基于智能问答技术的交互式校园信息查询系统

信息科学技术学院 汤济之 王一雪

摘要：本文阐述了我们在本科生科研训练当中完成的校园问答系统的设计思想，该系统目前能够回答涉及院系、教师、课程这三个方面以及它们之间的各种问题。我们将收集到的数据组织成相互连接的图形式；通过依存句法分析将问句也转化成图的形式，并通过集束搜索的方法在知识图上匹配得到答案；我们也使用了索引等方法加速搜索匹配。我们的系统能够支持一定程度上的模糊匹配和长问句匹配，但是需要问句句式较为简单。在已有的数据上，我们的系统表现不错，能够满足一般性问题的要求。

引言

生活在大学校园里，我们每天都得和各种信息打交道：我们需要知道某节课的上课时间，我们需要知道某老师的邮箱，我们需要知道某个食堂什么时候关门……然而现实是我们往往得从不同的地方、通过不同的渠道得到这些信息。这不仅使得信息的获取变得困难，也浪费了我们的宝贵时间。我们设计并开发了一个问答系统，希望能解决这个问题，或者至少解决一部分。

我们的问答系统采用了较为传统的知识图模型和检索式架构。系统接受用户的自然语言输入，将非结构化的自然语言变成结构化的图形式，再到知识图谱中进行子图匹配。主要难点有两个：

1. 如何把自然语言语句编程结构化的图表示；
2. 如何在整个知识图谱中进行子图匹配。

解决第一个问题的思路大致有两种：

1. 通过手工定义模板生成结构。True Knowledge QA 系统[1]就是采用这种离线人工定义的模板，但是该种方法存在模板定义代价高、拓展性差等问题。
2. 通过语义关系抽取出结构。Lei Zhou[2]通过语法分析生成语法依赖树，从中抽取出多个语义关系对（三元组），并将三元组指向同一个实体的结点关联起来，形成一个查询图。

而第二个问题，即子图同构问题，本质上是一个 NPC 问题，暂时找不出多项式解。对于一般的子图，目前已有的算法，多使用“回溯树”搜索加剪枝的方法，

[1] Tunstall-Pedoe W. True knowledge: Open-domain question answering using structured knowledge and inference[J]. AI Magazine, 2010, 31(3): 80-92.

[2] Zou L, Huang R, Wang H, et al. Natural language question answering over RDF: a graph data driven approach[C]//Proceedings of the 2014 ACM SIGMOD international conference on Management of data. ACM, 2014: 313-324.

如 Ullmann 算法[3]和 VF2 算法[4]。

在这里，我们使用依存句法分析的方法完成从问句自然语言到图结构的转换，使用 beam search 对最常见的问题句型进行图匹配。

本文接下来的内容中，我们将从以下三个方面来介绍我们的系统：数据获取和预处理、问答系统模型以及结果展示。在问答系统模型一节里，我们从问句分析模块、匹配模块和加速方法三个方面详细地阐述了我们的模型设计。在最后的讨论一节中，我们简要介绍了目前模型需要改进的地方和我们其他的一些尝试。

数据获取和预处理

目前为止，我们获取了三个方面的数据，分别是学校的院系基本信息、部分教师的基本信息以及学校 16-17 学年第二学期的本科生课表信息。

学院的基本信息直接从北京大学官方网站[5]及各院系官方网站（从官网页面上可以进入）上获取，包括各院系的中英文名称等。

教师的信息从各院系的官方网站上获得。由于网站较多且其编写排版等都完全不同，故直接手动摘录。包括教师的姓名、邮箱、教授课程等信息，由于各个学院网站的编排不一样，不同学院的老师可能会提供不同的信息。

