中国科学技术大学
University of Science and Technology of China

NLP Project

ON

USTC 问答系统

SUBMITTED BY

王禹

吴颖馨

代宇涵

UNDER THE GUIDANCE OF

PROF. 凌震华　TA: 陈越

(Academic Year: 2020 Autumn)

摘要

随着智能系统的发展，一个性能优异的问答系统（QA System）能够很大程度地改善用户的人机交互体验。而最初的问答系统大多基于大量的数据冗余来完成预测，它们通常没能考虑实际的语义特点。本项目针对这个问题进行了优化，并实现了一个与中国科学技术大学 (USTC) 相关知识的问答系统。这个系统首先对用户给定的问题进行语法分析，并以不同方式重写它们，从而让搜索引擎能够更好地利用其信息。以此为基础，本系统还结合了 N-gram Mining 技术完成压缩冗余，针对回答词类别正确性做了 N-gram Filtering（过滤），利用 N-gram Tiling 汇集答案。经过层层精细的处理后，再借由 NER filter 最终确认答案，以提高答案的准确性。

实验结果显示, 仅使用原论文中 AskMSR 的做法便可以获得比较好的效果. 再加上 NER 之后, 性能进一步提升. 同时我们也在实验中使用 Ablation Analysis, 测试了将不同的模块去掉之后的 recall 值和 ndcg 值, 从而说明每一个模块的必要性. 我们已将代码开源在 https://github.com/wangyu-ustc/USTC_QA_System.

Keywords—QA System, Short-question Answering, N-grams, USTC database, Mining, Filtering, Tiling.

1 背景介绍

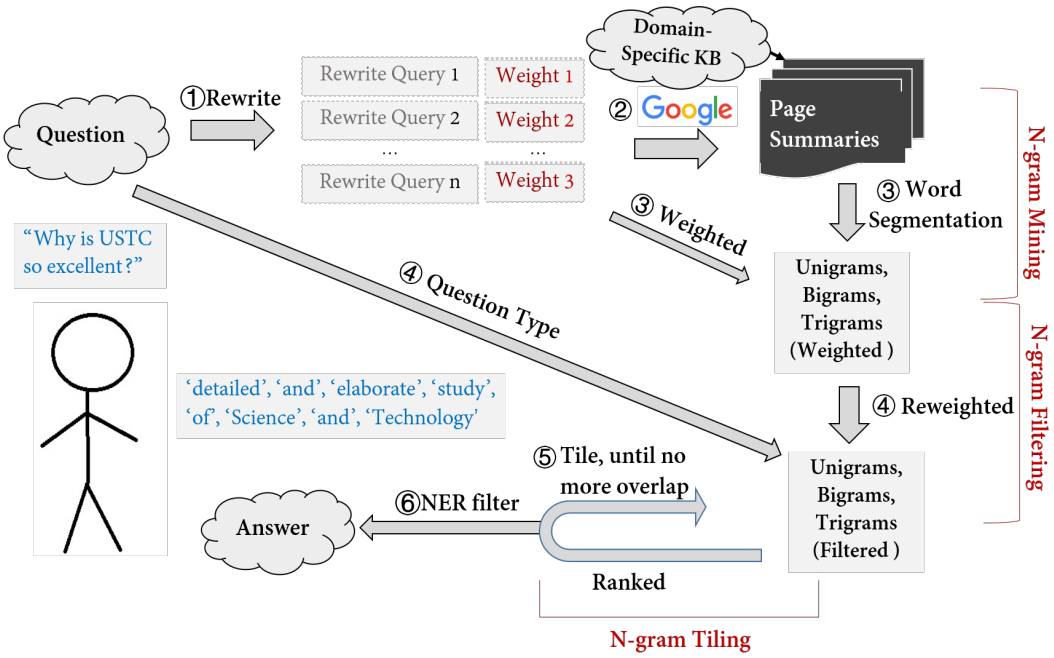
问答系统 (Question Answering System, QA) 作为信息检索系统的高级形式，可以用准确、简洁的自然语言回答用户用自然语言提出的问题。其研究兴起的主要原因是人们对快速、准确地获取信息的需求。问答系统是目前人工智能和自然语言处理领域中一个倍受关注并具有广泛发展前景的研究方向 [1]。

首先，利用机器来实现自动问答，需要确定其从外部获取知识的来源。从知识来源出发，QA 系统大致分为三类：基于知识库的问答、基于文档的问答、答案选择 [2]。本次的实验项目的知识来源为第一种，即在搜索引擎返回的知识库中进行进一步搜索与整合。不同于搜索引擎的是，搜索引擎需要人类来提供一些问题或者关键词，计算机将列出所有可能相关的答案，并最终由人来筛选自己所需要的信息；问答系统则能直接给出答案。

