**内容速览**

* 什么是知识库（knowledge base, KB）
* 什么是知识库问答（knowledge base question answering, KB-QA）
* 知识库问答的主流方法
* 知识库问答的数据集

**什么是知识库**

*“奥巴马出生在火奴鲁鲁。”*

*“姚明是中国人。”*

*“谢霆锋的爸爸是谢贤。”*

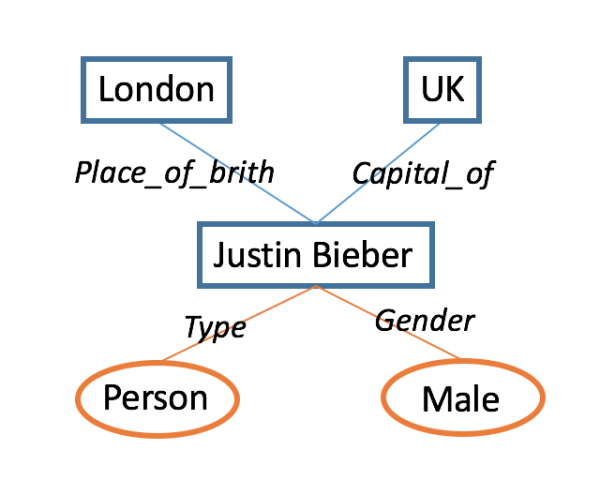
这些就是一条条知识，而把大量的知识汇聚起来就成为了知识库。我们可以在wiki百科，百度百科等百科全书查阅到大量的知识。然而，这些百科全书的知识组建形式是**非结构化**的**自然语言**，这样的组织方式很适合人们阅读但并不适合计算机去处理。为了方便计算机的处理和理解，我们需要更加形式化、简洁化的方式去表示知识，那就是*三元组（triple）*。

“奥巴马出生在火奴鲁鲁。” 可以用三元组表示为 *(BarackObama, PlaceOfBirth, Honolulu)。*

这里我们可以简单的把三元组理解为 (实体entity,实体关系relation,实体entity)，进一步的，如果我们把实体看作是结点，把实体关系（包括属性，类别等等）看作是一条边，那么包含了大量三元组的知识库就成为了一个庞大的知识图。

知识库可以分为两种类型，一种是以[Freebase](http://link.zhihu.com/?target=https%3A//developers.google.com/freebase),[Yago2](http://link.zhihu.com/?target=http%3A//citeseerx.ist.psu.edu/viewdoc/summary%3Fdoi%3D10.1.1.412.9675)为代表的Curated KBs，它们从维基百科和WordNet等知识库中抽取大量的实体及实体关系，可以把它们理解为是一种**结构化**的维基百科，被google收购的Freebase中包含了上千万个实体，共计19亿条triple。

值得一提的是，有时候会把一些实体称为*topic*，如Justin Bieber。实体关系也可分为两种，一种是**属性property**，一种是**关系relation。**如下图所示，属性和关系的最大区别在于，属性所在的三元组对应的两个实体，常常是一个topic和一个字符串，如属性Type/Gender，对应的三元组*（Justin Bieber, Type, Person）*，而关系所在的三元组所对应的两个实体，常常是两个topic。如关系Place\_of\_Brith，对应的三元组*（Justin Bieber, Place\_of\_brith,London）*。



（图中蓝色方块表示topic，橙色椭圆包括属性值，它们都属于知识库的实体，蓝色直线表示关系，橙色直线表示属性，它们都统称为知识库的实体关系，都可以用三元组刻画实体关系和实体）

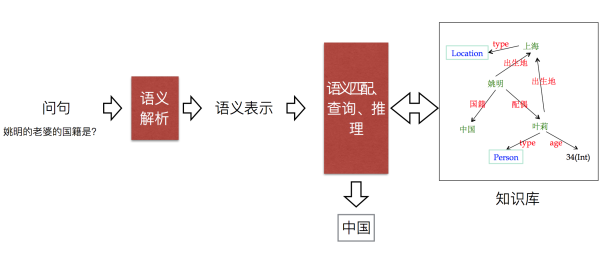
但是，像维基百科这样的知识库，与整个互联网相比，仍只能算沧海一粟。知识库的另外一种类型，则是以[Open Information Extraction (Open IE)](http://link.zhihu.com/?target=http%3A//www.aaai.org/Papers/IJCAI/2007/IJCAI07-429.pdf), [N](https://zhuanlan.zhihu.com/p/goog_1420917004)[ever-Ending Language Learning (NELL)](http://link.zhihu.com/?target=http%3A//www.cs.cmu.edu/%7Eacarlson/papers/carlson-aaai10.pdf) 为代表的Extracted KBs，它们直接从上亿个网页中抽取实体关系三元组。与Freebase相比，这样得到的知识更加具有**多样性**，而它们的实体关系和实体更多的则是**自然语言**的形式，如“奥巴马出生在火奴鲁鲁。” 可以被表示为*(“Obama”, “was also born in”, “ Honolulu”),*当然，直接从网页中抽取出来的知识，也会存在一定的noisy，其精确度要低于Curated KBs。

Extracted KBs 知识库涉及到的两大关键技术是

1. **实体链指(Entity linking)** ，即将文档中的实体名字链接到知识库中特定的实体上。它主要涉及自然语言处理领域的两个经典问题**实体识别**(Entity Recognition) 与**实体消歧**(Entity Disambiguation)，简单地来说，就是要从文档中识别出人名、地名、机构名、电影等命名实体。并且，在不同环境下同一实体名称可能存在歧义，如苹果，我们需要根据上下文环境进行消歧。
2. **关系抽取 (Relation extraction)**，即将文档中的实体关系抽取出来，主要涉及到的技术有词性标注 (Part-of-Speech tagging, POS)，语法分析，依存关系树 (dependency tree) 以及构建SVM、最大熵模型等分类器进行关系分类等。

**什么是知识库问答**

知识库问答（knowledge base question answering,KB-QA）即给定**自然语言**问题，通过对问题进行语义理解和解析，进而利用知识库进行查询、推理得出答案。如下图所示



（注：该图片来自中科院刘康老师在*知识图谱与问答系统前沿技术研讨会*中的报告）

与对话系统、对话机器人的交互式对话不同，KB-QA具有以下特点：

1. **答案：**回答的答案是知识库中的**实体或实体关系**，或者no-answer（即该问题在KB中找不到答案），当然这里答案不一定唯一，比如 *中国的城市有哪些 。*而对话系统则回复的是自然语言句子，有时甚至需要考虑上下文语境。
2. **评价标准：**回召率 (Recall)，精确率 (Precision) ，F1-Score。而对话系统的评价标准以人工评价为主，以及BLEU和Perplexity。

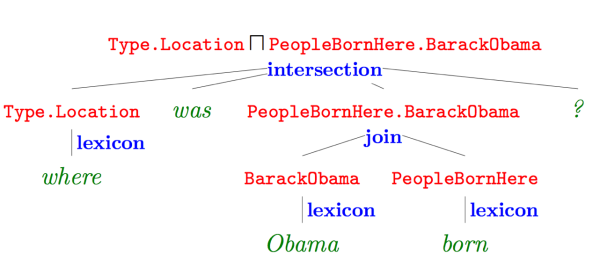
当我们在百度询问 *2016年奥斯卡最佳男主角*时，百度会根据知识库进行查询和推理，返回答案，这其实就是KB-QA的一个应用。

（注：该图片来自百度搜索）

**知识库问答的主流方法**

关于KB-QA的方法，个人认为，**传统**的主流方法可以分为三类：

* **语义解析（Semantic Parsing）**：该方法是一种偏linguistic的方法，主体思想是将**自然语言**转化为一系列形式化的**逻辑形式（logic form）**,通过对逻辑形式进行自底向上的解析，得到一种可以表达整个问题语义的逻辑形式，通过相应的查询语句（类似lambda-Caculus）在知识库中进行查询，从而得出答案。下图红色部分即逻辑形式，绿色部分*where was Obama born*为自然语言问题，蓝色部分为语义解析进行的相关操作，而形成的语义解析树的根节点则是最终的语义解析结果，可以通过查询语句直接在知识库中查询最终答案。