课程的信息从北京大学教务网[6]上获得，通过编写爬虫爬取，包括课程中英文名、上课时间、授课教师、学分等除了上课地点之外所有课程有关的信息。

原始的数据经过了如下的预处理：

- **数据扩充**：由于实际生活中学生老师对于一些长难名字的称呼往往是简称，比如用“信科”代替“信息科学技术学院”，因此我们对一些学院名和课程名手动标注了简称。另外，由于一些特定院系的课程简称只有特定院系的学生知道，获取这一部分信息是十分困难的。这里我们使用了近似的替代方法，将课程名分词，并将所有词的第一个字拼在一起形成简称，这样，除了少数具有特殊简称的课程，大部分课程的简称即使不完全正确也是差不多正确的。
- **数据格式转换**：原始数据中的一些是难以检索且难于表示的，比如原始数据中的上课时间就是完全通过一个课程表来表示，我们通过编写一些规则将这类数据全都转换成了自然语言描述，用字符串来表示。另外，对一些相同意思的不同表示也进行了归一化，比如“高数 b1”和“高数 Bi”统一表述成“高数 b1”。
- **词典生成**：由于专有名词较多，尤其是扩充了简称之后，直接使用现成的分词工具容易造成错误的分词。因此我们将所有课程、老师、以及学院的全称和简称都提取出来生成词典，帮助之后的分词更加精确。

[3] Ullmann J R. An algorithm for subgraph isomorphism[J]. Journal of the ACM (JACM), 1976, 23(1): 31-42.

[4] Cordella L P, Foggia P, Sansone C, et al. An improved algorithm for matching large graphs[C]//3rd IAPR-TC15 workshop on graph-based representations in pattern recognition. 2001: 149-159.

[5] http://www.pku.edu.cn/academics/index.htm

[6] http://dean.pku.edu.cn/pkudean/course/kcb.php?xnxq=16-17-2

知识库建立

我们采用了图数据库 Neo4j[7]来建立知识库。

相比于传统的关系型数据库如 MySQL,图数据库更加关心数据之间的联系，将整个知识体系通过知识结点与结点之间的联系构建成一张大图。在有关院系、教师、课程这三角的问题当中，经常会遇到这种关系的描述，比如“高等数学的老师是谁？”这样的问题。更常见的是，由于很多院系都开设了高等数学课程，为了问题表述准确，会加上开课院系来表明一个特定的课程，比如“信科的高等数学的老师是谁？”，又比如通过老师名来指定课程的，“XX 老师的高等数学什么时候上课？”。院系、老师、课程之间的关系在这里就变得非常重要。在描述这种关系的时候，图数据库如 Neo4j 比之于关系型数据库更加直接，使用起来更加简化。

问答系统模型

问答系统的模型主要分为两个部分，一部分是问句分析模块，另一部分是匹配模块。其中问句分析模块负责解析问句，将问句分析分解，逐步提取出需要在数据库中进行匹配的部分传递给匹配模块。而匹配模块负责两个层次的匹配，在高层次需要将整个问句和数据库中的一个子图进行匹配；在低层次，对于问句的每一个独立部分需要和图中的某个结点或者边进行匹配。

在这里我们只解决句法结构为单向链式或近似单向链式的这种句型较为简单的问句，单向链式结构的定义见问句分析模块一节。

1. 问句分析模块

问句分析模块主要采用了下面两种工具：哈工大 LTP[8]和 jieba 分词[9]，其中哈工大 LTP 用的是其 python 封装的离线版本 pyltp[10]。

