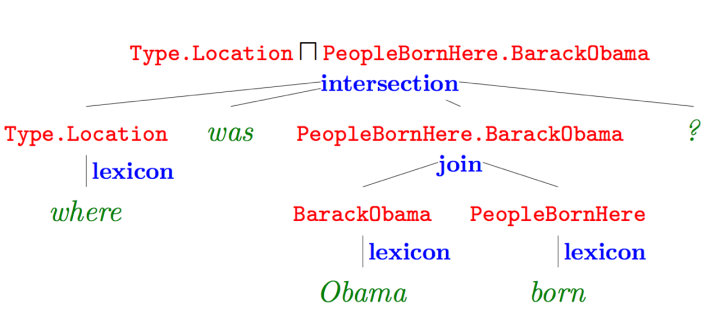
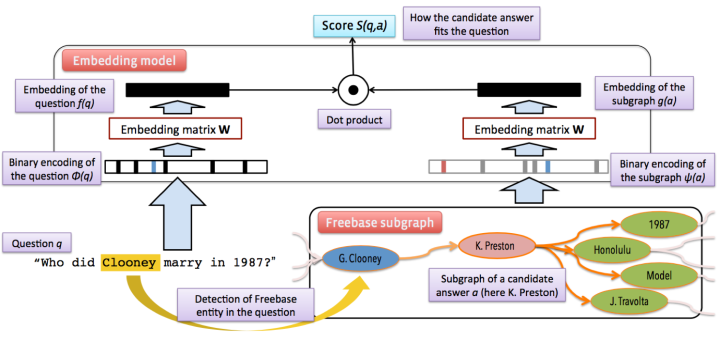
## 语义解析

语义解析（Semantic Parsing）：该方法是一种偏语言学的方法，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）

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。

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）

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）

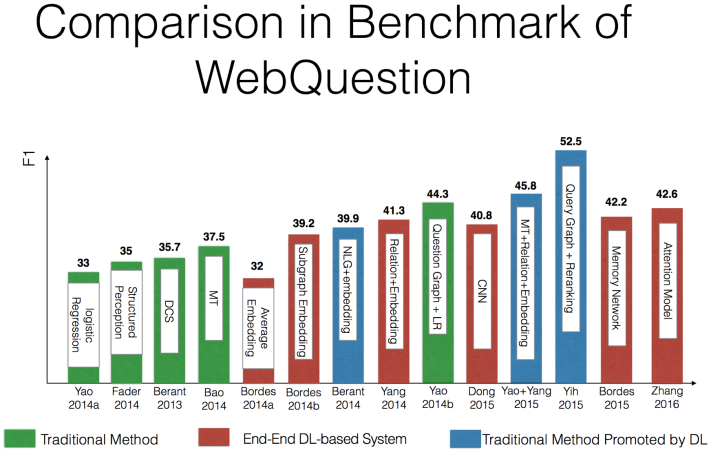
Zeng D, Liu K, Chen Y, et al. *Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks*[C]//EMNLP. 2015: 1753-1762.

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。



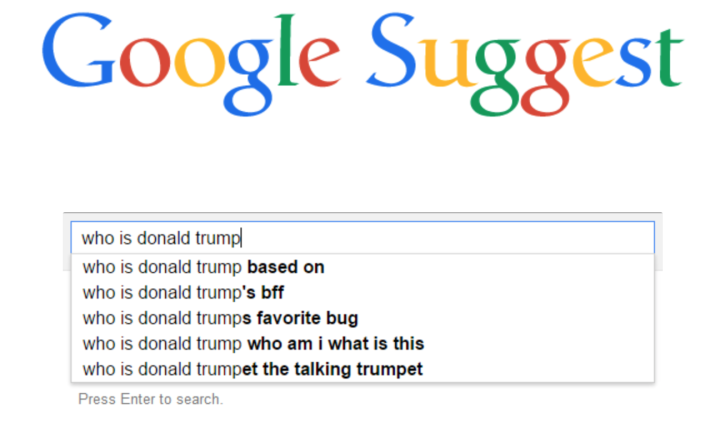
（注：该图片来自中科院刘康老师在知识图谱与问答系统前沿技术研讨会中的报告）

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

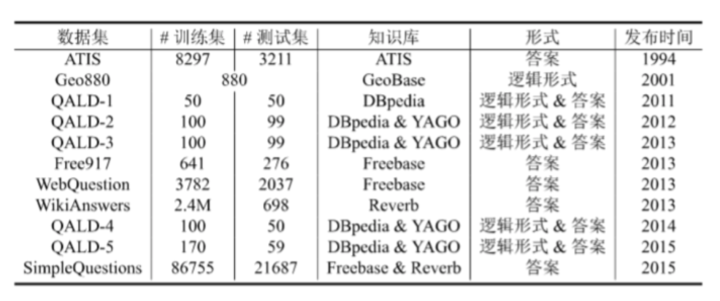
“What is James Madison most famous for?”

