目录

**[知识图谱技术原理介绍](#_Toc2247_WPSOffice_Level1)** **[2](#_Toc2247_WPSOffice_Level1)**

**[知识图谱的表示和在搜索中的展现形式](#_Toc23527_WPSOffice_Level1)** **[2](#_Toc23527_WPSOffice_Level1)**

**[【三大搜索引擎关于姚明的知识卡片(略)】](#_Toc25425_WPSOffice_Level1)** **[3](#_Toc25425_WPSOffice_Level1)**

**[知识图谱的构建](#_Toc26840_WPSOffice_Level1)** **[4](#_Toc26840_WPSOffice_Level1)**

**[1. 知识图谱的规模](#_Toc19427_WPSOffice_Level1)** **[4](#_Toc19427_WPSOffice_Level1)**

**[2. 知识图谱的数据来源](#_Toc13072_WPSOffice_Level1)** **[5](#_Toc13072_WPSOffice_Level1)**

[a) 百科类数据](#_Toc23527_WPSOffice_Level2) [5](#_Toc23527_WPSOffice_Level2)

[b) 结构化数据](#_Toc25425_WPSOffice_Level2) [6](#_Toc25425_WPSOffice_Level2)

[c) 半结构化数据挖掘AVP](#_Toc26840_WPSOffice_Level2) [7](#_Toc26840_WPSOffice_Level2)

[d) 通过搜索日志进行实体和实体属性等挖掘](#_Toc19427_WPSOffice_Level2) [8](#_Toc19427_WPSOffice_Level2)

**[3. 从抽取图谱到知识图谱](#_Toc19218_WPSOffice_Level1)** **[9](#_Toc19218_WPSOffice_Level1)**

[a) 实体对齐](#_Toc13072_WPSOffice_Level2) [9](#_Toc13072_WPSOffice_Level2)

[b) 知识图谱schema构建](#_Toc19218_WPSOffice_Level2) [10](#_Toc19218_WPSOffice_Level2)

[c) 不一致性的解决](#_Toc24351_WPSOffice_Level2) [11](#_Toc24351_WPSOffice_Level2)

**[4. 知识图谱上的挖掘](#_Toc24351_WPSOffice_Level1)** **[11](#_Toc24351_WPSOffice_Level1)**

[a) 推理](#_Toc172_WPSOffice_Level2) [12](#_Toc172_WPSOffice_Level2)

[b) 实体重要性排序](#_Toc27928_WPSOffice_Level2) [12](#_Toc27928_WPSOffice_Level2)

[c) 相关实体挖掘](#_Toc13839_WPSOffice_Level2) [13](#_Toc13839_WPSOffice_Level2)

**[5. 知识图谱的更新和维护](#_Toc172_WPSOffice_Level1)** **[13](#_Toc172_WPSOffice_Level1)**

[a) Type和Collection的关系](#_Toc27689_WPSOffice_Level2) [13](#_Toc27689_WPSOffice_Level2)

[b) 结构化站点包装器的维护](#_Toc9167_WPSOffice_Level2) [13](#_Toc9167_WPSOffice_Level2)

[c) 知识图谱的更新频率](#_Toc2094_WPSOffice_Level2) [14](#_Toc2094_WPSOffice_Level2)

[d) 众包(Crowdsourcing)反馈机制](#_Toc11659_WPSOffice_Level2) [14](#_Toc11659_WPSOffice_Level2)

**[知识图谱在搜索中的应用](#_Toc27928_WPSOffice_Level1)** **[14](#_Toc27928_WPSOffice_Level1)**

**[1. 查询理解](#_Toc13839_WPSOffice_Level1)** **[14](#_Toc13839_WPSOffice_Level1)**

**[2. 问题回答](#_Toc27689_WPSOffice_Level1)** **[15](#_Toc27689_WPSOffice_Level1)**

**[总结](#_Toc9167_WPSOffice_Level1)** **[15](#_Toc9167_WPSOffice_Level1)**

**知识图谱技术原理介绍**

近两年来，随着Linking Open Data[1] 等项目的全面展开，语义Web数据源的数量激增，大量RDF数据被发布。互联网正从仅包含网页和网页之间超链接的文档万维网(Document Web)转变成包含大量描述各种实体和实体之间丰富关系的数据万维网(Data Web)。在这个背景下，Google、百度和搜狗等搜索引擎公司纷纷以此为基础构建知识图谱，分别为Knowledge Graph、知心和知立方，来改进搜索质量，从而拉开了语义搜索的序幕。下面我将从以下几个方面来介绍知识图谱：知识图谱的表示和在搜索中的展现形式，知识图谱的构建和知识图谱在搜索中的应用等，从而让大家有机会了解其内部的技术实现和各种挑战。