我们使用依存句法来分析整个问句，一个典型的依存句法分析结果如图 1。

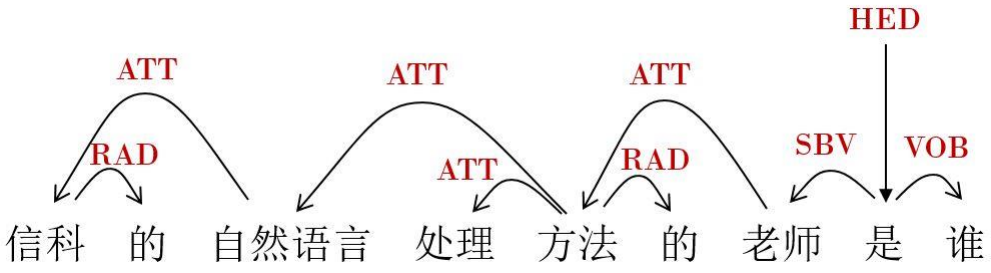

Figure 1.**依存句法分析**

依存句法主要分析了词在句子中的组成部分，是语法层面的分析。一般来说，相比于涉及语义的句法分析，如语义依存分析，依存句法分析因其更加简单、标注

[7] https://neo4j.com/
[8] https://www.ltp-cloud.com/
[9] https://github.com/fxsjy/jieba
[10] http://pyltp.readthedocs.io/zh_CN/develop/api.html

种类更少而正确率更高。表 1[11]展示了我们所用到了的几种最常见的依存句法标注的含义。

关系类型	符号	描述	例子
主谓关系	SBV	主语-动词	我送她一束花（我<-送）
动宾关系	VOB	直接宾语，动词-宾语	我送她一束花（送->花）
介宾关系	POB	介词-宾语	在贸易区内（在->内）
状中结构	ADV	副词	非常美丽（非常<-美丽）
定中关系	ATT	属性，形容词	红苹果（红<-苹果）
右附加关系	RAD	-	孩子们（孩子->们）
核心关系	HEAD	-	指整个句子的核心

Table 1. **依存句法分析基本标注类型**

我们只专注于解决具有单向链式结构或近似单向链式结构的句子。

单向链式结构，指的是若只保留核心谓语动词、所有名词及这些词之间关系，除两端结点一个只有出边一个只有入边外，其余结点都恰好只有一个出边一个入边的无环结构。图 1 中的句子，只保留核心谓语动词、所有名词及之间的关系，如下：

信科←自然语言←方法←老师←是→谁

这是一个链式结构，但不是一个单向链式结构，因为“是”字有两个出边。而“是”字左边，从“信科”到“老师”是一个单向链式结构。“是”字的右边只有一个结点，不存在结构关系，这种结构是我们所期望的，因为我们可以通过一些手段将整个句子变成一个具有单向链式结构的长词组。

一个具有更加复杂链式结构的问句的例子是：“信科的高数是什么学院的老师上的？”，经过依存句法分析后，其核心结构如下：

信科←高数←是→老师→学院

这里“是”字的两边都分别是一个单向链式结构，这是我们难以解决的。目前来看，要在图上匹配这种结构需要在图上同时不相关地匹配两个子结构，我们现在采用的方法的运算复杂度和时间复杂度都太高，无法满足实时回答的要求。当然，还存在更加复杂的结构，例如一个非链式的结构，这在使用多个名词修饰一个名词时出现。这些复杂结构需要更加强大的图匹配算法，另外也需要更加强大的句法分析器来保证句法分析本身的正确性。

事实上，在大多数情况下，人们提的问题都是更加直接简单的，即单向链式结构或之前提到的近似单向链式结构的问句占问题当中的大多数。

在“匹配模块”部分我们会看到，对于单向链式结构的名词性词组，我们有非常好的匹配方法。

整个句法分析过程分为两步：

第一步是简化问句结构，将问句还原成长的具有单向链式结构的名词性短语。若一个查询本身就是一个单向链式名词性短语，没有完整句子结构，即没有核心谓语动词，则直接跳过该步骤。一个完整的问句的骨干结构基本都是以下两种之一：

