间的联系，最后在 softmax 输出层中找到答案区间的开始和结束位置完成端到端的问答。

3.3 知识图谱问答模块

知识图谱问答模块主要分 4 步进行，方案框架如图 3.3 所示。

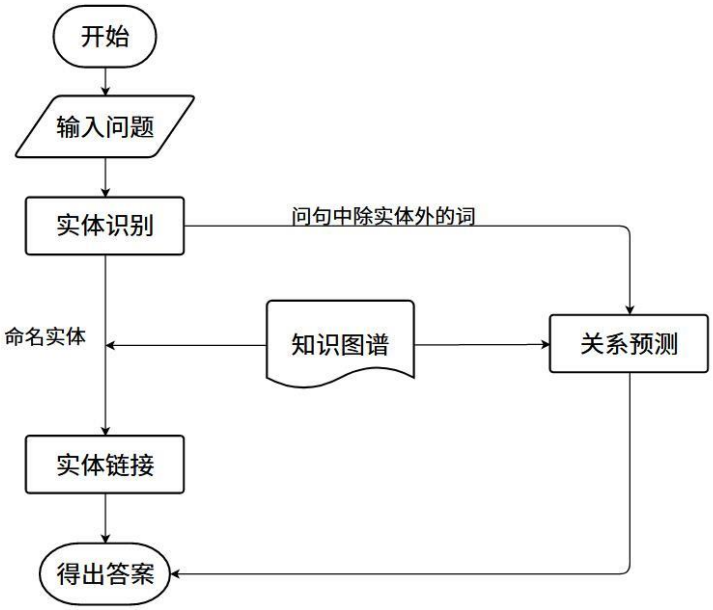

图 3.10 KBQA 流程图

3.3.1 知识图谱构建

知识图谱本质上是语义网络，是一种基于图的数据结构。项目采用 neo4j 以图数据库的形式依托“哈工大教师主页”的信息，为 3000 多名教师搭建了知识图谱，如图 3.9 所示：