知识图谱的表示和在搜索中的展现形式

正如Google的辛格博士在介绍知识图谱时提到的：“The world is not made of strings , but is made of things.”，知识图谱旨在描述真实世界中存在的各种实体或概念。其中，每个实体或概念用一个全局唯一确定的ID来标识，称为它们的标识符(identifier)。每个属性-值对(attribute-value pair，又称AVP)用来刻画实体的内在特性，而关系(relation)用来连接两个实体，刻画它们之间的关联。知识图谱亦可被看作是一张巨大的图，图中的节点表示实体或概念，而图中的边则由属性或关系构成。上述图模型可用W3C提出的资源描述框架RDF[2] 或属性图(property graph)[3] 来表示。知识图谱率先由Google提出，以提高其搜索的质量。

为了更好地理解知识图谱，我们先来看一下其在搜索中的展现形式，即知识卡片(又称Knowledge Card)。知识卡片旨在为用户提供更多与搜索内容相关的信息。更具体地说，知识卡片为用户查询中所包含的实体或返回的答案提供详细的结构化摘要。从某种意义来说，它是特定于查询(query specific)的知识图谱。例如，当在搜索引擎中输入“姚明”作为关键词时，我们发现搜索结果页面的右侧原先用于置放广告的地方被知识卡片所取代。广告被移至左上角，而广告下面则显示的是传统的搜索结果，即匹配关键词的文档列表。这个布局上的微调也预示着各大搜索引擎在提高用户体验和直接返回答案方面的决心。

【三大搜索引擎关于姚明的知识卡片(略)】

虽说三大搜索引擎在知识卡片的排版和内容展现上略有不同，但是它们都列出了姚明的身高、体重、民族等属性信息。此外，它们均包含“用户还搜索了”或“其他人还搜”的功能来展现相关的人物。该功能允许用户去浏览其他与姚明相关的人物的详细信息。细心的读者也发现Google在其知识卡片中也展示了很多与姚明相关的图片，以图文并茂的方式来展示姚明的方方面面。百度则结合了百度风云榜的信息，列出了姚明的类别(体坛人物)及其百度指数(今日排名和今日搜索热度等信息)。在搜索结果页面的左上角(在图中未给出)，百度还展示了其特有的专题搜索，包含了与姚明相关的百科、图片、微博、新闻、音乐、贴吧和视频等七大类的结果，基本涵盖了用户最基本的需求。搜狗在列出与姚明相关的百科、图片，电影和最新相关消息等专题的同时，其知识卡片额外显示了诸如“主持电视节目”、“效力篮球队”、“人物关系”等各种细粒度的语义关系。当遇到含有歧义的用户查询时，知识卡片还会列出其他可能的查询目标对象。在上面的例子中，搜狗还列出了一项“您是否要找”的功能，列出一位也叫姚明的一级作曲家。该功能用于去歧义，在显示最相关实体的同时也给出其他可能的对象，达到去歧义的作用。当搜索“李娜”或“长城”时，Google和百度也在其知识卡片下方展现了类似的功能。除了给出著名网球运动员李娜和万里长城之外，它们还列出歌手李娜和长城汽车供用户选择和浏览。更值得一提的是，当在搜狗知立方中输入“姚明的老婆的女儿的身高”如此复杂的查询时，其会直接返回其女儿的姓名(姚沁蕾)以及其身高(110cm)，并给出推理说明“叶莉的女儿是姚沁蕾”。如此详实的说明不仅为返回的答案提供了很好的解释，从另一个侧面也展示了知识图谱的强大，其不仅能识别出运动员姚明，也能抽取出关系“老婆”和“女儿”和属性“身高”等信息。当我们将查询修改为“姚明的妻子的女儿的身高”时，依然返回相同的结果，这也意味着知识图谱知道“妻子”和“老婆”代表相同的含义。