[11] 引自 LTP 官方文档 https://www.ltp-cloud.com/intro/

符号表示	解释	例子
SBV-HEAD-VOB	主谓宾结构	高数的老师是谁？
SBV-ADV-HEAD	主谓结构（状语一般包含疑问目标）	高数什么时候上课？

Table 2. **问句主干结构**

- 对于**主谓宾结构**，疑问部分一般在宾语之中，并且疑问词一般不提供任何非重复有效信息。比如表 2 当中的例子，在问句“高数的老师是谁？”中“高数的老师”已经具有了该问句所需要检索的全部信息，“谁”虽然提供了指代人物的信息，但是和之前的信息重复了。使用频率最高的疑问词“什么”就只表示疑问，并不提供任何与检索内容相关的信息。因此，我们直接去掉疑问词，将主语和宾语拼接在一起形成一个长名词词组，再重新对这个词组做依存句法分析。在上面例子中，“高数的老师是谁”，就变成了“高数的老师”。
- 对于**主谓结构**，可能涉及到用“在哪”询问地点，比如“高等数学在哪上课？”，其中“哪”字提供了其余部分没有涉及的检索信息，就是“地点”。因此，对“哪”，我们直接用词“地点”替换，其余疑问词直接去掉，然后将主语和中心谓语动词的状语直接拼接，形成长名词词组，重新做依存句法分析。上面例子中“高等数学在哪上课？”就变成了“高等数学地点上课”，这并不是一个完美的转换，但是之后的匹配机制能够让大部分这种情况匹配成功。

值得一提的是，上述转换能够使大部分问句在转换后成为一个单向链式结构，但并不能保证。

第二步是逐步提取出可能的词组部分，传给匹配模块到数据库中匹配。依存句法分析实际上将句子的各个部分组织成一棵树，而单向链式结构保证了除了在最底层，每层都有且只有一个非叶子结点。这样，句法树中每个以名词非叶子结点为根的子树都能表示一个可能存在的实体，并且不会有信息遗漏。以图 1 中问句为例，经过第一步变换之后，句子变成了“信科的自然语言处理方法的老师”，重新进行句法分析的结果和在原句中类似。取每个名词非叶子结点为根的子树，并去除标记为附加信息的词，如“RAD”标记的“的”，这样就得到“信科”、“信科的自然语言”、“信科的自然语言处理方法”、“信科的自然语言处理方法的老师”这四个名词短语，这四个名词短语即可进行下一阶段的匹配。

这样做的原因是，在课程和学院名称中，存在大量名词做修饰词的现象，如“数学物理方法”这门课三个词语都是名词，但“数学”和“物理”都是修饰词而不是中心词，这种情况在没有进行知识库匹配的情况下是无法分辨的。因此不能贪心地用名词一个个匹配，但也不能随意地进行划分。依存句法分析的树状结果为我们提供了很好的划分依据，我们可以在匹配的时候再决定哪种划分方法是最佳的。

2. 匹配模块

匹配分为两个层次，在低层次是一个名词短语和图中一个特定结点的匹配；高层次是一个句子的结构划分在图上的拓扑结构匹配。

在低层次的匹配中，我们得到从句法分析中获得的句子的一部分——一个名词性短语，和知识图中的某一个结点，或者是关系，或者是某一个结点的某个属

性，在这里为了表述方便，统一表述成结点。我们想要知道这个名词性短语是不是表示这个结点，通常使用某种相似度计算方法，分别计算这个短语和结点的全称、简称的相似度。计算了和很多结点的相似度之后，我们可以通过排序找到最可能的那个结点。给定名词短语 N，一个结点的名称 E，名词短语和结点名称的长度分别为 LN 和 LE，则它们的相似度计算公式如下：

$$Similarity(N,E) = 0.5 \times 归一化\mathit{编辑距离}(N,E) + 0.5 \times 覆盖率(N,E)$$

更精确的定义如下：

$$Similarity(N,E) = 0.5 \times (1 - \frac{\mathit{编辑距离}(N,E)}{\max(LN,\ LE)}) + 0.5 \times \frac{len(\mathit{最长公共子序列}(N,E))}{LN}$$

编辑距离和最长公共子序列的定义如下：

- **编辑距离**：两个字符串串之间，由一个转成另一个所需的最少编辑操作次数。许可的编辑操作包括将一个字符替换成另一个字符，插入一个字符，删除一个字符。
- **最长公共子序列**：一个子序列是一个字符串串随意地去掉若干字符剩下的序列，最长公共子序列是两个字符串公共的子序列中长度最长的一个。

