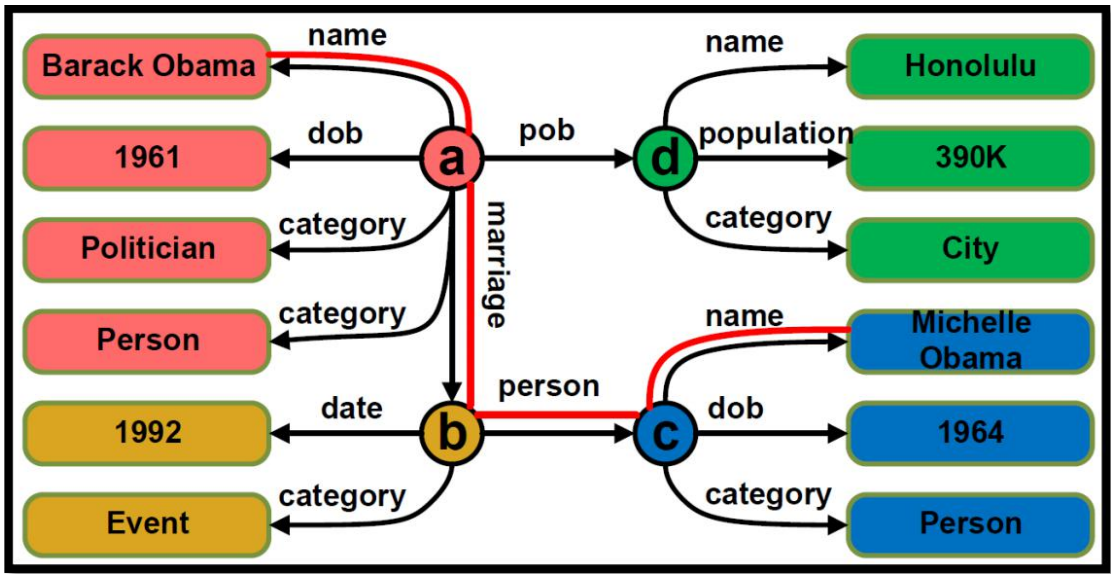
基于知识图谱的问答系统

一、问答系统与知识库

问答系统（QA）已经成为人类访问十亿级知识图谱的流行方式，它回答的是自然语言问题。 QA 系统最有名的故事之一就是 IBM WATSON 在 2011 年参加了 Jeopardy 竞赛，打败了所有人类竞争对手，获得了 100 万美元的奖励。

现在我们来谈谈知识库。近年来，我们目睹了知识库的发展，越来越多的大规模知识库涌现出来，如 Google Knowledge graph，Yago 和 Freebase 等。这些知识库具有体量大，质量高的特点。

一个知识库包含了大量的结构化数据。下图给出了一个关于 Obama 的知识图谱示例。知识库中的每一个三元组代表一个知识或某个事实。 例如，一个三元组（d，人口，390k）表示檀香山的人口为 390k。



二、KBQA

KBQA 指的是以知识库作为答案来源的问答系统。那么它是如何工作的呢？关键在于将自然语言问题转换为知识库上的结构化查询。例如，要回答“有多少人住在檀香山？”这个问题，我们需要将其转移到 SPARQL 或者 SQL 查询。 这里的关键问题是属性推断。

关于属性推断，我们面临两个挑战。

第一个挑战是问题表示。对于任意一个 QA 系统，我们需要一个具有代表性的问题表示来帮助识别具有相同语义的问题，同时区分不同意图的问题。第二个挑战是语义匹配，如何将问题表示映射到知识库中的结构化查询？

三、问题优化

然而，之前的解决方案并不能解决上述提出的挑战。

我们研究了两个主流的解决方案。

第一个是基于模板/规则的方法。这个方法用模板表示句子，语义解析往往通过人工标记来实现。这种方法的优点是它的结果是用户可控的，这使得它更适用于工业用途。缺点是严重依赖人工，成本太高，昂贵的人力成本使得它无法处理多样性的问题。