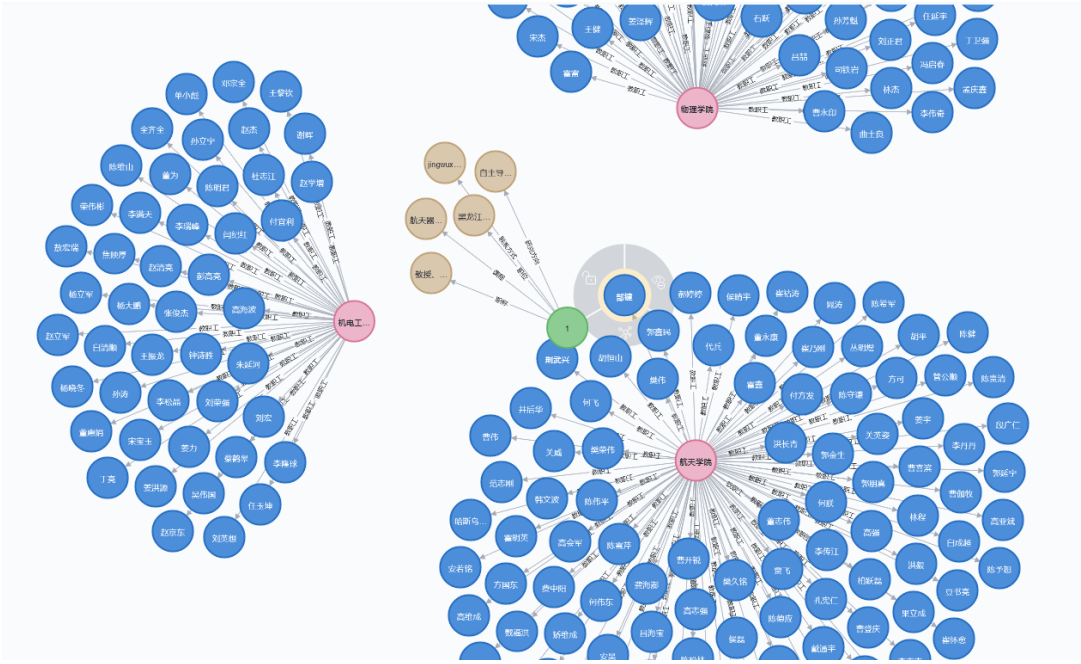

图 3.11 教师信息知识图谱

3.3.1 实体识别

将问题中关心的主要实体从问题中抽取出来，如图，项目使用 BiGRU+CRF 模型进行实体识别。

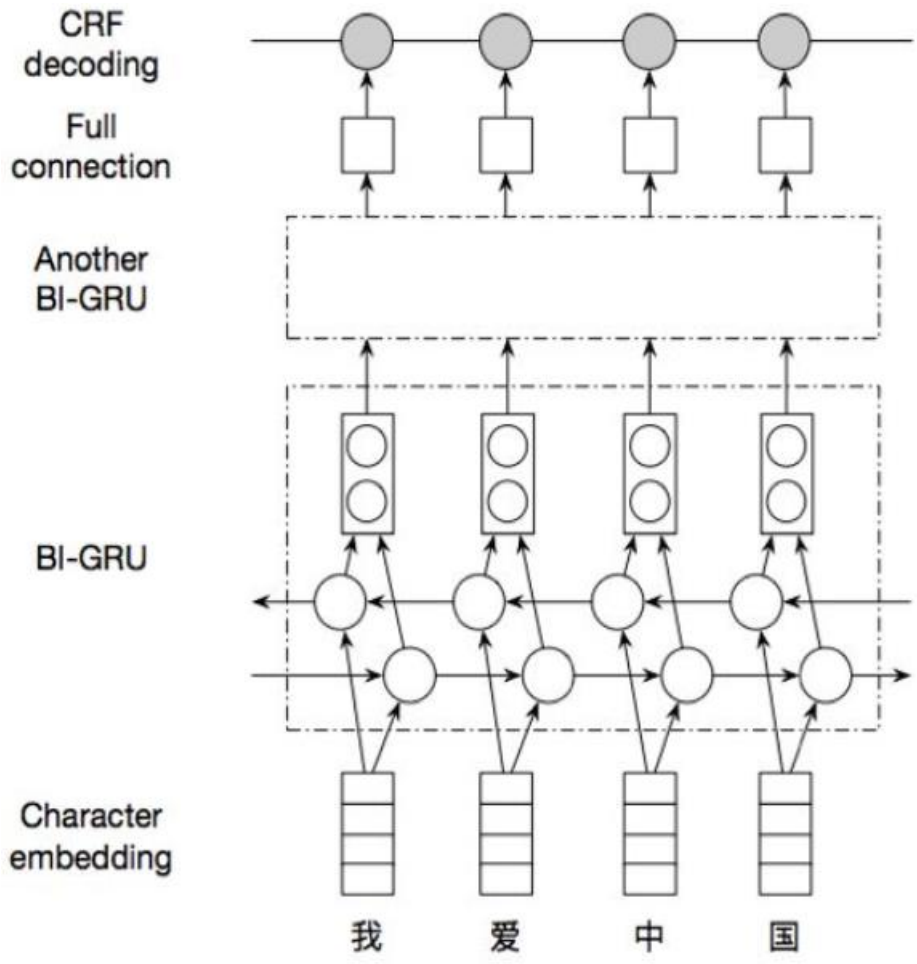

图 3.12 BiGRU+CRF 模型图

在 GRU 层，输入层为预先训练得到的词向量，然后在 GRU 层中进行 encode，之后接入全连接层中进行 softmax 分类，分类的 label 包括 I-Organization 、I-Person 、O、B-Organization 、I-Person 五类，“B” 即实体名的开始单词，“I” 为实体名的中间单词（或结尾词），“O” 为不是实体名的单词。

