## 信息抽取

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*（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。

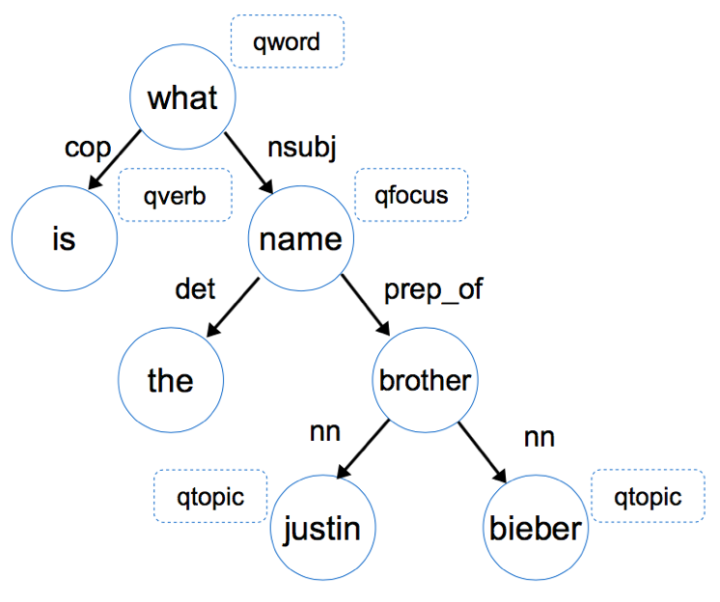
而信息抽取的方法，其灵感就是来自于刚才我们的这种思考方式。

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

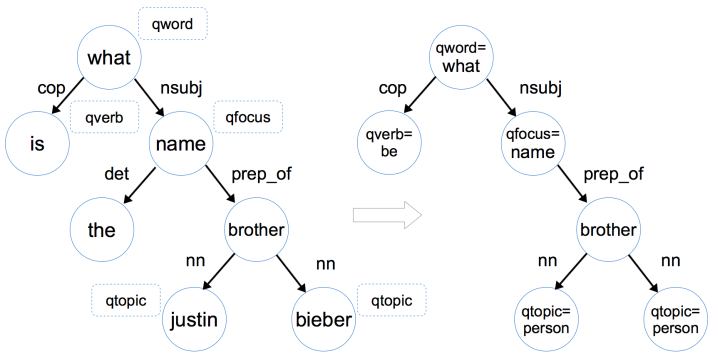
还是这个例子，让我们先放慢脚步，想想我们人类是怎么对这个问题进行信息抽取和推理的。首先，我们会潜意识地对这个句子结构进行分析，下图是 *“what is the name of Justin Bieber brother?"* 这个问句的**语法依存树（Dependency tree）**，如果你对依存树不了解，可以把它理解成是一种句子成分的形式化描述方式。



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）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。**

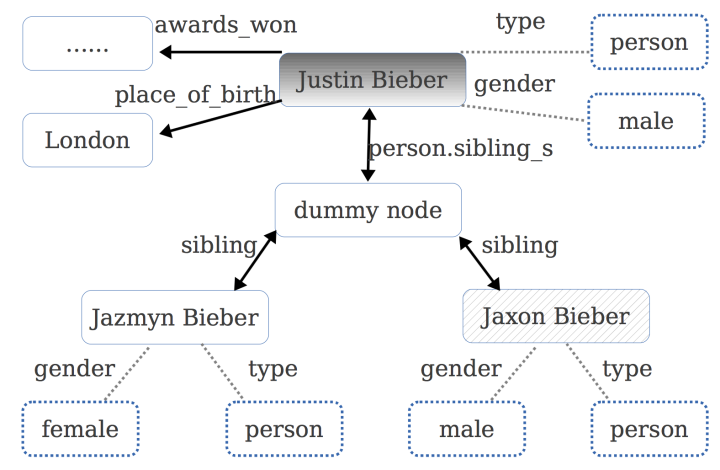
如何构建特征向量对候选答案进行分类

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）。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：