另一个是基于神经网络的方法。最近这种做法很受欢迎，它们通过 embedding 的方式来表示一个问题，并从 QA 语料库中学习出它的语义解析。这种方法的优点是 embedding 是灵活的，所以它可以理解各种各样的问题。缺点是基于神经网络的方法通常具有较差的解释

性，此外，结果是不可控的，所以他们并不适用于工业应用。

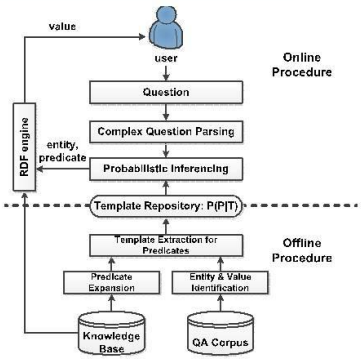
因此，我们不禁会想：能不能提出一种新的方法兼备这两种方法的优点？

为了做到这一点，我们用模板来表示自然语言问题。 例如，“檀香山有多少人？”的模板成为“城市里有多少人？”。因为使用了模板作为问题表示，我们的方法具有可解释性和用户可控性。

然而，我们并不是手动标记模板，而是从 QA 语料库中自动学习模板。 最终，我们为 2,782 个意图学到了 2,700 万个模板，这么大量的数据保证我们可以理解不同的问题。

System Architecture

- Offline procedure
  - Learn the mapping from templates to predicates: P (p|t),
  - Input: qa corpora, large scale taxonomy, KB
  - Output: P(P|T)
- Online procedure
  - Parsing, predicate inference and answer retrieval
  - Input: binary factoid questions (BFQs)
  - Output: answers in KG



这个系统体系结构如图所示。它主要包括两个过程：离线预处理部分和在线 QA 部分。

我们先来看看离线过程，离线过程的目标是学习出从模板到属性的映射。

再来看在线部分，当一个问题进来，系统首先将其解析和分解为一组二元事实型问题。对于每个二元事实型问题，系统使用概率推断来寻找它的值。这个推断是基于给定模板的属性分布来得到的。

Problem Model

- Given a question q, our goal is to find an answer v with maximal probability (v is a simple value)

$$\arg\max_{v} P(V = v|Q = q) \longrightarrow \arg\max_{v} \sum_{e,t,p} P(v|q,e,t,p)$$

e: entity; t: template; p: predicate

- Basic idea：We proposed a generative model to explain how a value is found for a given question,
- Rationality of probabilistic inference
  - *uncertainty* (e.g. some questions' intents are vague)
  - *Incompleteness* (e.g. the knowledge base is almost always incomplete),
  - *noisy* (e.g. answers in the QA corpus could be wrong)

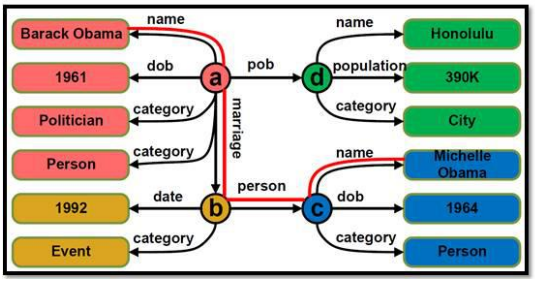
接下来，我们对这个问题进行形式化定义。给定问题 q，问答系统的目标是寻找具有最大概率的答案 v（其中，v 是一个简单值）。

我们提出了一个生成模型来解释如何为一个问题找到它的答案。

我们认为使用概率推断的方法来做 KBQA 是非常合理的。首先，一些问题的意图是模糊的。其次，大多数知识库都是不完整的。最后，QA 语料库中的答案也可能是错误的。

question2answer: a generative process

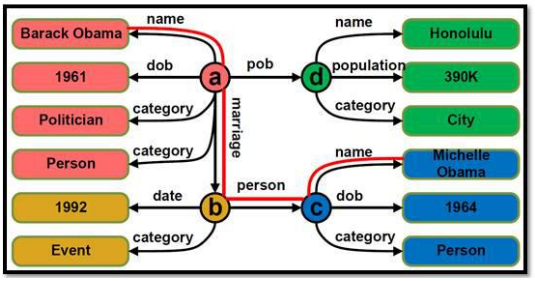
- A qa pair
  - Q: How many people live in Honolulu?
  - A: It's 390K.