在 CRF 层，通过统计 label 直接的转移概率对 BiGRU 的结果加以限制，改善最终的输出结果，例如 I 标签后面不能接 O 标签，B 标签后面不能接 B 标签。

3.3.2 关系预测

从原句中除去实体之外的其他内容预测出，应该到关系库中寻找何种关系，如图，计划采用 seq2seq 模型的 encode-decode 结构。

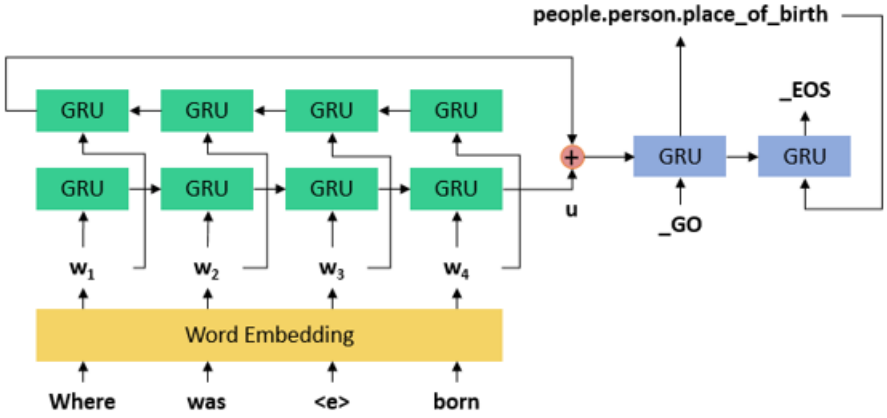

图 3.13 seq2seq 模型

在 encode 部分，如图所示采用双向 GRU 模型，将除去实体以外的句子成分输入双向 GRU 网络中，GRU 网络将句子 encode 成向量结构，而 decode 的单向 GRU 则将此向量结构 decode 成关系序列完成关系预测。

```python
class EncoderRNN(nn.Module):
    def __init__(self, config):
        super(EncoderRNN, self).__init__()
        self.input_size = config.source_vocab_size
        self.hidden_size = config.hidden_size
        self.num_layers = 1
        self.dropout = 0.1

        self.embedding = nn.Embedding(self.input_size, self.hidden_size)
        self.gru = nn.GRU(self.hidden_size, self.hidden_size, self.num_layers, dropout=self.dropout, bidirectional=True)
```

图 3.11BiGRUencoder

```python
class DecoderRNN(nn.Module):
    def __init__(self, config):
        super(DecoderRNN, self).__init__()
        # Define parameters
        self.hidden_size = config.hidden_size
        self.output_size = config.target_vocab_size
        self.num_layers = 1
        self.dropout_p = 0.1
        # Define layers
        self.embedding = nn.Embedding(self.output_size, self.hidden_size)
        self.dropout = nn.Dropout(self.dropout_p)
        self.gru = nn.GRU(self.hidden_size, self.hidden_size, self.num_layers, dropout=self.dropout_p)
        self.out = nn.Linear(self.hidden_size, self.output_size)
```

图 3.14 GRUdecoder

3.3.3 实体链接

注意到同一个实体可能有不同名字即需要实体对齐，或者同一个名字可能代表不同的实体即需要实体消歧。所以，只知道实体的名字实际上是不够的，而需要把抽取出来的实体名和知识库中具体的实体 id 对应起来。通常通过先收集知识图谱中所有实体的名称，然后构建单词到实体 id 的反向 map 表的方法完成实体链接。最后根据实体链接得到的实体 id 和关系预测中得到的关系相结合从知识图谱中检索出三元组。