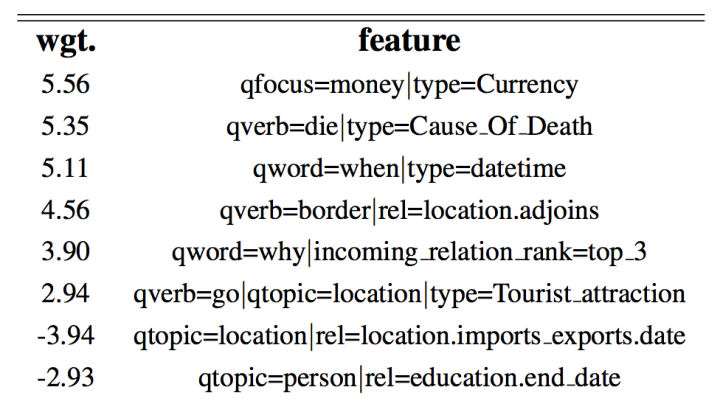


（注：图中虚线表示属性，实线表示关系，虚线框即属性值，实现框为topic node。在知识库中，如果同一个topic节点的同一个关系对应了多个实体，如Justin Bieber的preon.sibing\_s关系可能对应多个实体，那么freebase中会设置一个虚拟的**dummy node**，来连接所有相关的实体）

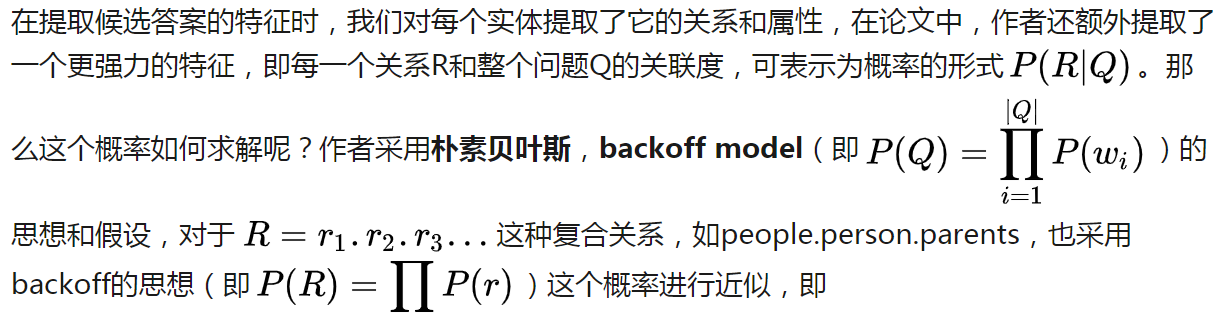
例如，对于Jaxon Bieber这个topic节点，我们可以提取出这些特征：*gender=male，type=person，rel=sibling 。*可以看出关系和属性都刻画了这个候选答案的特征，对判断它是否是正确答案有很大的帮助。

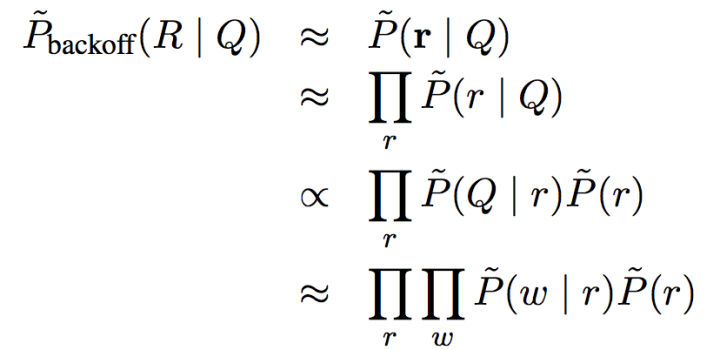
问题-候选答案特征：每一个问题-候选答案特征由问题特征中的一个特征和候选答案特征中的一个特征，组合（combine）而成（组合记作 | ）。我们希望一个关联度较高的问题-候选答案特征有较高的权重，比如对于问题-候选答案特征 qfocus=money|node type=currency（注意，这里qfocus=money是来自问题的特征，而node type=currency则是来自候选答案的特征），我们希望它的权重较高，而对于问题-候选答案特征qfocus=money|node type=person我们希望它的权重较低。