其次，本次项目获取答案的方式为事实检索式，而不同于目前比较受关注的生成式交互人机对话 [3]。另外，本次项目针对为类似于 “Who killed Abraham Lincoln?” 或 “How tall is Mount Everest?” 等简短问题，这种问题用户往往希望问答系统返回一个三元组或关键字。目前比较主流的解决方案有通过 Embedding Models[4]，或者结合 CNN、RNN 神经网络 [4] 等深度学习模型来得到回答。作为 QA 系统的初探，本次项目主要基于 AskMSR[5]，即对于 N-grams 采用三步走：mining, filtering, tiling，从而返回一个正确的概率较大的答案。

最后，AskMSR[5] 所提出的方法本身与 [3,7] 等架构类似，针对的都是 open-domain —开放式问答问题。本次实验所实现的 USTC 问答系统同样可以针对不同领域的知识做出回答，但为了保证本系统的优良性能，我们模型结结构作出一些 domain-specific 的修改。

2 模型结构



2.1 Rewrite Questions

重写问题分为两步：(1) 利用 parse tree 对输入的问题 Q 进行语法分析，从而得到每一个词在句子中的属性，例如对于"Where is the Louvre Museum located?"，经过剖析后我们可以对各个分词得到

- Where: where (WRB)
- is: be (VBZ/ B-VP)
- the: the (DT/ B-NP)
- Louvre: louvre (NNP-LOC/ I-NP)
- Museum: museum (NNP/ I-NP)
- located: locate (VBN, B-VP)

之后将上述剖析的结果与一系列给定的 pattern 进行匹配,可能的 pattern 如"{WP} {be} {NP}", "{WP} {do} {NP} {VB}" 等，从而得到符合一定语法要求的重述问句 $\mathcal{Q}_i$。根据经验，每个重写方式对应一定的权值，这个权值将在后续得到利用。即我们最后将得到 $\{(\mathcal{Q}_i, w_i)|i = 1, ..., |P|\}$，其中 $|P|$ 为可能的正确匹配个数。

这样的 rewrite 方式能帮助我们利用网络上庞大的数据冗余，出现相同答案的多种语言形式从而增加了找到答案的机会。

2.2 N-gram Mining

对问题进行重写为 $\mathcal{Q}_i$，我们将其作为搜索引擎的输入。特别地，我们采用 Google 提供的 API 来获取知识库。我们按照论文 [6] 中的要求，对于 $\mathcal{Q}_i$ 只获取对应的摘要，而不是返回的 URL 网页中的全部信息。由于我们的项目目的为构建针对 USTC 的问答系统，因此在按照论文 [6] 的做法以外，我们还引入了 domain-specific 的数据库，其中包括 USTC 的官方英文主页以及 Wikipedia 中的 USTC 的文本信息。

需要说明的是，在后期我们也试图通过获取每个 URL 对应的文本，以期获取更高的正确率，但更多的与问题无关文本的引入反而排挤了原本正确的回答。

在进一步操作之前，需要对返回的文本进行处理，一方面是去除无关符号并判断结束符，从而进行分句；另一方面是提取 Uni-gram, Bi-gram, Tri-gram(根据问题的复杂程度，可以取至 k-gram)。因此，在这一步中我们在上一步的基础上得到了每个 $\mathcal{Q}_i$ 所对应的 N-gram 知识库，即

$$\{(\mathcal{Q}_i, w_i, [G_i^{(1)}, ..., G_i^{(k)}])|i = 1, ..., |P|\} \tag{1}$$

其中 $G_i^{(k)} = \{G_{ij}^{(k)}|j = 1, ..., |n_i^{(k)}|\}$ 表示 $\mathcal{Q}_i$ 所对应的 k-gram 的集合。N-gram Mining 进一步计算 Gram-wise Score, 按照 [6] 的做法，我们不计算 n-gram $G_{ij}^{(k)}$ 在每个摘要中出现的频率，$G_{ij}^{(k)}$ 的最终分数基于生成它的重写结果 $\mathcal{Q}_i$ 所相关联的权重 w_i，以及生成 $G_{ij}^{(k)}$ 的 $\mathcal{Q}_i$ 数量, 即

$$Score(G_{ij}^{(k)}) = \sum_{i}^{|P|} w_i \mathbb{I}(G_{ij}^{(k)} \in G_i^{(k)}) \tag{2}$$

最终可以得到每个 n-gram 及其所对应的分数，即 $\{(G_l, S_l)|l = 1, ..., |G|\}$.