（注：该图片来自*Semantic Parsing on Freebase from Question-Answer Pairs*）

这里给出**语义解析方法**的一些代表论文

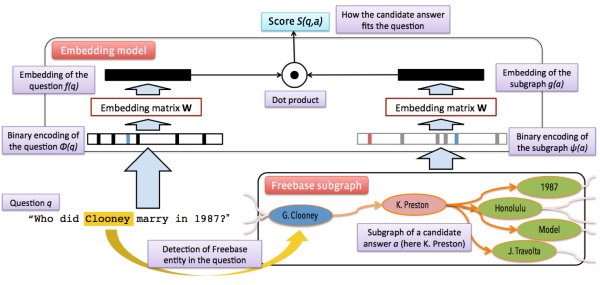
Berant J, Chou A, Frostig R, et al. *Semantic Parsing on Freebase from Question-Answer Pairs*[C]//EMNLP. 2013, 2(5): 6.

Cai Q, Yates A. *Large-scale Semantic Parsing via Schema Matching and Lexicon Extension*[C]//ACL (1). 2013: 423-433.

Kwiatkowski T, Choi E, Artzi Y, et al. *Scaling semantic parsers with on-the-fly ontology matching*[C]//In Proceedings of EMNLP. Percy. 2013.

Fader A, Zettlemoyer L, Etzioni O. *Open question answering over curated and extracted knowledge bases*[C]//Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2014: 1156-1165.

* **信息抽取（Information Extraction）**：该类方法通过提取问题中的实体，通过在知识库中查询该实体可以得到以该实体节点为中心的知识库子图，子图中的每一个节点或边都可以作为候选答案，通过观察问题依据某些规则或模板进行信息抽取，得到问题特征向量，建立分类器通过输入问题特征向量对候选答案进行筛选，从而得出最终答案。信息抽取的代表论文Yao X, Van Durme B. *Information Extraction over Structured Data: Question Answering with Freebase*[C]//ACL (1). 2014: 956-966.
* **向量建模（Vector Modeling）:**该方法思想和信息抽取的思想比较接近，根据问题得出候选答案，把问题和候选答案都映射为分布式表达（Distributed Embedding），通过训练数据对该分布式表达进行训练，使得问题和正确答案的向量表达的得分（通常以点乘为形式）尽量高，如下图所示。模型训练完成后则可根据候选答案的向量表达和问题表达的得分进行筛选，得出最终答案。



（注：该图片来自论文*Question answering with subgraph embeddings*）

**向量建模方法**的代表论文

Bordes A, Chopra S, Weston J. *Question answering with subgraph embeddings*[J]. arXiv preprint arXiv:1406.3676, 2014.

Yang M C, Duan N, Zhou M, et al. *Joint Relational Embeddings for Knowledge-based Question Answering[*C]//EMNLP. 2014, 14: 645-650.

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.

细心的朋友已经发现了，以上三种方法的代表论文都集中在13-14年。那么14年之后KB-QA的主流方法是什么呢？

随着**深度学习（Deep Learning）**在自然语言处理领域的飞速发展，从15年开始，开始涌现出一系列基于深度学习的KB-QA文章，通过深度学习对传统的方法进行提升，取得了较好的效果，比如：

使用**卷积神经网络**对**向量建模**方法进行提升：

Dong L, Wei F, Zhou M, et al. *Question Answering over Freebase with Multi-Column Convolutional Neural Networks*[C]//ACL (1). 2015: 260-269.

使用**卷积神经网络**对**语义解析**方法进行提升：

Yih S W, Chang M W, He X, et al. *Semantic parsing via staged query graph generation: Question answering with knowledge base*[J]. 2015.

（注 该paper来自微软，是ACL 2015年的**Outstanding paper**，也是目前KB-QA效果最好的paper之一）

使用**长短时记忆网络**（Long Short-Term Memory，LSTM），**卷积神经网络**（Convolutional Neural Networks，CNNs）进行实体关系分类：

Xu Y, Mou L, Li G, et al. *Classifying Relations via Long Short Term Memory Networks along Shortest Dependency Paths*[C]//EMNLP. 2015: 1785-1794.

Zeng D, Liu K, Lai S, et al. *Relation Classification via Convolutional Deep Neural Network*[C]//COLING. 2014: 2335-2344.（Best paper）

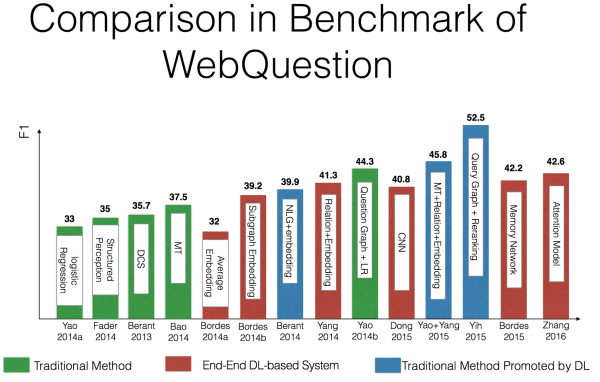
Zeng D, Liu K, Chen Y, et al. *Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks*[C]//EMNLP. 2015: 1753-1762.

使用**记忆网络**（Memory Networks），**注意力机制**（Attention Mechanism）进行KB-QA：

Bordes A, Usunier N, Chopra S, et al. *Large-scale simple question answering with memory networks*[J]. arXiv preprint arXiv:1506.02075, 2015.

Zhang Y, Liu K, He S, et al. *Question Answering over Knowledge Base with Neural Attention Combining Global Knowledge Information*[J]. arXiv preprint arXiv:1606.00979, 2016.

以上论文几乎都使用了**Freebase**作为knowledge base，并且在**WebQuestion**数据集上进行过测试，这里给出各种方法的效果对比图，给大家一个更加直观的感受。



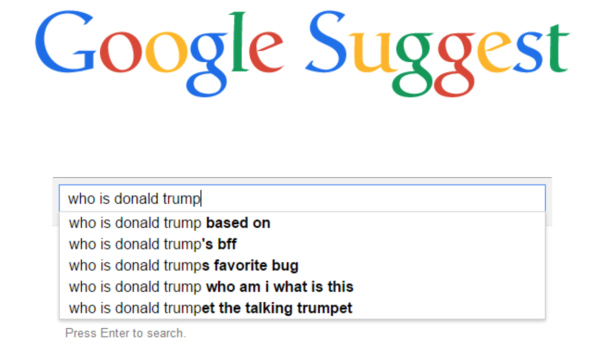
（注：该图片来自中科院刘康老师在*知识图谱与问答系统前沿技术研讨会*中的报告）

**知识库问答的数据集**

最后，我们再简单地介绍一下KB-QA问题的Benchmark数据集——WebQuestion。

该数据集由Berant J, Chou A, Frostig R, et al.在13年的论文*Semantic Parsing on Freebase from Question-Answer Pairs*中公开。

作者首先使用Google Suggest API获取以wh-word（what，who，why，where，whose...）为开头且只包含一个实体的问题，以*“where was Barack Obama born?”*作为问题图谱的起始节点，以Google Suggest API给出的建议作为新的问题，通过宽度优先搜索获取问题。具体来讲，对于每一个队列中的问题，通过对它删去实体，删去实体之前的短语，删去实体之后的短语形成3个新的query，将这三个新query放到google suggest中，每个query将生成5个候选问题，加入搜索队列，直到1M个问题被访问完。如下图所示