“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：

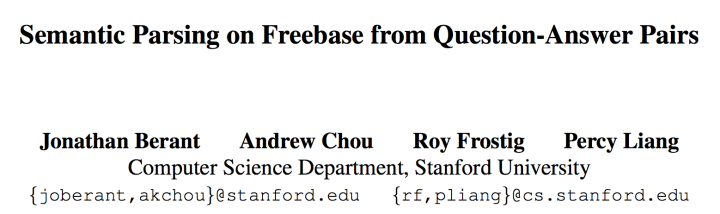


（注：该图片来自中科院刘康老师在*知识图谱与问答系统前沿技术研讨会*中的报告）

## 语义解析

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，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

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）**。

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？

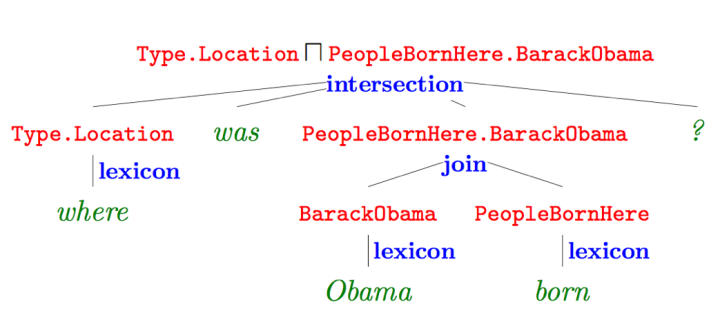
语义解析KB-QA的方法框架

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：

**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。

**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：

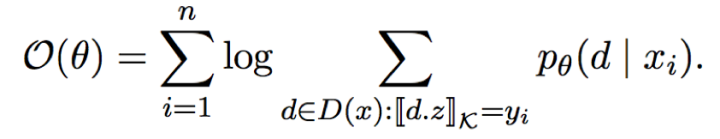
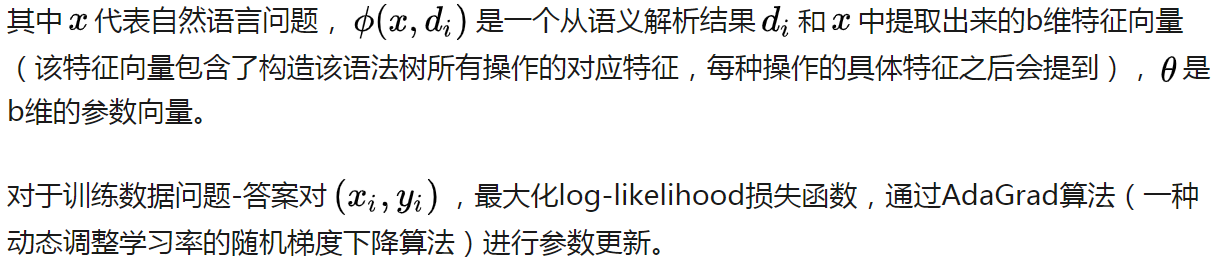
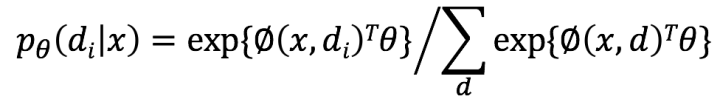


上图红色部分即**逻辑形式**，绿色部分*where was Obama born*为自然语言问题，蓝色部分为**词汇映射（Lexicon）**和**构建（Composition）**使用的操作，最终形成的语义解析树的根节点即语义解析结果。

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

分类器的任务是计算每一种语义解析结果d（Derivation）的概率，作者通过discriminative log-linear model进行modeling，使用Softmax进行概率归一化，公式如下：