Mining 可以起到压缩冗余,并且加强频繁词的评分,但仍然有干扰词出现,如在 $\mathcal{Q}$ ="Who is the president of USTC?" 中的出现 tri-gram = "Energy chemistry Speech"; 以及出现 uni-gram = "Bao" 或"Xinhe" 等语义不完整现象。

2.3 N-gram Filtering

一个理想的回答需要符合正确性，而正确性的前提为与问题的相关性。首先考虑提问词，若对于一个"How many" 问题，系统返回一个实体；或者对于一个"Where" 问题，系统返回一个人名，这种结果无疑都是失败的。因此，对于一个特定问题，我们定义削减集合 $\mathcal{R}_-$ 与增益集合 $\mathcal{R}_+$. 集合中可能的元素代表如 Location、Number、Name 等一类词。随着提问词的不同，$\mathcal{R}_-$ 与 $\mathcal{R}_+$ 也有所不同，以下是部分案例:

表 1: Reweight 中提问词所对应的削减集合 $\mathcal{R}_-$ 与增益集合 $\mathcal{R}_+$。

Question type	$\mathcal{R}_-$	$\mathcal{R}_+$
when	Name, Location	Time, Number
who	Location	Name, Organization
where	Name, Time	Location, Upper
how many/old/much	Name, Location	Number
which/what country/countries	-	Country

2.4 N-gram Tiling

较大程度地确定了回答的正确性后，我们从重叠的较小的答案片段中贪婪地汇集所有可能的较长答案，以最终返回用户一个完整的回答。我们同样列出一些样例如下所示:

表 2: Tiling 的示例

Original	After one round	After two rounds
is BAO	president is	president is BAO Xinhe
USTC president	is BAO Xinhe	USTC president is BAO Xinhe
president Donald Trump	USTC president	USTC president Donald Trump

如果我们能设计出一种合适的方法来重新调整 Tiling 之后短语或句子的权重，我们就能得到关于"Who is the president of USTC?" 的正确答案 "USTC president is BAO Xinhe"。

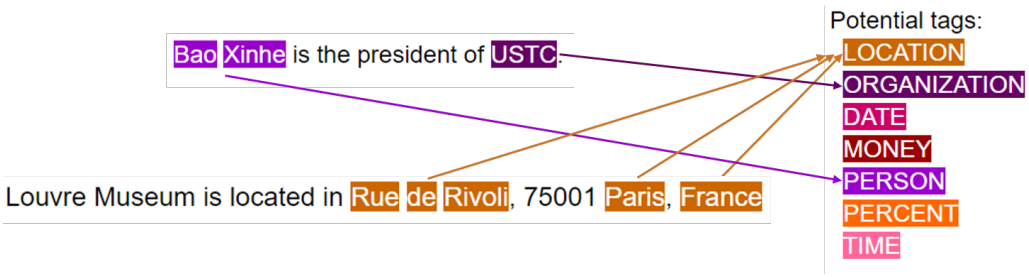
2.5 NER filter

关于实体的识别，我们在实验中使用了 Stanford Named Entity Recognition model (NER), 其可以给出某一实体所对应的标签。我们进一步用这类标签信息，在返回用户最后答案时对答案过滤一次，从而再次确保答案的准确性。

由于 NER 耗时较长, 因此只对 Tile 返回的结果的前 10 个答案进行命名实体识别, 并对这些答案进行 rewrite. 这样一个问题的耗时大约增加 2-3s.

表 3: NER 识别出来的 TAG 所对应的削减集合 $\mathcal{R}_-$ 与增益集合 $\mathcal{R}_+$.

Question type	$\mathcal{R}_-$	$\mathcal{R}_+$
when	PERSON, LOCATION, ORGANIZATION	TIME
who	LOCATION,ORGANIZATION,O	PERSON
where	PERSON,TIME	LOCATION
how many/old/much	PERSON, LOCATION	MONEY



3 实验结果及分析

3.1 数据集

数据集采用 127 条与 USTC 相关的 question-answer 对，根据提问词，数据集相关信息如下表所示：

表 4: USTC-QA 数据集信息

	What	Who	When	Which	How	Where
数目	19	15	28	30	11	24

具体数据集可见开源代码 USTC-QA-System.