（注：该图片来自Google Suggest）

获取完问题后，随机选取100K个问题交给Amazon Mechanical Turk (AMT)的工人，让工人回答答案。注意，这里对答案进行了**限制**，让AMT的工人只能把答案设置为Freebase上的实体（entity），实体列表，值（value）或者no-answer。

最终，得到了5,810组问题答案对，其词汇表包含了4,525个词。并且，WebQuestion还提供了每个答案对应知识库的主题节点（topic node）。

可以看出WebQuestion的问题与freebase是不相关的，更加偏向自然语言，也更多样化。这里给出一些例子

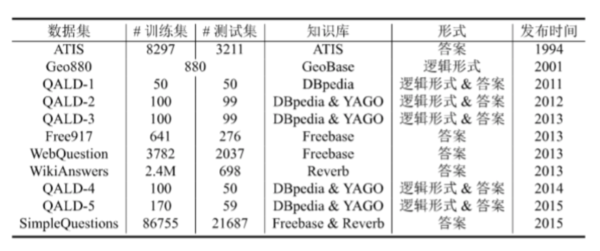
*“What is James Madison most famous for?”*

*“What movies does Taylor Lautner play in?”*

*“What music did Beethoven compose?”*

*“What kind of system of government does the United States have?”*

除了该数据集，这里再补充一些其他数据集的信息，如下图所示：

（注：该图片来自中科院刘康老师在*知识图谱与问答系统前沿技术研讨会*中的报告）

**下篇预告**

在接下来的文章中，我们会为大家对以上提到的主流方法和代表文章进行解读分析和讨论，一步步为大家揭开KB-QA的面纱。

下期，我们将从传统方法**语义解析**开始，介绍一个经典的baseline，来自斯坦福的Berant J, Chou A, Frostig R, et al. *Semantic Parsing on Freebase from Question-Answer Pairs*[C]//EMNLP. 2013, 2(5): 6.

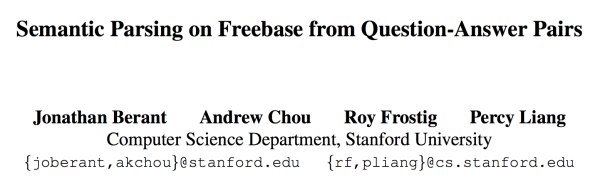
该文章是语义解析的经典代表作，也提供了Benchmark数据集WebQuestion，敬请期待。

**内容速览**

* 什么是语义解析（Semantic Parsing）
* 什么是逻辑形式（Logic Form）
* 语义解析KB-QA的方法框架
* 实验结果

本期我们从传统方法之一的**语义解析**（有时也被称为语义分析）开始，以一个经典的语义解析baseline方法为例，介绍语义解析如何进行KB-QA。该方法来自斯坦福Berant J, Chou A, Frostig R, et al. 的[*Semantic Parsing on Freebase from Question-Answer Pairs*](http://link.zhihu.com/?target=http%3A//171.64.67.140/pubs/semparseEMNLP13.pdf)*，*文章发表于2013年的EMNLP会议。

*注：语义解析的方法涉及到一些传统linguistic的知识，也是KB-QA三大传统方法中最难以理解的一种方法。这里由于篇幅有限，我们将不再对相应的linguistic知识进行详细介绍，为了方便大家理解，我们可能并未使用最标准的定义来解释linguistic相关的名词，而是给出方便大家理解的直觉上的解释。如果您对linguistic相关的知识感兴趣，可以关注我，之后我们将****开设专栏****，对传统linguistic的知识进行介绍和梳理，敬请期待。*



**什么是语义解析**

在[揭开知识库问答KB-QA的面纱1·简介篇](https://zhuanlan.zhihu.com/p/25735572)中我们谈到，知识库Freebase由大量的三元组组成，并且这些三元组的实体和实体关系都是形式化的语言，比如

*(BarackObama, PlaceOfBirth, Honolulu)*

给定一个自然语言的问题

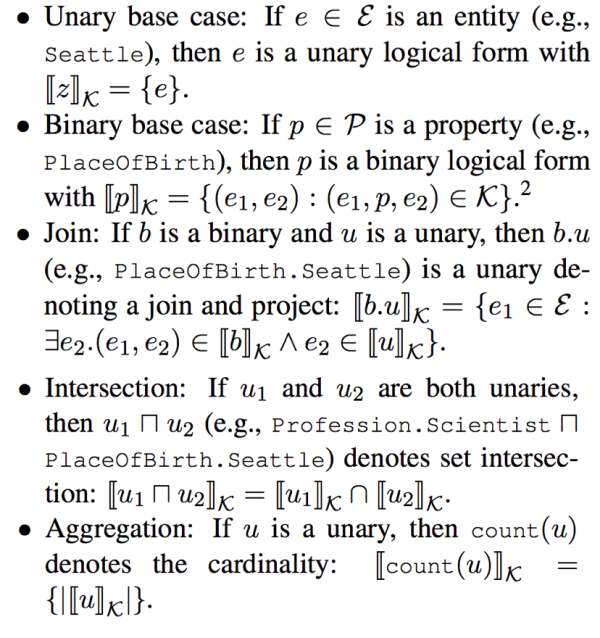
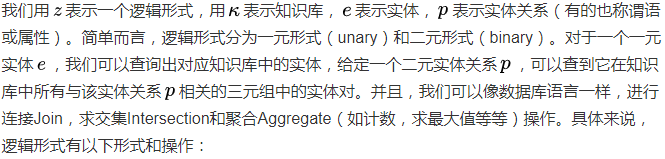
*“Where was Obama born？”*

我们面临的第一个挑战，就是如何建立问题到知识库的映射？

语义解析KB-QA的思路是通过对**自然语言**进行语义上的分析，转化成为一种能够让知识库“看懂”的**语义表示**，进而通过知识库中的知识，进行**推理（Inference）查询（Query）**，得出最终的答案。

简而言之，语义解析要做的事情，就是将自然语言的问题，转化为一种能够让知识库“看懂”的语义表示，这种语义表示即**逻辑形式（Logic Form）**。

**什么是逻辑形式**



有了上面的定义，我们就可以把一个自然语言问题表示为一个可以在知识库中进行查询的逻辑形式，比如对于问句

*“Number of dramas starring Tom Cruise?”*

它对应的逻辑形式是



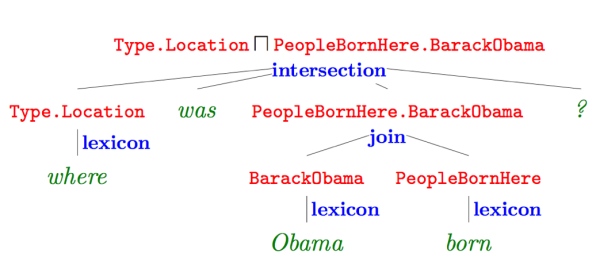
当自然语言问题转化为逻辑形式之后，通过相应的逻辑语言（转化为SPARQL query）查询知识库就可以得到答案。那么，语义解析要如何把自然语言问题正确地转化为相应的逻辑形式呢？

**语义解析KB-QA的方法框架**

语法解析的过程可以看作是自底向上构造语法树的过程，树的根节点，就是该自然语言问题最终的逻辑形式表达。整个流程可以分为两个步骤：

1. **词汇映射**：即构造底层的语法树节点。将单个自然语言短语或单词映射到知识库实体或知识库实体关系所对应的逻辑形式。我们可以通过构造一个**词汇表**（Lexicon）来完成这样的映射。
2. **构建**（Composition）：即自底向上对树的节点进行两两合并，最后生成根节点，完成语法树的构建。这一步有很多种方法，诸如构造大量手工规则，组合范畴语法（Combinatory Categorical Grammars，CCG）等等，而我们今天要讲的这篇论文，采用了最暴力的方法，即对于两个节点都可以执行上面所谈到的连接Join，求交Intersection，聚合Aggregate三种操作，以及这篇文章独创的桥接Bridging操作（桥接操作的具体方式稍后会提到）进行结点合并。显然，这种合并方式复杂度是指数级的，最终会生成很多棵语法树，我们需要通过对训练数据进行训练，训练一个分类器，对语法树进行筛选。