我们以这个问答对来说明这个生成过程。

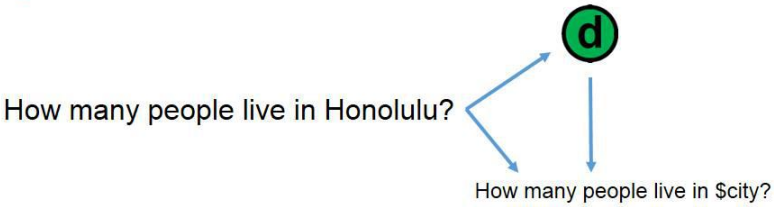
question2answer: entity linking

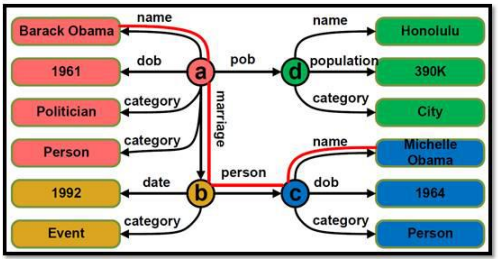
How many people live in Honolulu? → d



从用户问题 q 开始，我们首先生成或者说识别出其中对应的知识库中的实体 d。

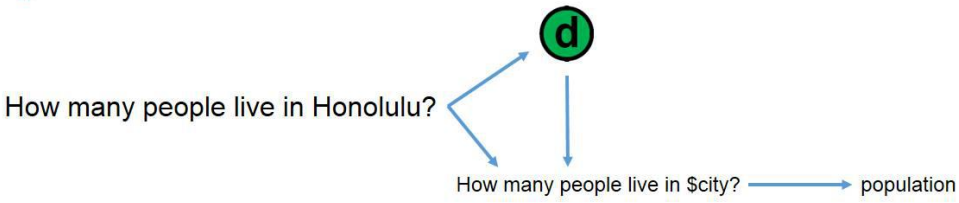
question2answer: conceptualization

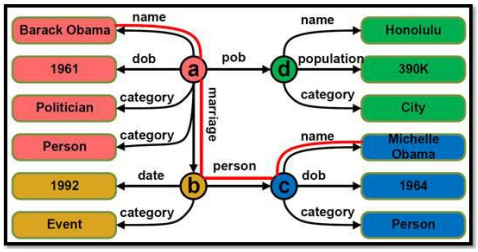




在知道问题和实体之后，我们根据 d 的概念分布生成模板 t。 这样，我们得到了一个模板“有多少人住在某城市？”