可以看出特征向量的训练实际上是一种弱监督训练（准确的说是一种远程监督，Distant Supervison）。

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。

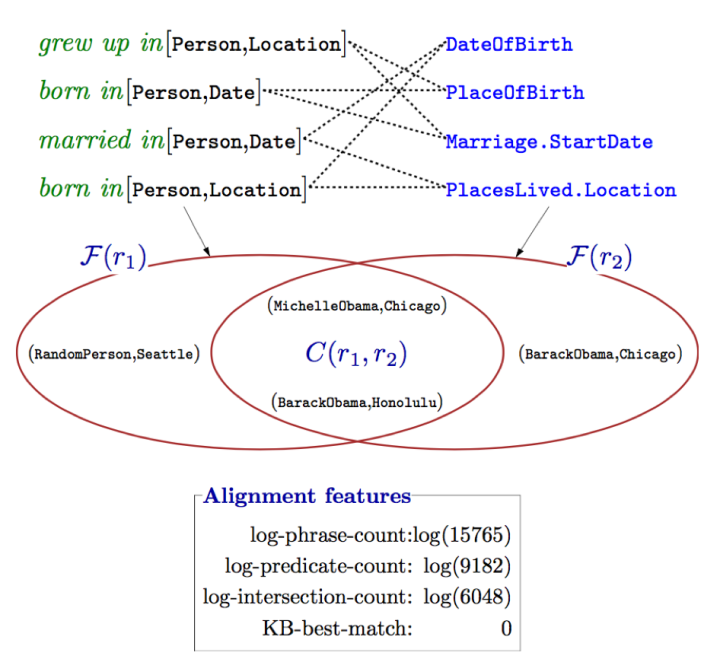
比如*（“Barack Obama”，“Honolulu”）,（“MichelleObama”，“Chicago”）*等实体对在文档中经常作为“was also *born in”*这个短语的主语和宾语，并且它们也都和实体关系*PlaceOfBirth*组成三元组出现在知识库中。

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在ClueWeb09（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）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。

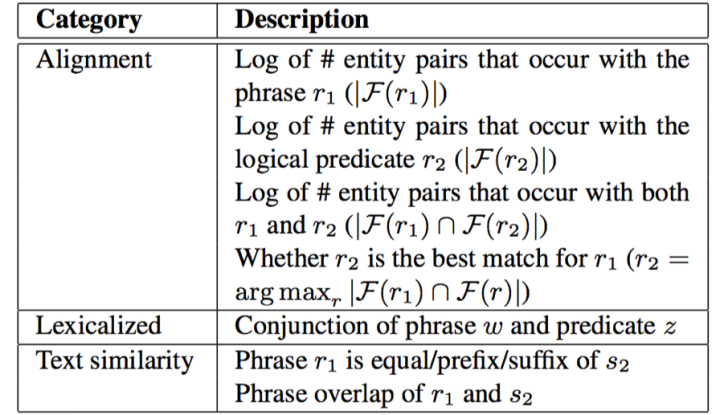
如(“Obama”, “was also born in”, “August 1961”) 经过预处理后转化为 (BarackObama, “was also born in”, 1961-08)。

接着我们对每一个三元组中的自然语言短语r1两边的实体对（entity1，entity2）进行统计，注意，由于自然语言短语r1和知识库实体关系r2的对应关系是**多对多**的，比如“was also born in”可能对应PlaceOfBirth，也可能对应DateOfBrith，我们需要对每一个r1进行区分，我们可以通过知识库查询到每一个实体的**类型（type），**比如1961-08的类型是date而honolulu的类型是place，我们对r1两边的实体类型进行查询可以得到主语实体的类型t1和宾语实体的类型t2，因此r1可以进一步表示为r1[t1,t2]，我们对其所在三元组两边的实体进行统计，得到实体对集合F(r1[t1,t2])。

同样的，通过对知识库进行统计，对每一个知识库三元组中的实体关系r2也统计其两边的实体，可以得到实体对集合F(r2)，通过比较集合F(r1[t1,t2])和集合F(r2)类似Jaccard距离（集合交集元素数目比集合并集元素个数）这样的特征来确定是否建立词汇映射，如下图所示



图中绿色字体为r1，蓝色字体为r2。作者定义了词汇映射操作的三种特征（用于训练分类器），对齐特征（Alignment features），文本相似度特征（Text similarity features），和词汇化特征（Lexicalized features），具体内容如下表所示

其中文本相似度特征中的s2指r2的freebase name。

在实际使用中，我们可以通过词性标注（POS）和命名实体识别（NER）来确定哪些短语和单词需要被词汇映射（Lexicon），从而忽略对一些skipped words进行词汇映射。并且，作者还建立了18种手工规则，对问题词（*question words*）进行逻辑形式的直接映射，如“*where，how many”*映射为*Type.Location* 和 *Count。*