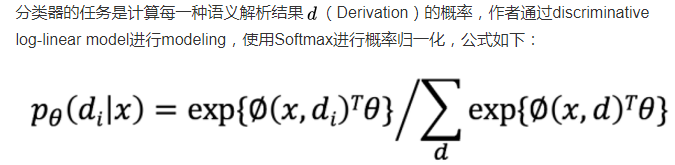
自然语言转化为逻辑形式的流程如下图所示：

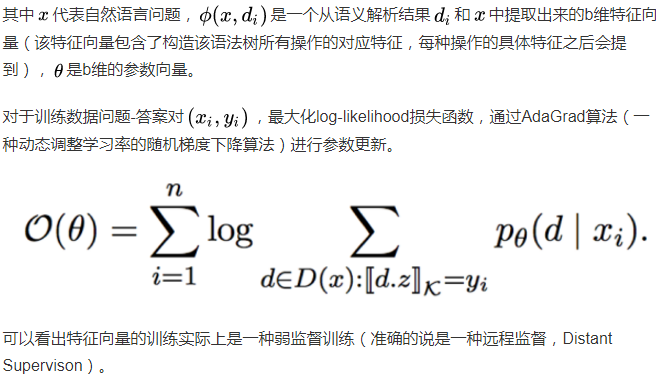


上图红色部分即**逻辑形式**，绿色部分*where was Obama born*为自然语言问题，蓝色部分为**词汇映射（Lexicon）**和**构建（Composition）**使用的操作，最终形成的语义解析树的根节点即语义解析结果。

接下来，我们还剩最后三个待解决的问题，如何训练分类器？如何构建词汇表？什么是桥接操作？

* 训练分类器





* 构建词汇表

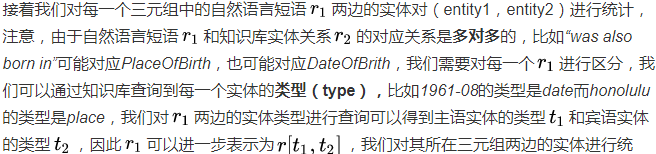
词汇表即自然语言与知识库实体或知识库实体关系的单点映射，这一操作也被称为**对齐（Alignment）。**我们知道自然语言实体到知识库实体映射相对比较简单，比如将*“Obama was also born in Honolulu.”*中的实体*Obama*映射为知识库中的实体*BarackObama*，可以使用一些简单的字符串匹配方式进行映射。

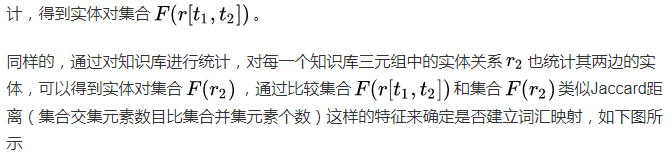
但是要将自然语言短语如*“was also born in”*映射到相应的知识库实体关系，如*PlaceOfBirth，*则较难通过字符串匹配的方式建立映射。怎么办呢？没错，我们可以进行统计。直觉上来说，在文档中，如果有较多的实体对（entity1，entity2）作为主语和宾语出现在was also born in的两侧，并且，在知识库中，这些实体对也同时出现在包含*PlaceOfBirth*的三元组中，那么我们可以认为*“was also born in”*这个短语可以和*PlaceOfBirth*建立映射。

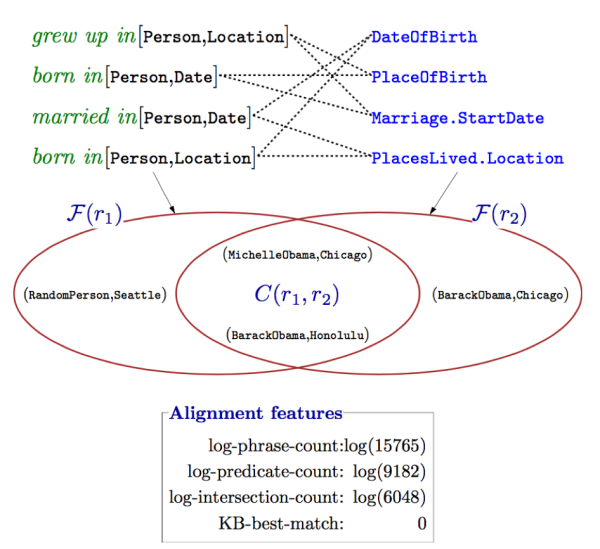
比如*（“Barack Obama”，“Honolulu”）,（“MichelleObama”，“Chicago”）*等实体对在文档中经常作为“was also *born in”*这个短语的主语和宾语，并且它们也都和实体关系*PlaceOfBirth*组成三元组出现在知识库中。

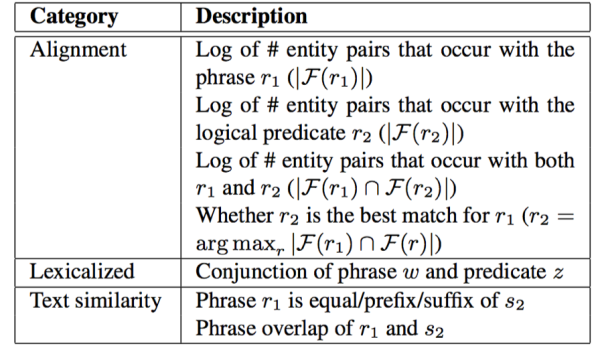
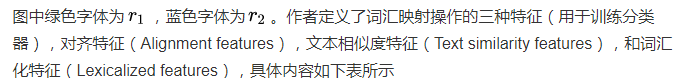
有了这样的直觉，我们再来看看这篇文章是怎么构建词汇表的，利用[ReVerbopen IE system](http://link.zhihu.com/?target=http%3A//reverb.cs.washington.edu)在[ClueWeb09](http://link.zhihu.com/?target=http%3A//lemurproject.org/clueweb09/FACC1/)（注：该数据集由卡耐基梅隆学校在09年构建，还有一个12年的版本，[ClueWeb12](http://link.zhihu.com/?target=http%3A//lemurproject.org/clueweb12/)）上抽取15millions个三元组构成一个数据集，如*(“Obama”, “was also born in”, “August 1961”)，*可以看出三元组的实体和关系都是自然语言的形式，取出其中的一个三元组子集，对里面的每一个三元组的主语实体和宾语实体通过字符匹配的方式替换为知识库的实体，并使用[SUTime](http://link.zhihu.com/?target=http%3A//nlp.stanford.edu/pubs/lrec2012-sutime.pdf)对数据进行归一化。

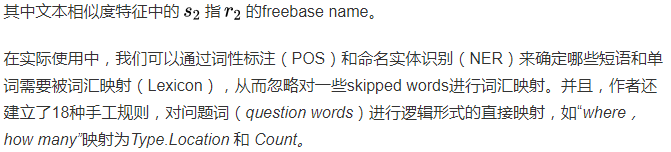
如*(“Obama”, “was also born in”, “August 1961”)*经过预处理后转化为 *(BarackObama, “was also born in”, 1961-08)*。