4、 结论（成果介绍）

4.1 校园场景 FAQ 语料库

通过相关文件解读、整合网络资源和实地采集等等方法，我们构建了工大迎新、生活、学习等等场景下学生们经常会遇到的常问问题集，共 700 余条，部分问答对如下图所示：

```
[ "校园卡怎么充值啊？ ","可以在哈工大APP或者HIT校园卡公众号里进行充值。" ],
[ "饭卡怎么充值啊？ ","可以在哈工大APP或者HIT校园卡公众号里进行充值。" ],
[ "怎么查看自己的校园卡余额呢？ ","可以在哈工大APP或者HIT校园卡公众号里查看校园卡余额。" ],
[ "怎么查看自己的饭卡余额呢？ ","可以在哈工大APP或者HIT校园卡公众号里查看校园卡余额。" ],
[ "校园卡丢失了怎么办？ ","如果校园卡搞丢了，要先到图书馆办理挂失，然后带着挂失单到师生服务大厅缴费15元办理新卡。" ],
[ "饭卡丢失了怎么办？ ","如果校园卡搞丢了，要先到图书馆办理挂失，然后带着挂失单到师生服务大厅缴费15元办理新卡。" ],
[ "校园卡损坏了怎么办？ ","校园卡损坏了可以携带旧卡到师生服务大厅免费办理新卡。" ],
[ "学生证丢失了怎么补办啊？ ","如果学生证丢失可以集中在15教学周左右到院系教学管理办公室办理，补办费30元。" ],
[ "学生证损坏了怎么办？ ","可以集中在15教学周左右到院系教学管理办公室办理更换，换发学生证10元，只换芯片的话5元。" ],
[ "保卫处电话是什么？ ","一区保卫处电话：86412110；二区保卫处电话：86283110；科学园保卫处电话：86402110。" ],
```

图 4.1 部分 FAQ 问答对

4.2 基于倒排检索和词向量的 FAQ 问答系统

项目首先使用倒排检索系统进行粗排序，找出最可能成为最终答案的 6 个候选，接着使用预先训练的词向量经过 SIF 算法给出各句的句向量，采用余弦相关度句子相似度算法进行精排序，选出得分最高的一个答案，从而实现了 FAQ 的全部功能。一次 FAQ 问答如图所示：

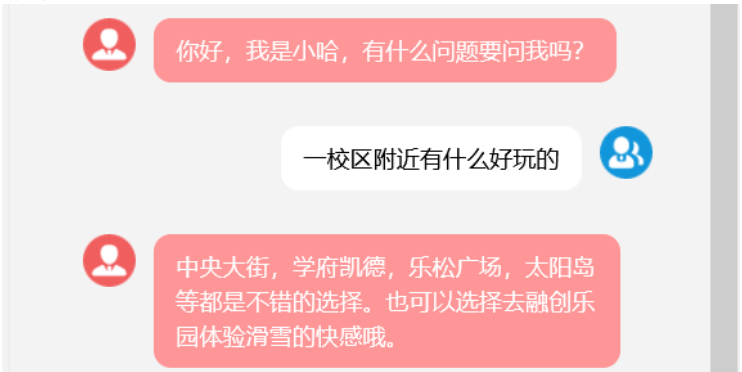

图 4.2 FAQ 示例

4.3 基于 RoBERTa 中文预训练模型的抽取式问答模块

考虑到抽取式问答需要时间较长，我们将抽取式问答模块作为 FAQ 模块的补充，在 FAQ 遇见不在问答库中无法回答的问题时，询问是否使用抽取式问答，用户肯定后，再调用抽取式问答模块进行回答。一次抽取式问答如图所示：