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。

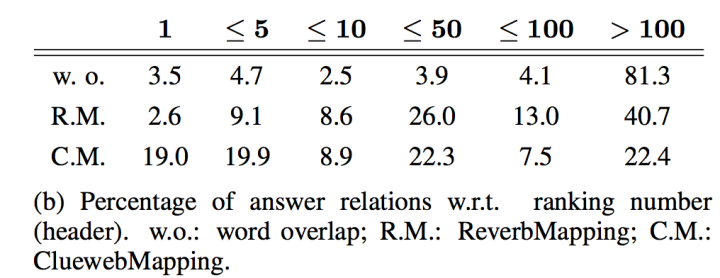




作者通过freebase知识库和两个数据集分别对上面的概率进行统计估算。第一个数据集是Berant J, Chou A, Frostig R, et al. 中[Semantic Parsing on Freebase from Question-Answer Pairs](http://link.zhihu.com/?target=http%3A//171.64.67.140/pubs/semparseEMNLP13.pdf)使用到的利用[ReVerbopen IE system](http://link.zhihu.com/?target=http%3A//reverb.cs.washington.edu)在[ClueWeb09](http://link.zhihu.com/?target=http%3A//lemurproject.org/clueweb09/FACC1/)抽取的三元组数据集，包含了15million个三元组，该数据集记作ReverbMapping，第二个数据集是含1.2 billion对齐对（alignment pairs）的CluewebMapping。

值得一提的是，这些数据中并不直接包含知识库中的关系*rel*，那么要如何去估算呢? 作者采用了一个近似的办法，如果一个数据集中的三元组包含的两个实体和知识库中的关系r包含的两个实体一样，就认为这个三元组中存在该关系r，计数加一。

在两个数据集上分别完成统计和计算后，作者使用了WebQuestion进行了测试，分别计算给定每一个问题Q，答案对应的relation，它们的概率和其他relation相比，排名在top1, top5, top10, top50, top100和100之后这六种情况的百分比，如下表所示



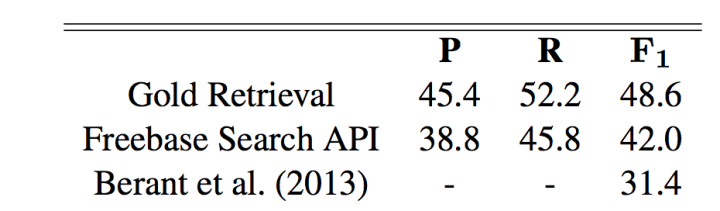
可以看出当使用ClueWebMapping这个超大数据集训练后，给定一个问题，每次选择概率最高的rel时，有19%的正确率。因此，作者对于主题图中的每一个候选答案，增加这样一种特征，即该候选答案对应的每一个关系rel，其概率在所有关系中的排名情况（如top1、top3、top5、top50、top100等等），比如特征income\_relation\_rank=top\_3 (这里因为relation是有方向的，所以用income前缀对方向加以区分)。

至此，我们已经搞清楚了这篇文章方法涉及到的所有元素：**问题特征**，**候选答案特征**，每个候选答案和问题的**特征向量**以及**分类器**。最后，我们再简单介绍下实验。

论文实验与总结

候选答案的主题图是根据问题中的主题词确定的，而一个问题可能包含多个主题词。作者先通过命名实体识别提取问题中的所有命名实体（如果提取不到一个命名实体，则使用名词短语代替），将所有命名实体输入到Freebase Search API中，选取返回排名最高的作为最终的主题词，使用Freebase Topic API得到相应的主题图。

当然使用Freebase Search API这个方法可能会错过真正和答案相关的主题词（topic），作者也测试了模型在真实的主题词（Gold Retrieval）下的F1 score，结果如下：

可以看出，信息抽取的这个方法，相比我在[揭开知识库问答KB-QA的面纱2·语义解析篇 - 知乎专栏](https://zhuanlan.zhihu.com/p/25759682)中介绍的[Semantic Parsing on Freebase from Question-Answer Pairs](http://link.zhihu.com/?target=http%3A//171.64.67.140/pubs/semparseEMNLP13.pdf)方法在F1-score下有较高的提升，到达了42.0的F1-score。

信息抽取的办法，总体来说涉及到了不少linguistic的知识，比较符合人类的直觉。虽然也涉及到了很多手工和先验知识的东西，但个人认为它的思想还是很不错的。

作者在构造候选答案特征时，引入了和相关的特征，这个思路是一个很好的思路，但是对的估计方式总体来说还是比较粗暴（比如使用backoff），个人认为可以使用Deep Learning的方法进行提升。

信息抽取的特征向量，每一维都代表了一个可解释的特征，是一种离散的表达。在下一期中，我们将介绍KB-QA的第三种传统方法——向量建模，它将使用分布式表达（Distributed Embedding）来构造问题和答案的特征向量，该方法相比信息抽取和语义解析，涉及到较少的linguistic和手工规则，也较容易通过Deep Learning进行提升。我们将以Bordes A, Chopra S, Weston J. 的Question answering with subgraph embeddings这篇文章为例，为大家进一步揭开KB-QA的面纱。