在主要匹配类型为名称的匹配当中，顺序的一致是非常重要的，例如“生物化学”和“化学生物学”就是两门差异很大的课，因此在这种情况下一些不考虑顺序的方法如向量空间模型加余弦距离效果并不好。而编辑距离正是一种考虑顺序的匹配方法，又因为原始编辑距离和字符串本身长度有关，因此需要归一化。另外，由于提问者提问时有遗漏和简化倾向，而基本不会凭空添加或修改词语，比如对于课程“自然语言处理中的经验性方法”，提问可能基本都是简化的如“自然语言”、“自然语言处理经验”等。因此目标实体名对于提问者提问的短语的覆盖率也是很重要的因素，比如之前的例子中，若只考虑编辑距离的话，“自然语言”匹配“程序设计语言”的可能性比“自然语言处理中的经验方法”更高。在这里我们使用二者的最长公共子序列在原词组的长度占比表示覆盖率。编辑距离和最长公共子序列都可以用类似的动态规划算法计算，复杂度是$\mathrm{O(LN \cdot LE)}$，考虑到结点名称是固定的并且都不长（很少有超过 10 个字符的），可以看作常数，因此该算法计算复杂度变为$\mathrm{O(LN)}$。

另外一个要考虑的因素是相似性计算的粒度的问题，一般来说以词级别的粒度是最好的，因为词是正确表示语义的基本单位。但是在这里，大量专有名词和简称会导致分词工具出现无法正确有效分词的情况，因此字粒度也是必要的。

故综合上述，我们采用的相似度打分公式如下：

$$Similarity(N,E) = 0.5 \times SimilarityCharacter(N,E) + 0.5 \times SimilarityWord(N,E)$$

高层次的匹配是句子结构在图上的匹配。该类问题的本质是子图同构问题，而子图同构是著名的 NP-完全问题，目前不存在多项式时间的解。这里我们只关注单向链式结构在图上的匹配。

我们使用集束搜索(beam search)算法，实际上是广度优先搜索加上剪枝。定义搜索状态如右。包括三个部分，已匹配部分，记录了每一步匹配在待匹配词组中的部分、匹配的结点名称以及该步匹配的分数；未匹配部分保留了还未在图中进行匹配的词组剩余部分；总得分是从初始状态匹配到当前状态位置这一条路径上

已匹配部分：（高数，高数 B，0.7）
未匹配部分：老师
总得分：0.7

的总得分，在这里总得分通过计算路径上每一步匹配的平均得分获得。在初始状态中，只有未匹配部分为完整待匹配词组，其余都为空。

1 beam search图匹配过程

```
1: function BeamSearch
2:     stateSet ← φ
3:     finalSet ← φ
4:     initState
5:     将initState加入stateSet
6:     while stateSet 非空 do
7:         finalSet ← stateSet
8:         newStateSet ← φ
9:         for state in stateSet do
10:            if state已匹配完成 then
11:                将state加入newStateSet
12:            else
13:                将state未匹配部分通过分析模块取得待匹配词组nounPhrases
14:                for nounPhrase in nounPhrases do
15:                    在图中和待匹配结点匹配,计算相似度similarity
16:                    if similarity > threshold then
17:                        生成新的搜索结点newState
18:                        将newState加入newStateSet
19:                    end if
20:                end for
21:            end if
22:        end for
23:        SORT(newStateSet)
24:        stateSet ← newStateSet取前K个状态
25:    end while
26:    return finalSet
27: end function
```

匹配过程如下：

1. 将初始状态放入待匹配集合。
2. 从待匹配集合中取出状态，将未匹配部分通过分析模块得到一系列待匹配词组，分别放入图中匹配计算相似度，并生成新的搜索状态。若取到一个已匹配完成的状态，直接将其当作新的搜索状态。
3. 对本轮新生成的搜索状态剪枝：
   - 总分低于阈值则剪枝；
   - 排序后只留下分数最高的几个。
4. 用剩下的搜索状态生成新的待匹配集合，跳到 2.；若没有剩下新生成的搜索状态集合，说明匹配完成，跳到 5.。
5. 取出最终排名最高的前几个搜索状态，已匹配部分最后的匹配上的即是答案。