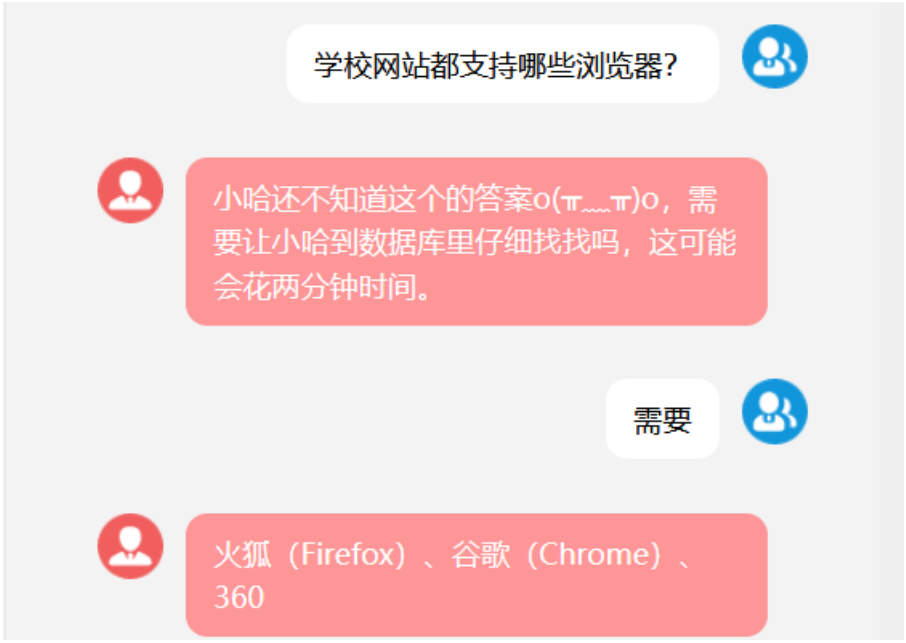

图 4.3 抽取式问答

4.4 KBQA 模块

在构建了全校教师信息知识图谱的基础上，我们完成了知识图谱问答模块的搭建。一次知识图谱问答如图所示：

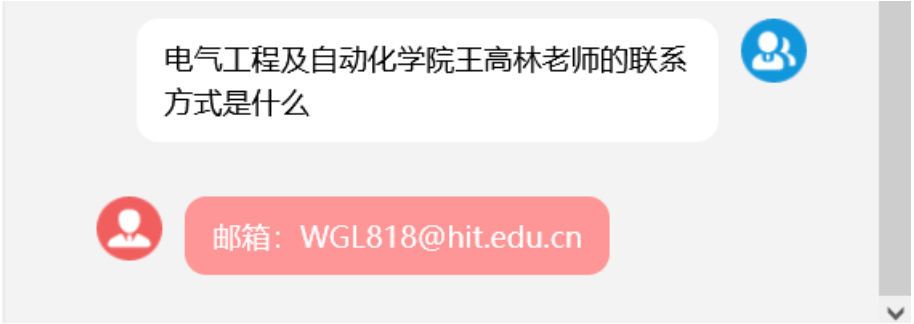

图 4.4 KBQA

4.5 “小哈”的网页 UI 设计与发布

为了方便用户的使用，项目设计了“小哈”的用户界面 UI 设计，设计了类似聊天窗口的界面，并基于 flask、nginx、gunicorn 和 supervisor 发布在 http://121.37.66.61/chat，用户界面如图所示。