3.2 定量分析

3.2.1 评价指标

为了较好地评价搜索返回的结果，我们采用 Recall(用于表征前 k 个候选答案的正确率) 与 NDCG(一种排序指标, 目标答案 (GroundTruth) 在所有候选答案中的排序越靠前, 指标值越高) 两个评价指标，其定义如下:

$$\text{Recall@}k = \frac{\text{取前 } k \text{ 个候选答案, 其中存在 GroundTruth 的问题数量}}{\text{总问题数量}}$$

$$\mathrm{CG_p} = \sum_{i=1}^{p} rel_i, \quad \mathrm{DCG_p} = \sum_{i=1}^{p} \frac{2^{rel_i}-1}{\log_2(i+1)}$$

$$\mathrm{IDCG_p} = \sum_{i=1}^{|REL_p|} \frac{2^{rel_i}-1}{\log_2(i+1)}, \quad \mathrm{NDCG_p} = \frac{\mathrm{DCG}_p}{\mathrm{IDCG}_p}$$

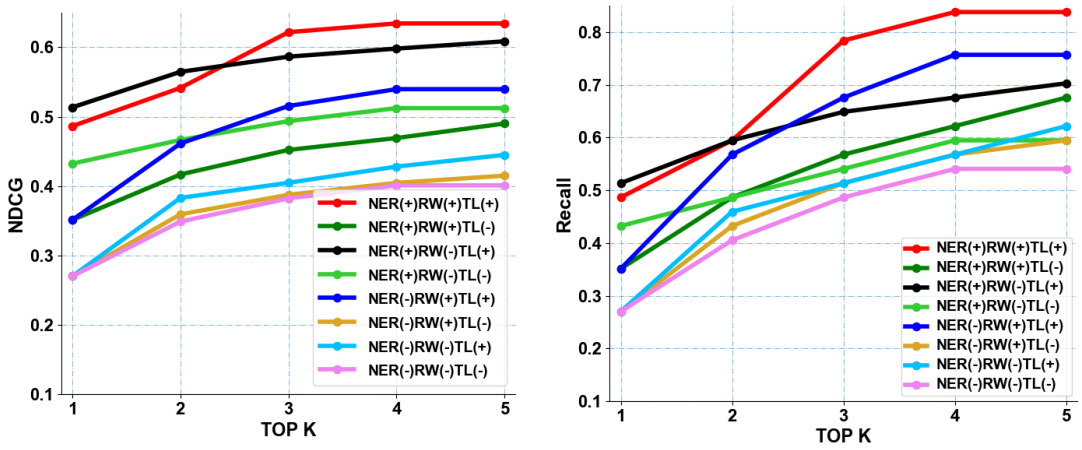


图 1: 评价结果

3.2.2 评价结果

程序评价结果可见图1. 其中 NER,RW,TL 分别表示 Named Entity Recognition, Rewrite 以及 Tile. 后面的 (+) 和 (-) 表示是否包含这一模块. 从图中见过可以得出如下几点结论:

- 所有模块全部加上的效果最好, 全部去掉时的效果最差.
- 从图中可以看出有 NER 时没有 Rewrite 反而在 Top-1 和 Top-2 处有提升, 可能是因为 rewrite 能够获得更多的 query, 搜索返回更多 summaries, 同时也增大了产生更多的干扰项的概率, 进一步会影响真实答案的位置, 即影响 Top-1 和 Top-2 的性能。但通过比较总体的性能, 进行 Rewrite 后的实验效果要更好.
- 没有 NER 时, Rewrite + TILE 一起使用远胜其它方法, 这样的结果也与 AskMSR 原论文中保持一致.
- 图中深蓝色线条为原论文的复现结果, 可见 NER 在最后环节的引入能够在一定程度上提升性能。

如表 5 所示, 其中 Avg Drop Rate 表示 Recall@2 和 Recall@5 下降率的平均. 从表中可以大致看出各个模块的相对重要性.

表 5: 模块重要性分析

model	recall@2	recall@5	Avg Drop Rate
Baseline	0.595	0.838	0.0%
No NER	0.568	0.757	7.1%
No Tile	0.486	0.676	18.8%
No Rewrite	0.595	0.703	8.1%

表 6: 正确样例

Questions	Ground Truth	Answers
Who is the Dean of the School of Computer Science of USTC?	Dr Enhong Chen	Dr Enhong Chen
Who is the leader of quantum physics in USTC?	Jian-wei Pan	Jian-Wei Pan Jianwei
How many students are in USTC?	15500, 13794	totals around 13794 8243 undergraduate
Where is the electronic teaching building of USTC?	West Campus	West Campus
What is the name of the human-like robot developed by USTC?	Jiajia	conversation with Kevin Kelly Jia Jia as
Which city was USTC relocated to?	Hefei	to Hefei in 1970
When did USTC launch Micius?	In 2016	space on August 15 2016

3.3 定性分析

3.3.1 正确样例展示

具体结果见表6. 从表格中可以看出来, 我们的 QA system 可以对一些问题做到有效地回答. 虽然仍有部分有所瑕疵;: 如 tile 过剩问题, 比如目标答案是"Jiajia", 但系统却回答出一些多余的无用信息。虽然如此，但在大部分的问题上，我们的系统总能提取出正确答案，其总体结果较为令人满意。

3.3.2 错误样例分析

具体样例及结果见表 7 . 表中的 Reasons 在此进行解释:
(1) **搜索结果中目标答案出现次数太少**: 目标答案在 rewrite 的句子通过 google

引擎搜索到的多个 sumamry 中只出现过一两次, 因此即便有符合条件的结果, 由于出现次数太少, 也很难被赋予高分.