3. 加速方法

由于在线的系统需要满足实时性的要求，匹配速度不能过慢，我们使用了两

种方法来加速：构建索引加速匹配以及在前端后端之间建立缓存。

构建索引的作用有两个：第一是在 Neo4j 的结点定位时可以直接通过结点 id 得到该结点信息，速度比一般用字符串和匹配规则快；第二，经过分析，上述搜索方法中，搜索空间过大是造成搜索速度慢的主要原因，而索引可以帮助在图搜索的时候缩小搜索范围。我们构建了两个层次的索引，一个是基于词的，将所有 Neo4j 结点的名字分词，去停用词，做成倒排索引，直接索引到 id；另一个是基于字的。在搜索状态中加入一个属性，记录可能的结点集合，这样每个搜索状态在进行下一步搜索时，只需要和可能结点集合中的结点进行匹配。

初始状态的可能结点集合通过下述方法获得:

1. 得到待匹配词组的每个词索引到的结点集合的并集，令其为集合 A；
2. 若集合 A 的大小大于阈值，则直接使用集合 A 作为可能结点集；否则，再得到待匹配词组每个字索引到的结点集合的并集，令其为集合 B，使用 A 和 B 的并集作为可能结点集。

而在匹配当中的搜索状态的可能结点集由如下方法得到：

1. 得到该状态上一步匹配到的结点的相邻结点集，令其为集合 A;
2. 得到该状态未匹配部分通过字索引得到的结点集，令其为集合 B，使用 A 和 B 的交集作为可能结点集。

在实践中这种索引的方法大大提高了搜索速度，使得上述复杂的匹配系统变得可用。

另外一种方法是现在大多数层次间速度不一致系统使用的方法，即加入中间缓存，这在计算机内存-CPU 协调、服务器-浏览器协调之间都能见到。我们在网页前端和服务器后端之间之间建立了一张表进行双方的交互，见 Figure 2.，表中存储了问题答案对，若已经回答过的问题，网页前端会直接在该缓存表中获取。

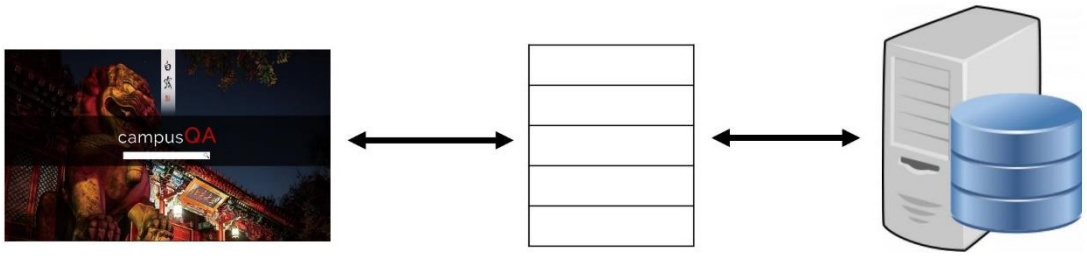

Figure 2. **缓存表**

结果展示

我们编写了一个简洁的网页前端，见 Figure 3.，只有一个提问题的交互窗口。Figure 4.展示了我们的系统对长名实体的支持以及模糊匹配的成果[12]。Figure 5. 展示了在匹配词极短甚至都是口语化简称的时候我们系统的支持[13]，