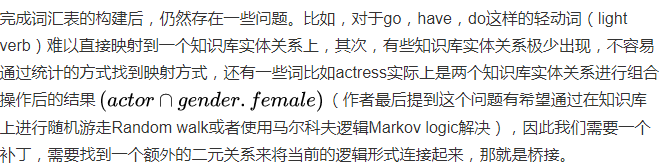


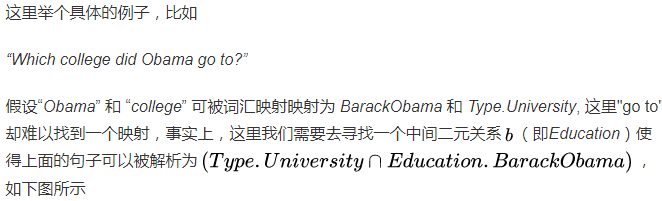


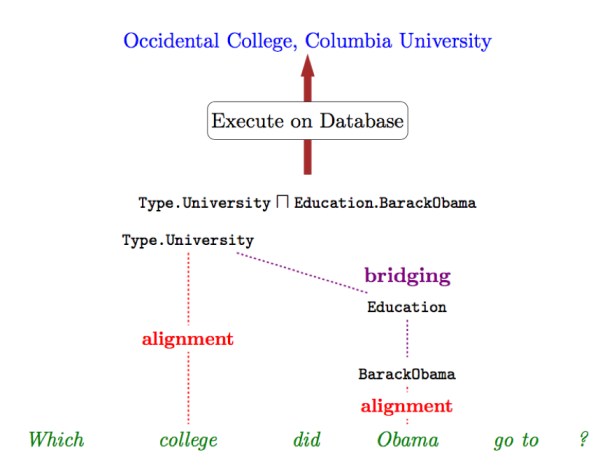


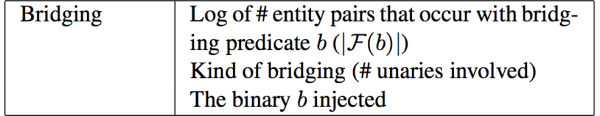
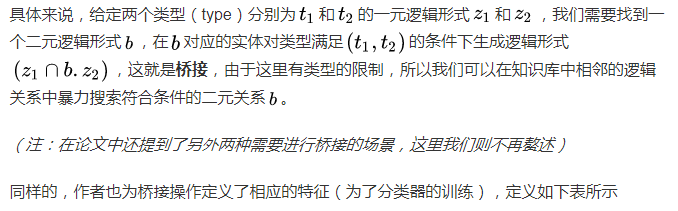


* **桥接操作**

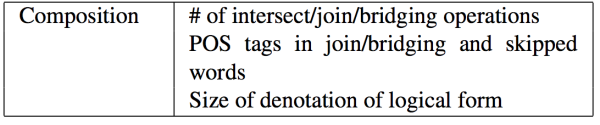








对于构建（composition）的其他三种操作，连接Join，求交集Intersection和聚合Aggregate，作者也定义了相应的特征（为了分类器的训练），如下表所示

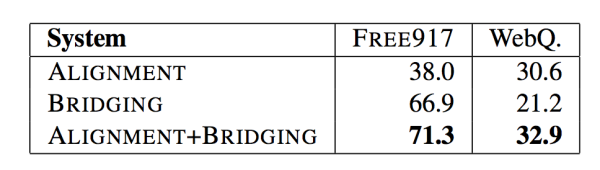


至此，语法树的构建，分类器的训练，和分类器的输入——特征向量 的构造方式我们都已经介绍完毕。最后我们再简单的介绍一下实验和实验结果。

* **实验结果**

由于语义解析树的构建方式是指数级的，因此，在训练和测试的时候，作者执行了标准的**自底向上的集束分析器（Beam-based bottom-up parser）**，如果不了解Beam search的同学，请[点击这里](http://link.zhihu.com/?target=https%3A//en.wikipedia.org/wiki/Beam_search)。在这篇论文之前，KB-QA流行的数据集是由[Cai and Yates (2013)](http://link.zhihu.com/?target=http%3A//www.anthology.aclweb.org/P/P13/P13-1042.pdf)构建的Free917，该数据集只包含了917组问题答案对，因此，作者构建了一个更大的benchmark数据集WebQuestion，包含了5810组问题答案对，该数据集的构建方式我在[揭开知识库问答KB-QA的面纱·简介篇](https://zhuanlan.zhihu.com/p/25735572)中进行了简单介绍。

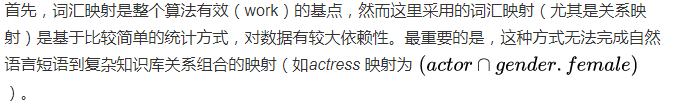
作者测试了仅使用Alignment和Bridging以及都使用下的正确率，如下表所示



作者该论文的语义解析器Sempre进行了开源，感兴趣的朋友可以[点击这里](http://link.zhihu.com/?target=http%3A//nlp.stanford.edu/software/sempre/)。

我们可以看出传统的语义解析方法还是存在大量的手工规则，也涉及到了一些linguistic的知识，对于没有传统NLP先验知识的朋友可能理解起来会稍微困难一些。

最后，让我们再**思考一下该方法有些什么缺陷**？



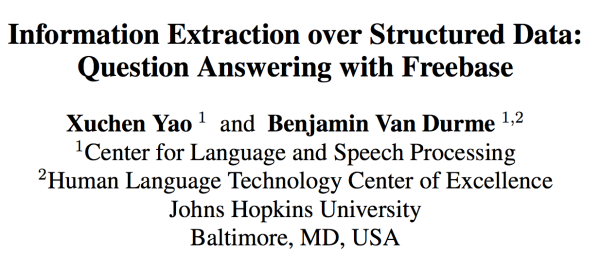
其次，在答案获取的过程中，通过远程监督学习训练分类器对语义树进行评分，注意，这里的语义树实际的组合方式是很多的，要训练这样一个强大的语义解析分类器，需要大量的训练数据。我们可以注意到，无论是Free917还是WebQuestion，这两个数据集的问题-答案对都比较少。

那么这些问题怎么解决呢？

在下一期中，我们将以2014年ACL的Yao X, Van Durme B. *Information Extraction over Structured Data: Question Answering with Freebase*[C]//ACL (1). 2014: 956-966. 这篇论文为例，介绍KB-QA的第二种传统方法——信息抽取，该方法在WebQuestion数据集上的F1-score相比本篇论文有一个较大的提升（大于10%）。

**内容速览**

* 你是如何通过知识库回答问题的
* 如何确定候选答案
* 如何对问题进行信息抽取
* 如何筛选候选答案
* 论文实验与总结



本期我们将介绍KB-QA传统方法之一的信息抽取（Information Extraction），我们以一个该方法的经典代表作为例，为大家进一步揭开知识库问答的面纱。该方法来自约翰·霍普金斯大学Yao X, Van Durme B.的 [*Information Extraction over Structured Data: Question Answering with Freebase*](http://link.zhihu.com/?target=http%3A//citeseerx.ist.psu.edu/viewdoc/download%3Fdoi%3D10.1.1.650.469%26rep%3Drep1%26type%3Dpdf)（文章发表于2014年的ACL会议）。

该类方法通过提取问题中的实体，通过在知识库中查询该实体可以得到以该实体节点为中心的知识库子图，子图中的每一个节点或边都可以作为候选答案。通过观察问题，依据某些规则或模板进行信息抽取，得到表征问题和候选答案特征的特征向量，建立分类器，通过输入特征向量对候选答案进行筛选，从而得出最终答案。

**你是如何回答问题的**

想一想，如果有人问你 *“what is the name of Justin Bieber brother?"* ，并且给你一个知识库，你会怎么去找答案？显然，这个问题的**主题**（Topic）词就是*Justin Bieber*，因此我们会去知识库搜索*Justin Bieber*这个实体，寻找与该实体相关的知识（此时相当于我们确定了答案的范围，得到了一些**候选答案**）。接下来，我们去寻找和实体关系brother相关的实体（事实上freebase里没有brother这个实体关系，而是sibling，我们需要进行一个简单的推理），最后得到答案。

