**1 Introduction**

本文想要实现一个可操作地，拥有技能匹配能力的Question&Answer 系统。

具体的功能可以归纳为，输入一个问题query，通过遍历已有资源（类似于问答对库，包含大量的问答对），通过将已有资源的问题与我们输入的query进行对比，找到最相近的问题，进行整理归纳，输出意思最相近的可靠回答。

现在市面主流的问答系统有三种，一是基于知识图谱的QA、二是基于阅读理解的QA、三是基于交互式的QA。

关于基于知识图谱的QA，其主要分为两种，分别是基于符号表示的KB-QA和基于向量表示的QA。对于第一种呢主要就是利用语义解析的方法对问题进行解析，将问题转化成逻辑表达式，再结合相应的规则，得到一个结构化的SQL查询语句，在查询知识库得到答案。

对于第二种（于向量表示的QA），就是直接进行向量点乘，利用CNN方式进行改进，通过训练相应模型，通过score进行筛选，取最高得分作为最终答案。对于以上方式做一个小小的总结，我认为可以改进的点有：1.有些模型只能解决简单回答，灵活性不够。2.对于开放域问答，单一的知识库不能完全回答所有问题。

关于基于阅读理解的QA，主要的过程就是对非结构化文章进行阅读理解得到答案，分成匹配时、抽取式和生成式。其中抽取式QA算是比较主流，那我们以它为例讨论一下。现在认同度极高的数据集应该就是斯坦福的SQuAD数据集了，数据形式就是给定一篇文章，然后围绕这个文章提出一些问题，接下来就是直接从文章中扣答案出来。一般情况下对于词进行embedder，然后对于文章以及问题用LSTM进行建模，进行交互。用query-aware的文章表示来预测答案，一般是用两个网络分别预测答案起止位置，或者直接对文章进行答案标注。

最后我们要介绍的，也是本文着重研究的QA方法，基于交互式的QA系统。

**2 Preliminaries(预赛)**

**3 Reliable Properties**

**4 Verification**

**5 Testing**

**6 Interpretability(解释性)**

**7 Future Challenges**

**8 Conclusions**