[12] 注:该问题的匹配路径是 信息科学技术学院→自然语言处理中的经验性方法→老师

[13] 注:该问题的匹配路径是 信息科学技术学院→计算机组成→老师→邮箱

值得一提的是这里输入了一个检索目标，而非完整问题。而更加简短常见，描述完整的问题，比如“地震概论的上课时间”，我们的系统能够更加准确、快速地给出回答。

Figure 3.网页前端

Figure 4. 部分问题回答展示 1

Figure 5. **部分问题回答展示** 2

讨论

目前我们的问答模型，在现有数据（学院、教师、课程）上是较为令人满意的，数据本身的高度结构化和图数据库以及图搜索的相关算法正应为配合得当、相互适合，效果才好。实际上，针对类似的数据（纯课程数据），我们曾经开发过一套使用当下最为流行的深度学习技术的系统，其主要架构来自于 Severyn A 等的工作[14]。然而效果并没有想象中的好，对于最基本的问题回答的正确率也是很低的。事实上，这并不是架构的问题，他们的神经网络架构早已被证明在短文本匹配方面是有效的，这是因为并没有选择合适的方法。在课程数据中，专有名词多，而词向量表示对于特定领域的专有名词有“先天的缺陷”，因为它们是从一般的文本中训练来的；另外，缺乏大量训练数据，通过规则生成的数据质量不高也是使深度学习模型表现不好的原因。总而言之，方法、模型应该是要和特定的场合、数据相适应的，硬套的结果很肯能不好。

然而我们这里使用的方法也有很大的改进空间和余地，最主要的问题是对问句形式的要求太死了，实际上出现句式结构各种各样的问句都是可能的。要处理一般形式上的问句一定会碰上以下两个问题，而这两个问题就是关键点：

1. 如何将问句表示为图结构；

[14] Severyn A, Moschitti A. Learning to rank short text pairs with convolutional deep neural networks[C]//Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2015: 373-382.

2. 如何解决子图同构问题，尤其是涉及模糊匹配的子图同构问题。

第一个问题是关于问题的“可解性”的，这个方法不正确则匹配出的结果肯定是不正确的；第二个问题是关于“效率”的，不能用很快的算法找到答案这个系统就不可用，考虑到这是一个实时系统，需要满足“交互式使用”的需求，暂时我们对效率更加看重一点。我们曾尝试过做更复杂问句的匹配，但是搜索空间的骤然加大导致匹配时间无法忍受。以我们来看，第二个问题更加关键一些，目前，在一些复杂的长问题上，我们的系统还是得花费几秒钟去搜索答案。

我们还有一些想法可以进一步提高速度，但是因为一些原因并没有来得及实现，下面是两个比较可行的方案：

1. 进一步优化索引，在同构匹配之前一次性从数据库中取得子图。我们的系统，之所以在长问题上速度显著变慢，是因为逐步的检索方案增加了 I/O 次数，因此一个好的方案是想办法精确地一次性提取出一个子图。这个子图需要足够大，对本次查询能够完全包含；并且足够小，能够存放在内存当中。这可能需要不断尝试确定索引的结点集的大小。
2. 利用中间缓存表的问答对信息，开发一个伴存的社区问答系统。一个问题先在该社区问答系统匹配，没有类似问题再和知识图匹配。但是这种方法的缺点是显著依赖于使用该系统的人数，使用人数过少时，缓存表里几乎没有问答对，该系统基本没有用处；而使用人数过多时，该社区问答系统的速度可能成为瓶颈。

从大来看，对比我们在立项时候的目标，我们实现的内容实在是太少了。事实上要回答校园内的一般性问题，至少还需要解决两种类型的问题：

1. 和文本抽取相关的问题，数据类型是无结构的纯文本，比如一些新闻类的问题，这种和一般文本相关的问题尤其适合深度学习模型；
2. 路径规划类的问题，这些是另一类图上的问题了，然而目的不是匹配，而是给出一个最优解，比如问“从 A 到 B 怎么走”这类问题。

上述两种类型的问题其实分别都是很大的门类了，而且都比我们目前做的有结构匹配要难，而且数据也更难获取，因此要真正完整地做出一个“完善”的校园问答系统还是任重而道远。

致谢

首先，感谢赵东岩老师、冯岩松老师对我们学习、生活上的帮助，尤其是对我们的本科生科研项目的支持。他们从一开始的选题、到框架设计、模型选择、再到论文的修改都给了我们十分中肯的建议，鼓励我们尝试、往深往广探索。可以说没有两位老师的悉心教导，我们的项目不足以成型，我们在这一年多也不会取得这样的进步。