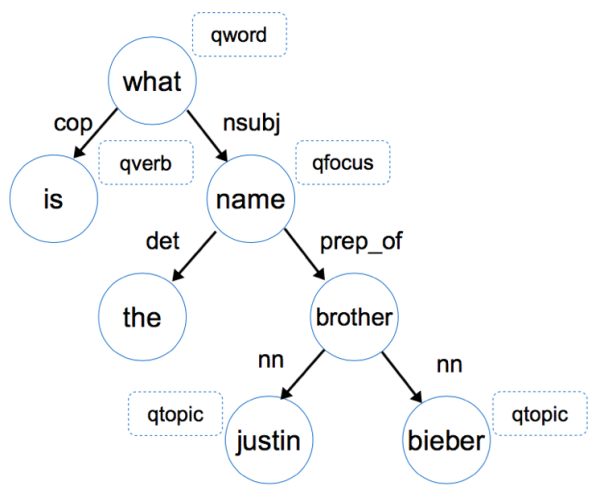
而信息抽取的方法，其灵感就是来自于刚才我们的这种思考方式。

**如何确定候选答案**

根据我们人的思维，当我们确定了问句中的主题词，我们就可以去知识库里搜索相应的知识，确定出候选答案。如果我们把知识库中的实体看作是图节点，把实体关系看作是边，那么知识库就是一个庞大的图，通过问句中的主题词可以找到它在知识库中对应的图节点，我们将该图节点相邻几跳（hop）范围内的节点和边抽取出来得到一个知识库的子图，这个子图作者称为**主题图（Topic graph）**，一般来说，这里的跳数一般为一跳或两跳，即与主题词对应的图节点在一条或两条边之内的距离。主题图中的节点，即是**候选答案**。接下来，我们需要观察问题，对问题进行信息抽取，获取能帮助我们在候选答案中筛选出正确答案的信息。

**如何对问题进行信息抽取**

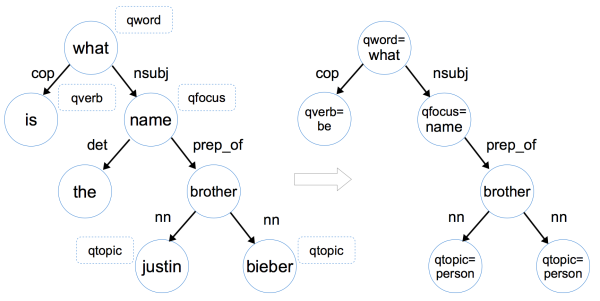
还是这个例子，让我们先放慢脚步，想想我们人类是怎么对这个问题进行信息抽取和推理的。首先，我们会潜意识地对这个句子结构进行分析，下图是 *“what is the name of Justin Bieber brother?"* 这个问句的**语法依存树（Dependency tree）**，如果你对依存树不了解，可以把它理解成是一种句子成分的形式化描述方式。



我们首先通过依存关系nsubj(what, name) 和 prep\_of(name, brother) 这两条信息知道答案是一个名字，而且这个名字和brother有关，当然我们此时还不能判断是否是人名。进一步，通过nn(brother, justin bieber)这条信息我们可以根据justin bieber是个人，推导出他的brother也是个人，综合前面的信息，我们最终推理出来我们的答案应该是个人名。（注：这里nsubj代表名词性主语，prep\_of代表of介词修饰，nn代表名词组合，如果你想了解更多tag，可以[点击这里](http://link.zhihu.com/?target=http%3A//www.hankcs.com/nlp/chinese-sentences-svo-java-extraction.html)）当确定了最终答案是一个人名，那么我们就很容易在备选答案中筛选出正确答案了。

我们刚才进行的步骤，实际上就是在对问题进行**信息抽取**，接下来，让我们看看这篇文章具体是怎么实施的。

首先我们要提取的第一个信息就是**问题词**（question word，记作***qword***）, 例如 who, when, what, where, how, which, why, whom, whose，它是问题的一个明显特征 。第二个关键的信息，就是**问题焦点**（question fucus, 记作***qfocus***），这个词暗示了答案的类型，比如name/time/place，我们直接将问题词***qword***相关的那个名词抽取出来作为*qfocus*，在这个例子中，what name中的name就是***qfocus***。第三个我们需要的信息，就是这个问题的主题词（word topic，记作***qtopic***），在这个句子里Justin Bieber就是***qtopic***，这个词能够帮助我们找到freebase中相关的知识，我们可以通过命名实体识别（Named Entity Recognition，NER）来确定主题词，需要注意的是，一个问题中可能存在多个主题词。最后，第四个我们需要提取的特征，就是问题的中心动词（question verb ，记作***qverb***）,动词能够给我们提供很多和答案相关的信息，比如play，那么答案有可能是某种球类或者乐器。我们可以通过词性标注（Part-of-Speech，POS）确定*qverb。*通过对问题提取 **问题词*qword***，**问题焦点*qfocus***，**问题主题词*qtopic***和**问题中心动词qverb**这四个问题特征，我们可以将该问题的依存树转化为**问题图（Question Graph）**，如下图所示

具体来说，将依存树转化为问题图进行了三个操作

1）将问题词*qword*，问题焦点*qfocus*，问题主题词*qtopic*和问题中心动词*qverb*加入相对应的节点中，如what -> qword=what。

2）如果该节点是命名实体，那就把该节点变为命名实体形式，如justin -> qtopic=person （justin对应的命名实体形式是person）。这一步的目的是因为数据中涉及到的命名实体名字太多了，这里我们只需要区分它是人名 地名 还是其他类型的名字即可。

3）删除掉一些不重要的叶子节点，如限定词（determiner，如a/the/some/this/each等），介词（preposition）和标点符号（punctuation）。

**从依存树到问题图的转换，实质是就是对问题进行信息抽取，提取出有利于寻找答案的问题特征，删减掉不重要的信息。**

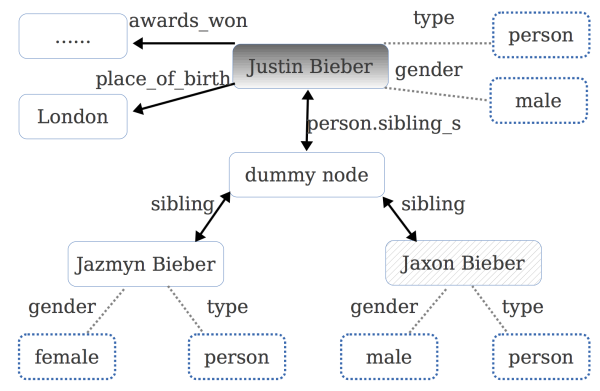
**如何构建特征向量对候选答案进行分类**

在候选答案中找出正确答案，实际上是一个**二分类问题**（判断每个候选答案是否是正确答案），我们使用训练数据问题-答案对，训练一个分类器来找到正确答案。那么分类器的输入特征向量怎么构造和定义呢？

特征向量中的每一维，对应一个问题-候选答案特征。每一个问题-候选答案特征由问题特征中的一个特征，和候选答案特征的一个特征，**组合（combine）而成**。

**问题特征：**我们从问题图中的每一条边e(s,t)，抽取4种问题特征：s，t，s|t，和s|e|t。如对于边prep\_of(qfocus=name，brother)，我们可以抽取这样四个特征：*qfocus=what，brother，qfocus=what|brother*和 *qfocus=what|prep\_of|brother。*

**候选答案特征：**对于主题图中的每一个节点，我们都可以抽取出以下特征：该节点的所有**关系**（relation，记作rel），和该节点的所有**属性**（property，如type/gender/age）。（注：我在[揭开知识库问答KB-QA的面纱1·简介篇](https://zhuanlan.zhihu.com/p/25735572)中对知识库中属性和关系的区别进行了讲解）对于Justin Bieber 这个topic我们可以在知识库找到它对应的主题图，如下图所示：