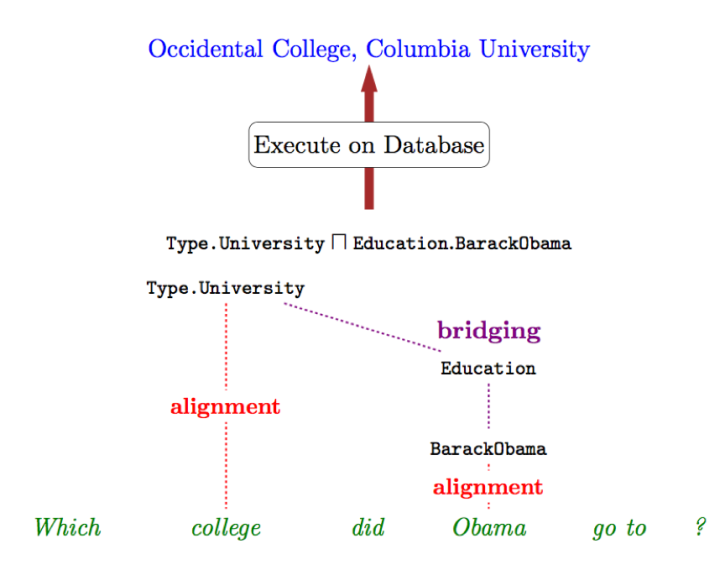
桥接操作

完成词汇表的构建后，仍然存在一些问题。比如，对于go，have，do这样的轻动词（light verb）难以直接映射到一个知识库实体关系上，其次，有些知识库实体关系极少出现，不容易通过统计的方式找到映射方式，还有一些词比如actress实际上是两个知识库实体关系进行组合操作后的结果（作者最后提到这个问题有希望通过在知识库上进行随机游走Random walk或者使用马尔科夫逻辑Markov logic解决），因此我们需要一个补丁，需要找到一个额外的二元关系来将当前的逻辑形式连接起来，那就是桥接。

这里举个具体的例子，比如

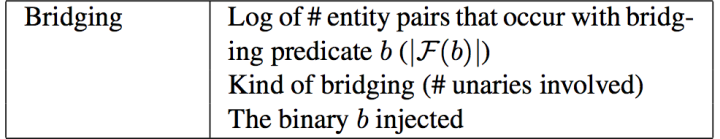
*“Which college did Obama go to?”*

假设“*Obama*” 和 “*college*” 可被词汇映射映射为 *BarackObama*和 *Type.University*, 这里"go to" 却难以找到一个映射，事实上，这里我们需要去寻找一个中间二元关系b（即*Education*）使得上面的句子可以被解析为，如下图所示

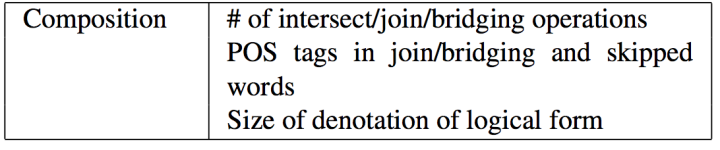


具体来说，给定两个类型（type）分别为t1和t2的一元逻辑形式z1和z2，我们需要找到一个二元逻辑形式b，在b对应的实体对类型满足(t1,t2)的条件下生成逻辑形式，这就是**桥接**，由于这里有类型的限制，所以我们可以在知识库中相邻的逻辑关系中暴力搜索符合条件的二元关系b。

同样的，作者也为桥接操作定义了相应的特征（为了分类器的训练），定义如下表所示



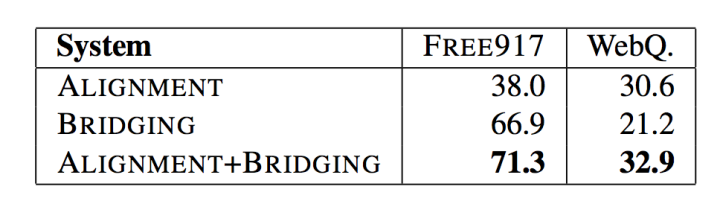
对于构建（composition）的其他三种操作，连接Join，求交集Intersection和聚合Aggregate，作者也定义了相应的特征（为了分类器的训练），如下表所示

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）。在这篇论文之前，KB-QA流行的数据集是由Cai and Yates (2013)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.

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



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**？

首先，词汇映射是整个算法有效（work）的基点，然而这里采用的词汇映射（尤其是关系映射）是基于比较简单的统计方式，对数据有较大依赖性。最重要的是，这种方式无法完成自然语言短语到复杂知识库关系组合的映射（如actress 映射为）。

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%）。