其次，感谢实验室的邹磊老师、严睿老师、贾爱霞老师，他们经常在我们的工作报告上给予我们有益的指导意见，耐心地向我们解释问题。

感谢来雨轩师兄、罗炳峰师兄、韩喆师兄、吕超师兄，他们以过来人的身份给了我们许多有用的建议，包括项目上的，和学习生活上的。

最后，再次对关心、帮助我们的老师和同学表示由衷感谢！

参考文献：

[1] 邹磊. 图数据库中的子图查询算法研究[D].华中科技大学,2009.
[2] 张宁,朱礼军. 中文问答系统问句分析研究综述[J]. 情报工程,2016,2(01):32-42. [2017-10-11].
[3] 唐素勤,黄运有,王娜娜. 基于依存语法及本体技术的问句分析[J]. 广西师范大学学报(自然科学版),2014,32(04):52-58. [2017-10-11]. DOI; 10.16088/j.issn.1001-6600.2014.04.009
[4] 刘跃红. 问句依存句法及语义分析研究[D].昆明理工大学,2011.
[5] 陈康,樊孝忠,刘杰,余正涛. 受限领域问答系统的中文问句分析研究[J]. 计算机工程,2008,(10):25-27. [2017-10-11].
[6] Kim Y. Convolutional neural networks for sentence classification[J]. arXiv preprint arXiv:1408.5882, 2014.
[7] Severyn A, Moschitti A. Learning to rank short text pairs with convolutional deep neural networks[C]//Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2015: 373-382.
[8] Bordes A, Weston J, Usunier N. Open question answering with weakly supervised embedding models[C]//Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer Berlin Heidelberg, 2014: 165-180.
[9] Dong L, Wei F, Zhou M, et al. Question Answering over Freebase with Multi-Column Convolutional Neural Networks[C]//ACL (1). 2015: 260-269.
[10] Wang Z, Zhang J, Feng J, et al. Knowledge Graph Embedding by Translating on Hyperplanes[C]//AAAI. 2014: 1112-1119.
[11] Tunstall-Pedoe W. True knowledge: Open-domain question answering using structured knowledge and inference[J]. AI Magazine, 2010, 31(3): 80-92.
[12]Zou L, Huang R, Wang H, et al. Natural language question answering over RDF: a graph data driven approach[C]//Proceedings of the 2014 ACM SIGMOD international conference on Management of data. ACM, 2014: 313-324.
[13] Ullmann J R. An algorithm for subgraph isomorphism[J]. Journal of the ACM (JACM), 1976, 23(1): 31-42.
[14] Cordella L P, Foggia P, Sansone C, et al. An improved algorithm for matching large graphs[C]//3rd IAPR-TC15 workshop on graph-based representations in pattern recognition. 2001: 149-159.

作者简介：

汤济之，男，北京大学信息科学技术学院 2014 级学生，目前在北京大学计算机研究所互联网信息处理组实习，感兴趣的方向为互联网数据挖掘、自然语言处理、信息检索。

王一雪，女，1995 年 11 月出生于福建省厦门市，2014 年从福建省厦门第一中学考入北京大学信息科学技术学院计算机系，在校期间努力学习天天向上积极进取。

指导教师简介：

赵东岩，研究员，男，2000 年在北京大学获得计算机应用科学专业博士学位。现任北京大学研究员，全国新闻出版信息标准化技术委员会委员。主要研究方向为计算机网络与数据库应用技术、数字出版技术、Intelligent Agent。近年来承担国家级项目 5 项、主持 1 项，省部级科研项目 6 项、主持 2 项，发表学术论文 10 余篇，申请发明专利 7 项，先后五次获得国家和省部级奖励。其中，主持研发了具有国际先进水平的报业数字资产管理系统，获 2006 年度国家科技进步二等奖。个人获北京大学优秀共产党员（2004 年）、第十届中国青年科技奖（2007 年）等荣誉。