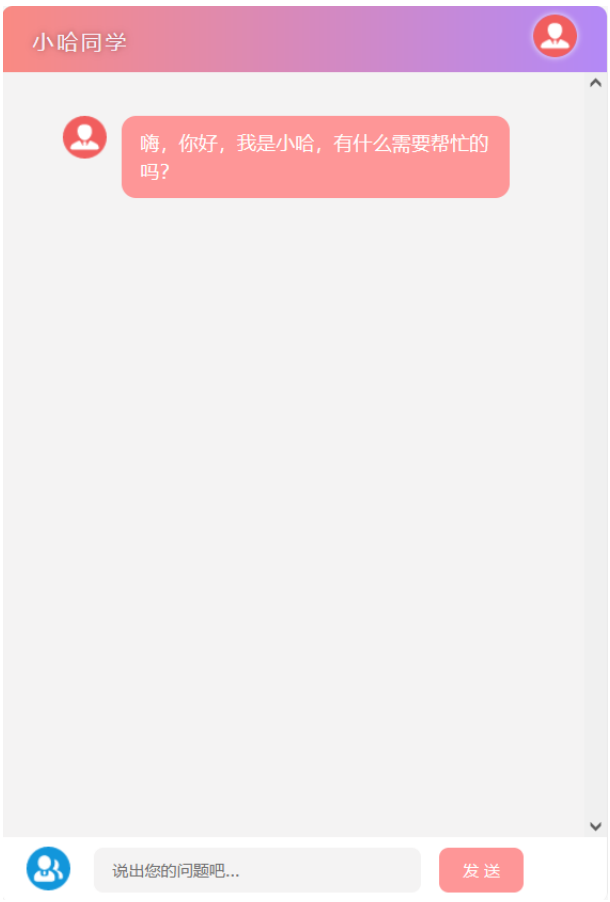

图 4.5

5、 经费使用情况

表 1 经费使用

书籍购买	1000 余元
云服务器租赁	396 元
GPU 算力租赁	500 元
训练内存	738 元

6、 问题、体会与收获

6.1 深入理解 NLP 技术

在项目各模块实施的调研阶段，项目小组接触了许多 NLP 领域的相关理论算法和实践技术路线，对整个 NLP 技术的技术发展历程有了一个相对清晰的认识，尤其是项目相关的技术的发展历程，也对当下热门的 NLP 技术有了一些浅层的接触。同时，对项目涉及到的部分算法和技术进行了实践和效果评估，从效果评估中验证了不同模型特长和缺点。

6.2 深入理解深度学习技术

在项目各个模块中几乎都涉及到了神经网络的训练,例如:FAQ 中做 word-embedding 时使用的 word2vec 和 fasttext 模型、抽取式问答中的预训练模型和 KBQA 中的意图识别模型都是深度学习的神经网络模型。为了实现这些模型，项目小组较为系统地学习了深度学习的一些基础内容，并对以上模型进行了实践。

6.3 技术实践体会

项目实施过程中，项目小组体会到经过长时间的发展，NLP 领域针对问答系统的技术、算法已经非常的丰富，许多模型都可以满足同一个任务的实现。但是，除了模型本身的优点和缺陷以外，模型是否能够很好地完成任务要求还取决于模型本身的训练方式、预先为模型准备的相关工作等等许多因素，很多时候仅仅从理论上分析模型是不够的，还需要实践模型来进行评估。同时，项目在搭建各种语料库的过程中也深刻体会到了一个好的数据集对其所对应任务的重大意义

7、建议

目前项目中的知识图谱模块中还存在着知识图谱不够标准、内容不够丰富等问题，在这样单薄的知识图谱中，我们发现基于一系列神经网络的 KBQA 甚至不如基于规则库的 KBQA 更快更准,希望接下来能进一步继续系统地丰富知识图谱，以取得更好的效果。

同时，我们还希望给“小哈”添加一些更加人性化的功能如：支持直接检索学校政策文件、根据每日访问量提供今日热点问题等等，并进一步优化已有模块性能。

8、结束语与致谢

项目能一点点从零开始找到实现整个系统的方向和技术路径都要感谢张宇老师的悉心指导和支持，同时也感谢项目组每个成员的付出，相信以此项目为引，我们一定能更加深入地探索问答系统。

9、参考文献

[1]. H., W. and L. X. Question Answering System with Enhancing Sentence Embedding. in 2022 11th International Conference of Information and Communication Technology (ICTech)). 2022.

[2]. J., Y. Research on Question Answering System Based on BERT Model. in 2022 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL & ICCEA). 2022.

[3]. R., E. and S. M. PeCoQ: A Dataset for Persian Complex Question Answering over Knowledge Graph. in 2020 11th International Conference on Information and Knowledge Technology (IKT). 2020.

[4]. Seo, M., et al., Bidirectional Attention Flow for Machine Comprehension. 2016. p. arXiv:1611.01603.

五、附件（专利、发表论文及其他成果支撑材料）