(2) **出现足以以假乱真的答案**: 即便目标答案出现过很多次, 但是由于出现一些足以以假乱真的答案, 同样满足 Filter 中的多个判定, 会被赋予与真实答案相似或更高的分数, 但实际上是错误的. 例如样例 2 中的电话号码, 错误答案的出现次数和真实答案的出现次数差不多, 但一旦我们将其切分为 n-gram, 便无法分辨哪一个才是正确答案了.

(3) **干扰项太多**: 在 Google 引擎中搜索时, 可能会有一些答案出现了很多次, 但是可能只是与 USTC 高度相关, 或与其中的某些关键词相关, 而并非与所要问的问题高度相关. 因此干扰项太多就容易造成即便找到了正确答案, 也很难与这种分数很高的结果相比较. 在表中的第 5 个例子中, "Guo Moro" 也出现在了候选答案当中, 分数为 22, 但 Award 分数为 34, 高居首位.

(4) **Tile 出的句子过长**: 答案虽然已找到, 但是 tile 出来的结果颇有画蛇添足之味道.

表 7: 错误样例分析

Questions	Ground Truth	Answers	Reasons
what is the ratio of student to faculty in USTC?	8 students per 1 faculty	totals around 13794 8243 students	(1)
What is the telephone number for the President's Office of USTC?	+86-551-63602184, 551-63602184	MO 65409 573-341-4111 800-522-0938 Contact	(1)(2)
What is the rank of USTC in U.S. News Rankings 2020?	124	2020/21 1 18 8	(2)
Which city was USTC founded in?	Beijing	Hefei	(2)
Which prize is the biggest scholarship of USTC?	Guo Moruo	Award	(3)
Where is USTC established?	Beijing	founded in Beijing in Beijing in September	(4)

4 程序使用文档

File Name(.py)	Usage
search	定义了主函数，包括系统的输入输出.
query	定义 Query 对象
engine	定义搜索引擎对象，输入 query 后能够返回一系列 summaries.
filter	根据特定的 query 以及 n-gram 集合，对每一个 n-gram 进行 reweight.
tile	结合文本库，对单个 n-gram 进行贪婪地堆叠.

表 8: 代码布局

5 成员分工

分工如下:
(1) 前期分工为: 王禹负责 Filter + NER 模块, 吴颖馨负责 Rewrite + google search 模块; 代宇涵负责 Tile 模块;
(2) 后期分工: 王禹负责找 Who, What 相关问题 + 改进 NER, 代宇涵负责找 Which, When 相关问题, 并将第一个页面的 url 全部拉出来并一个个进入提取信息 (虽实验结果表明未有性能提升, 但仍然是一个有意义的尝试) ; 吴颖馨负责找 Where, How 相关问题, 并调研更多的方法。
(3) 论文部分：吴颖馨负责摘要、背景以及理论框架部分的撰写，王禹、代宇涵负责实验结果与分析部分的撰写。
注：实际工作过程中, 每个模块每个人都有参与.

6 Reference

[1] 百度百科，https://baike.baidu.com/item/问答系统/9641943?fr=aladdin
[2] CSDN, https://blog.csdn.net/u012892939/article/details/79476756
[3] Haoyu Song, Yan Wang, Wei-Nan Zhang1, Xiaojiang Liu, Ting Liu, Generate, Delete and Rewrite: A Three-Stage Framework for Improving Persona Consistency of Dialogue Generation

[4] Antoine Bordes, Jason Weston, and Nicolas Usunie. rOpen Question Answering with Weakly Supervised Embedding Models, 2014.
[5] David Golub, Xiaodong He. Character-Level Question Answering with Attention. ACL(2016).
[6] Brill E., Dumais, S. and Banko, M. (2002). An Analysis of the AskMSR question-answering system. In EMNLP 2002, pp 257-264.
[7] Danqi Chen, Adam Fisch, Jason Weston & Antoine Bordes. Reading Wikipedia to Answer Open-Domain Questions.