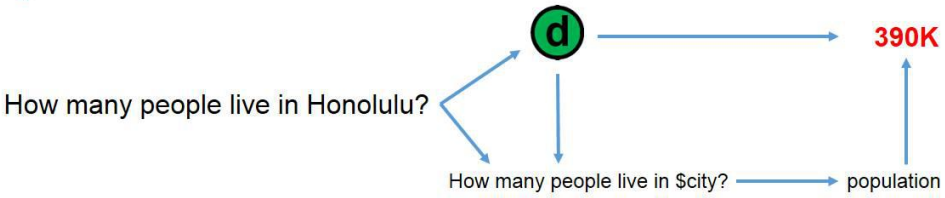
question2answer: predicate inference

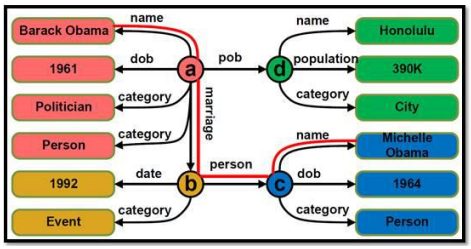




由于属性只与模板有关，所以我们推断出这个属性的模板为“population”。

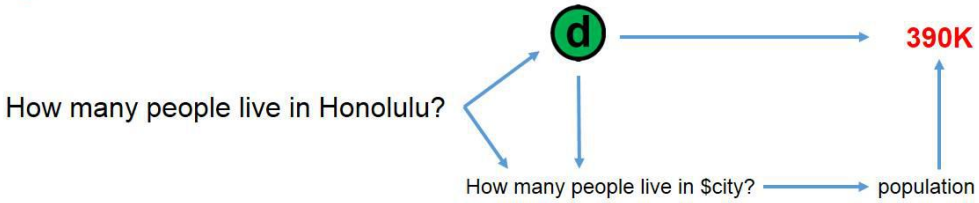
question2answer: value lookup





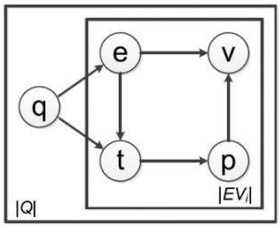
最后，给定实体 d 和属性 population，我们通过查找知识库来得到它的答案。

Probabilistic graph model



$$P(q,e,t,p,v) = P(q)P(e|q)P(t|e,q)P(p|t)p(v|e,p)$$

$$\arg\max_{v} \sum_{e,t,p} P(v|q,e,t,p)$$



kw.fudan.edu.cn/qa

通过这种方法，我们完成了从一个自然语言问题到生成答案的整个过程。这个过程可以建模为一个概率图模型。

基于这个生成模型，可以得到一个联合概率分布，进而用来解决给定其他变量求最大 v 的条件概率问题。

Probability Computation

- Source
  - QA corpora (42M Yahoo! Answers)
  - Knowledge base such as Freebase
  - Probase(a large scale taxonomy)
- Directly estimated from data
  - Entity distribution P (e|q)
  - Template distribution P(t|q,e)
  - Value (answer) distribution P(v|e,p)

Question	Answer
When was Barack Obama born?	The politician was born in 1961.
When was Barack Obama born?	He was born in 1961.
How many people are there in Honolulu?	It's 390K.

Yahoo! Answers QA pairs

kw.fudan.edu.cn/qa

下一个问题是如何计算出联合概率分布公式中的每一种概率。

我们可以从语料库直接估计出来大部分的概率。例如实体分布的概率，模板分布的概率以及值分布的概率。

我们从雅虎问答的 4200 万的 QA pairs 中，学习出问题模板和属性的映射关系。表中展示了 QA 语料库中的一些例子。

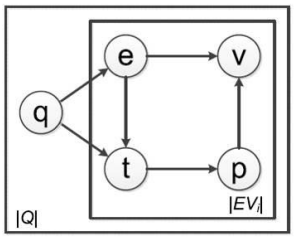
P(P|T) estimation

- We treat P(P|T) as parameters, and learn the parameter using maximum likelihood estimator, maximizing the **likelihood** of observing QA corpora
- An EM algorithm is used for parameter estimation

$$\hat{\theta} = \arg\max L(\theta)$$

$$L(\theta) = \sum_{i=1}^{m} \log P(x_i) = \sum_{i=1}^{m} \log P(q_i, e_i, v_i)$$

$$= \sum_{i=1}^{m} \log[\sum_{p \in P, t \in T} P(q_i)P(e_i|q_i)P(t|e_i,q_i)\theta_{pt}P(v_i|e_i,p)]$$



kw.fudan.edu.cn/qa

最后我们来估计 P(P|T)的值。基本思路是将 P(P|T)作为参数，然后使用极大似然法来估计 P(P|T)。

这里我们使用了 EM 算法来进行参数估计。

Answering complex questions

- When was Barack Obama's wife born?
  - (Who is) Barack Obama's wife?
  - When was Michelle Obama born?
- How to decompose the question into a series of binary questions?