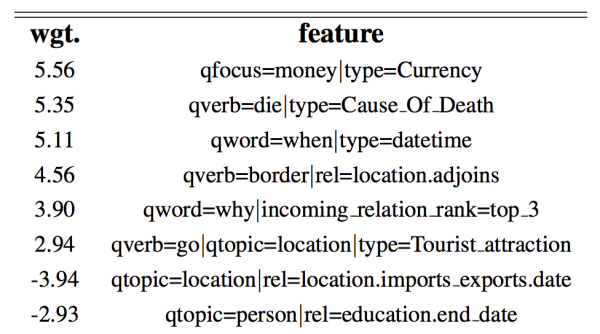


（注：图中虚线表示属性，实线表示关系，虚线框即属性值，实现框为topic node。在知识库中，如果同一个topic节点的同一个关系对应了多个实体，如Justin Bieber的preon.sibing\_s关系可能对应多个实体，那么freebase中会设置一个虚拟的**dummy node**，来连接所有相关的实体）

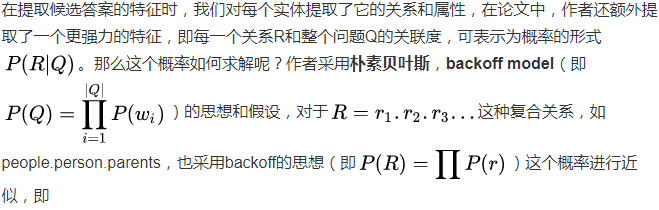
例如，对于Jaxon Bieber这个topic节点，我们可以提取出这些特征：*gender=male，type=person，rel=sibling 。*可以看出关系和属性都刻画了这个候选答案的特征，对判断它是否是正确答案有很大的帮助。

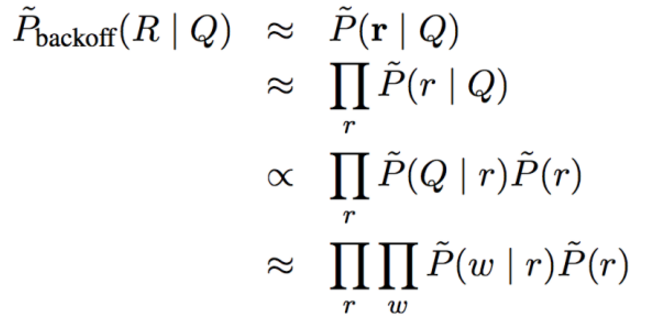
**问题-候选答案特征**：每一个问题-候选答案特征由问题特征中的一个特征和候选答案特征中的一个特征，组合（combine）而成（组合记作 | ）。我们希望一个**关联度较高**的问题-候选答案特征有较高的权重，比如对于问题-候选答案特征 *qfocus=money|node type=currency*（注意，这里*qfocus=money*是来自问题的特征，而*node type=currency*则是来自候选答案的特征），我们希望它的权重较高，而对于问题-候选答案特征*qfocus=money|node type=person*我们希望它的权重较低。

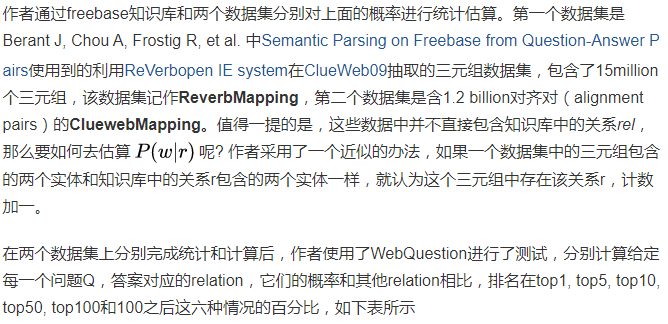
接下来我们用WebQuestion作为训练数据，使用[Stanford CoreNLP](http://link.zhihu.com/?target=http%3A//stanfordnlp.github.io/CoreNLP/)帮助我们对问题进行信息抽取。训练集中约有3000个问题，每个问题对应的主题图约含1000个节点，共计有3 million的节点和7 million种问题-候选答案特征，作者用带L1正则化的逻辑回归(logistic regression)作为分类器，训练每种问题-候选答案特征的权值（L1正则化的逻辑回归很适合处理这种稀疏的特征向量，作者表示其效果好于感知机Percptron和支持向量机SVM）。 训练完毕后，得到了3万个非零的特征，下表列出了部分特征和它相应的权值，可以看出问题特征和候选答案特征相关度较高时，其权值较高。

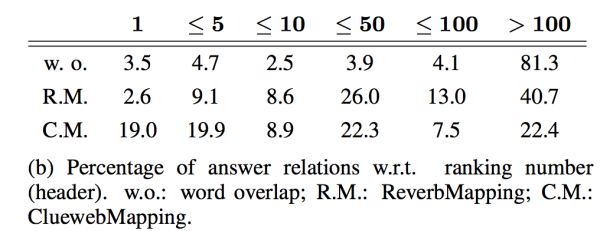


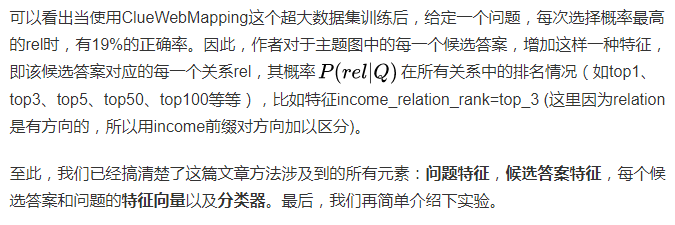
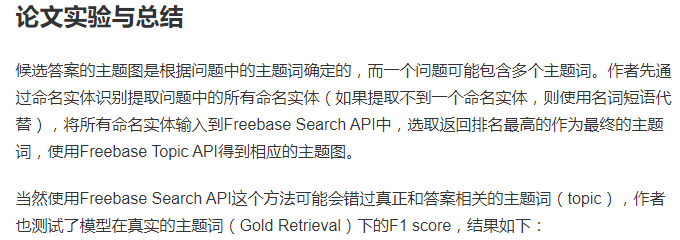
因此，在使用的时候，对于每一个候选答案，我们抽取出它的特征（假设有k个特征）后，再和问题中的每一个特征**两两结合**（假设有m个特征），那么我们就得到了k\*m个问题-候选答案特征，因此我们的输入向量就是一个k\*m-hot（即k\*m维为1，其余维为0）的3万维向量。

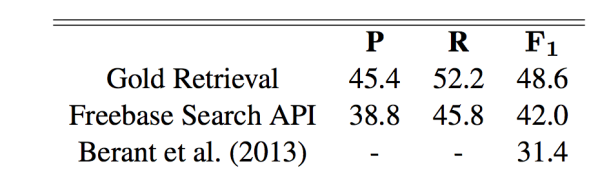


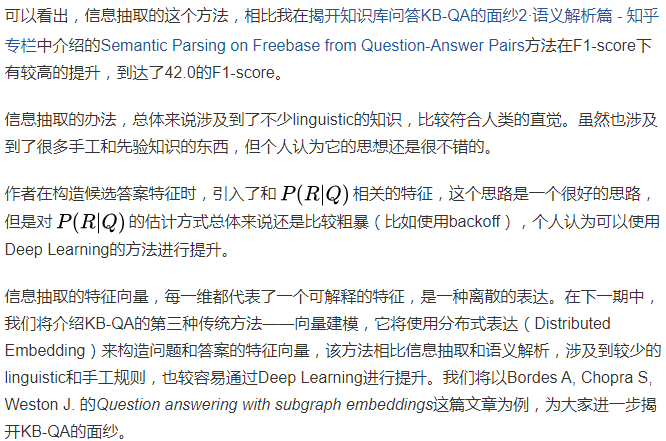






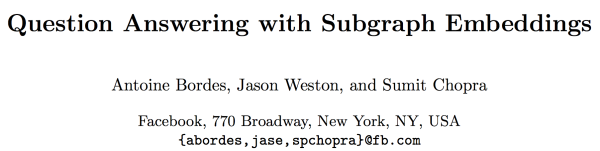




**揭开知识库问答KB-QA的面纱4·向量建模篇**

**内容速览**

* 向量建模的核心思想
* 如何用分布式表达表示答案和问题
* 如何训练分布式表达
* 论文实验与总结



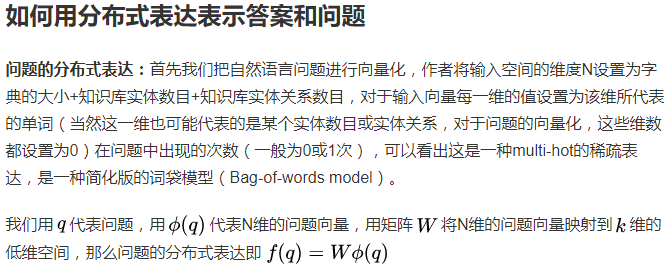
本期我们将介绍KB-QA传统方法之一的向量建模（Vector Modeling），我们以一个该方法的经典代表作为例，为大家进一步揭开知识库问答的面纱。该方法来自Facebook公司Bordes A, Chopra S, Weston J的论文 [Question answering with subgraph embeddings](http://link.zhihu.com/?target=https%3A//arxiv.org/abs/1406.3676)（文章发表于2014年的EMNLP会议）。

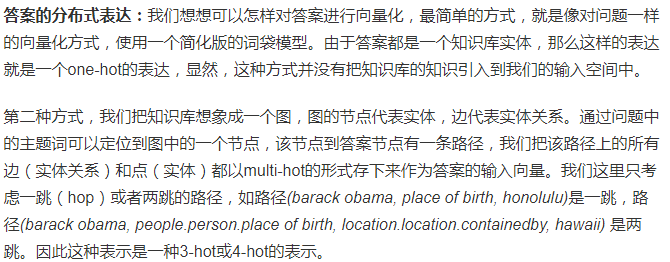
**向量建模的核心思想**

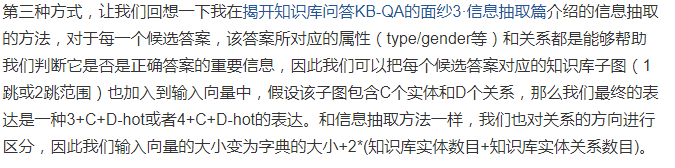
向量建模方法的思想和信息抽取的思想比较接近。首先根据问题中的主题词在知识库中确定候选答案。把问题和候选答案都映射到一个**低维空间**，得到它们的**分布式表达**（Distributed Embedding），通过训练数据对该分布式表达进行训练，使得问题向量和它对应的正确答案向量在低维空间的关联得分（通常以点乘为形式）尽量高。当模型训练完成后，则可根据候选答案的向量表达和问题表达的得分进行筛选，找出得分最高的作为最终答案。

此时，你的心中可能会出现两个问题，一是如何将问题和答案映射到低维空间，显然我们不能仅仅将自然语言的问题和答案进行映射，还要将知识库里的知识也映射到这个低维空间中（否则我们就只是在做QA而非KB-QA了）。第二个问题是，如果做过类似工作（one-shot，imgae caption，word embedding等）的朋友应该知道，使用这种方法是需要**大量数据**去训练这个低维空间的分布式表达的，而KB-QA中的benchmark数据集WebQuestion只含有5800多个问题答案对，这样的数据是难以训练好这种表达的。

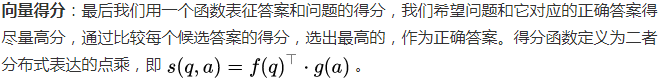
接下来，就让我们带着这两个问题，一起看看作者是怎么解决的。



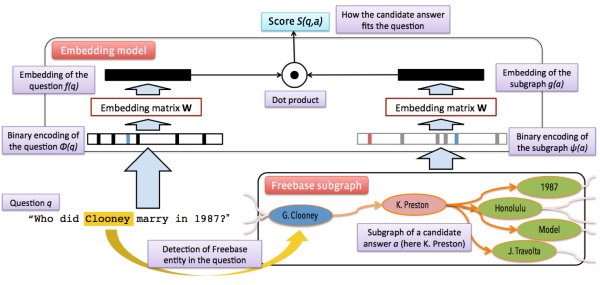


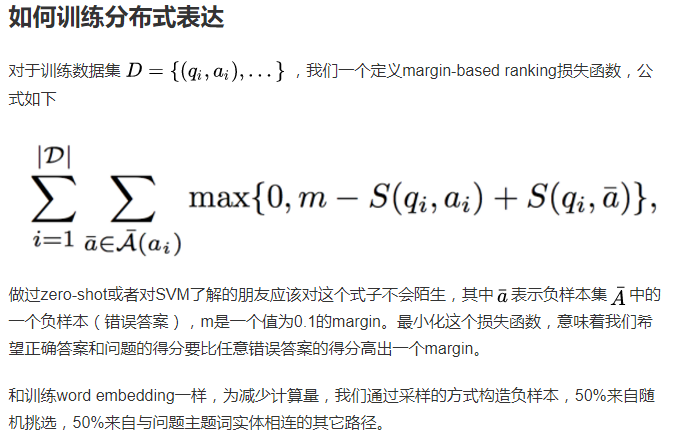


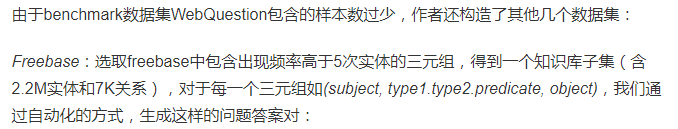




上述整个流程如下图所示



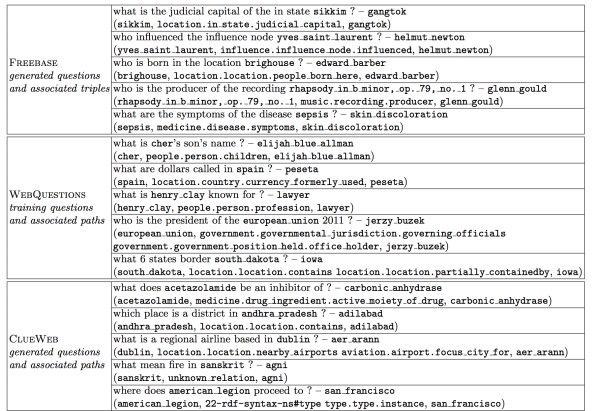




*Quesiton：“What is the predicate of the****type2subject?****” Answer：****object***

*ClueWeb Extractions：*由于 *Freebase*的三元组都是形式化语言，并不贴近自然语言，我们也用同样的方式将ClueWeb上提取出的三元组*(subject, “text string”, object)*通过少量模板作同样的变换（作者提取了2M对三元组）。

这样我们就在WebQuestion数据集的基础上，得到了一个新的扩展数据集，该数据集的例子如下表所示：

可以看出，扩增版的数据集，问题大多数都是自动构造的，缺乏多样性和真实性。怎么办呢？我们希望**训练数据中问题的分布式表达尽量贴近它所类似的真实问题的分布式表达**。因此，作者在WikiAnswers中抓取了2.2M问题（不含答案），通过问题的分类标签，将它们分为了350k个类簇（可以理解为每个类簇里的自然语言问题它所表达的意思是一样的）。如下表所示