$$\arg\max_{\mathcal{A}\in\mathbb{A}(q)} P(\mathcal{A})$$

- A binary question sequence is meaningful, only if each of the binary question is meaningful.

$$P(\mathcal{A}) = \prod_{\check{q}\in\mathcal{A}} P(\check{q})$$

- A dynamic programming (DP) algorithm is employed to find the optimal decomposition.

kw.fudan.edu.cn/qa

KBQA 的另一个难点就是回答复杂问题。在面对复杂问题时，我们采用了分治算法。首先，系统把问题分解为一系列的二元事实型问题，然后系统依次回答每个问题。每个问题的答案都是一个概率，我们通过动态规划算法找到最优分解。

Experiments

	KBQA	Bootstrapping
Corpus	41M QA pairs	256M sentences
Templates	27,126,355	471,920
Predicates	2782	283
Templates per predicate	9751	4639

KBQA finds significantly more templates and predicates than its competitors despite that the corpus size of bootstrapping is larger.

$marriage \rightarrow person \rightarrow name$

Who is $person marry to?
Who is $person's husband?
What is $person's wife's name?
Who is the husband of $person?
Who is marry to $person?

Concept based templates are meaningful

接下来我们来看看实验部分。我们首先通过实验证明属性推断的有效性。我们从学习出的属性数量和模板数据来对比我们的方法和 bootstrapping 方法。结果表明，我们的 KBQA 方法能得到更多的属性和模板， 这意味着 KBQA 在属性推理中更有效。大量的模板可以确保 KBQA 理解不同的问题模板，同时，大量的属性可以确保 KBQA 理解不同的关系。

Experiments

	#pro	#ri	#par	R	R^*	P	P^*
Xser	42	26	7	0.52	0.66	0.62	0.79
APEQ	26	8	5	0.16	0.26	0.31	0.50
QAnswer	37	9	4	0.18	0.26	0.24	0.35
SemGraphQA	31	7	3	0.14	0.20	0.23	0.32
YodaQA	33	8	2	0.16	0.20	0.24	0.30
				R R_{BFQ}	R^* R^*_{BFQ}		
KBQA+KBA	7	5	1	0.10 0.42	0.12 0.50	0.71	0.86
KBQA+Freebase	6	5	1	0.10 0.42	0.12 0.50	0.83	1.00
KBQA+DBpedia	8	8	0	0.16 0.67	0.16 0.67	**1.00**	**1.00**

Results over QALD-5. The results verify the effectiveness of KBQA over BFQs.

我们也在很多 benchmarks 上用到了我们的 KBQA。图为 QALD-5 的结果。结果表明，KBQA 具有最高的准确度。由于 KBQA 只回答二元事实型问题，因此召回率相对较低。如果我们只考虑二元事实型问答，召回率能上升到 0.67。

Experiments

Hybrid systems

- First KBQA
- If KBQA gives no reply, then baseline systems.

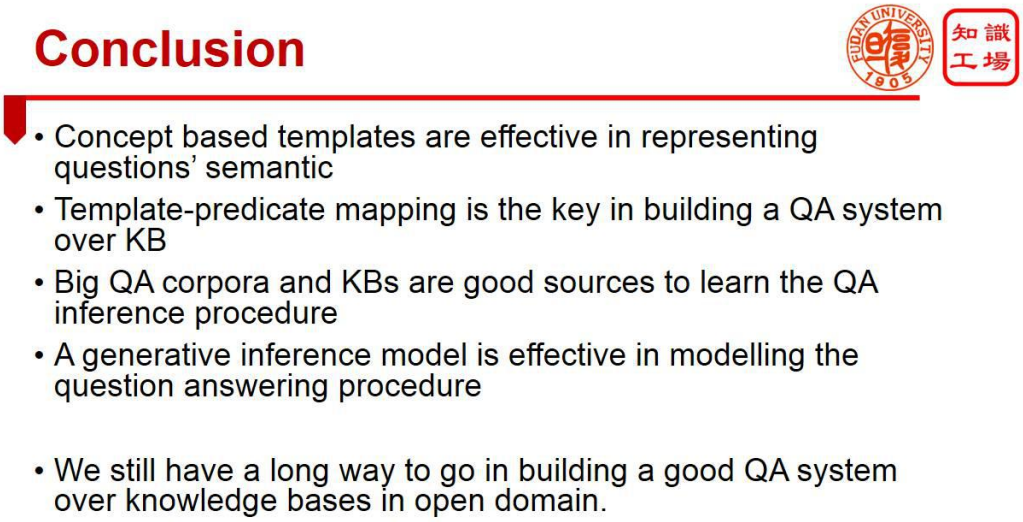
System	R	R*	P	P*
SWIP	0.15	0.17	0.71	0.81
KBQA+SWIP	0.33(+0.18)	0.35(+0.18)	0.87(+0.16)	0.92(+0.11)
CASIA	0.29	0.37	0.56	0.71
KBQA+CASIA	0.38(+0.09)	0.44(+0.07)	0.66(+0.10)	0.76(+0.05)
RTV	0.3	0.34	0.34	0.62
KBQA+RTV	0.39(+0.09)	0.42(+0.08)	0.66(+0.32)	0.71(+0.09)
gAnswer	0.32	0.43	0.42	0.57
KBQA+gAnswer	0.39(+0.07)	-	-	-
Intui2	0.28	0.32	0.28	0.32
KBQA+Intui2	0.39(+0.11)	0.41(+0.09)	0.39(+0.11)	0.41(+0.09)
Scalewelis	0.32	0.33	0.46	0.47
KBQA+Scalewelis	0.44(+0.12)	0.45(+0.12)	0.60(+0.14)	0.62(+0.15)

Results of hybrid systems on QALD-3 over DBpedia. The results verify the effectiveness of KBQA for a dataset that the BFQ is not a majority.

即使在一个不以二元事实型问题为主的数据集中（如 WEBQUESTIONS，QALD-3），KBQA 也可以作为混合问答系统的一个完美组件。

我们这样构建混合问题系统：一个问题过来，首先提交给我们的 KBQA 系统。如果 KBQA 系统不能回答，这意味着这个问题很可能不是二元事实型问题。然后，我们再将这个问题提交给 baseline 系统。

结果表明，当使用了我们的 KBQA 系统后，baseline 系统的性能都有了很明显的提高。



最后，我们对本文进行总结。我们构建了一个基于知识库的问答系统 KBQA。 我们的 QA 系统和以前的系统有两个明显区别：第一，它使用模板理解问题；第二，它从非常大的 QA 语料库中学习语义解析。

我们认为系统还有很多可以改进的地方。 首先，目前关于 QA 系统的研究主要建立在开放领域的知识库上。因此，研究如何使这些系统适应不同特定领域的应用是非常重要的。其次，我们希望可以通过常识推理来更深入的理解问题。 再者，由于知识库仍然存在数据缺陷问题，如何使用互联网作为外部知识变得非常重要。

以上内容翻译自自复旦大学知识工场肖仰华教授在 VLDB 2017 会议上的论文报告，题目为《KBQA: Learning Question Answeringover QA Corpora and Knowledge Bases》，作者包括：崔万云博士（现上海财经大学讲师），肖仰华教授（复旦大学）等等。
由于我们所采用的的 QA 系统借鉴该文章所采用的方法，故在此不多赘述，仅整理此文为读者介绍。