通过上述的介绍，大家应该对知识图谱的表示以及其在搜索中的展现形式有了更深的了解。接着，我将介绍知识图谱的构建以及如何在搜索中应用知识图谱返回相应的知识卡片以及答案。

知识图谱的构建

1. 知识图谱的规模

据不完全统计，Google知识图谱到目前为止包含了5亿个实体和35亿条事实(形如实体-属性-值，和实体-关系-实体)。其知识图谱是面向全球的，因此包含了实体和相关事实的多语言描述。不过相比占主导的英语外，仅包含其他语言(如中文)的知识图谱的规模则小了很多。与此不同的是，百度和搜狗主要针对中文搜索推出知识图谱，其知识库中的知识也主要以中文来描述，其规模略小于Google的。

2. 知识图谱的数据来源

为了提高搜索质量，特别是提供如对话搜索和复杂问答等新的搜索体验，我们不仅要求知识图谱包含大量高质量的常识性知识，还要能及时发现并添加新的知识。在这种背景下，知识图谱通过收集来自百科类站点和各种垂直站点的结构化数据来覆盖大部分常识性知识。这些数据普遍质量较高，更新比较慢。而另一方面，知识图谱通过从各种半结构化数据(形如HTML表格)抽取相关实体的属性-值对来丰富实体的描述。此外，通过搜索日志(query log)发现新的实体或新的实体属性从而不断扩展知识图谱的覆盖率。相比高质量的常识性知识，通过数据挖掘抽取得到的知识数据更大，更能反映当前用户的查询需求并能及时发现最新的实体或事实，但其质量相对较差，存在一定的错误。这些知识利用互联网的冗余性在后续的挖掘中通过投票或其他聚合算法来评估其置信度，并通过人工审核加入到知识图谱中。

a) 百科类数据

维基百科[4] ，通过协同编辑，已经成为最大的在线百科全书，其质量与大英百科媲美。可以通过以下方式来从维基百科中获取所需的内容：通过文章页面(Article Page)抽取各种实体;通过重定向页面(Redirect Page)获得这些实体的同义词(又称Synonym);通过去歧义页面(Disambiguation Page)和内链锚文本(Internal Link Anchor Text)获得它们的同音异义词(又称Homonym);通过概念页面(Category Page)获得各种概念以及其上下位(subclass)关系;通过文章页面关联的开放分类抽取实体所对应的类别;通过信息框(Infobox)抽取实体所对应的属性-值对和关系-实体对。类似地，从百度百科和互动百科抽取各种中文知识来弥补维基百科中文数据不足的缺陷。此外，Freebase[5] 是另一个重要的百科类的数据源，其包含超过3900万个实体(其称为Topics)和18亿条事实，规模远大于维基百科。对比之前提及的知识图谱的规模，我们发现仅Freebase一个数据源就构成了Google知识图谱的半壁江山。更为重要的是，维基百科所编辑的是各种词条，这些词条以文章的形式来展现，包含各种半结构化信息，需要通过事先制定的规则来抽取知识;而Freebase则直接编辑知识，包括实体及其包含的属性和关系，以及实体所属的类型等结构化信息。因此，不需要通过任何抽取规则即可获得高质量的知识。虽然开发Freebase的母公司MetaWeb于2010年被Google收购，Freebase还是作为开放的知识管理平台独立运行。所以百度和搜狗也将Freebase加入到其知识图谱中。

b) 结构化数据

除了百科类的数据，各大搜索引擎公司在构建知识图谱时，还考虑其他结构化数据。其中，LOD项目在发布各种语义数据的同时，通过owl:sameAs将新发布的语义数据中涉及的实体和LOD中已有数据源所包含的潜在同一实体进行关联，从而实现了手工的实体对齐(entity alignment)。LOD不仅包括如DBpedia[6] 和YAGO[7] 等通用语义数据集，还包括如MusicBrainz[8] 和DrugBank[9] 等特定领域的知识库。因此，Google等通过整合LOD中的(部分)语义数据提高知识的覆盖率，尤其是垂直领域的各种知识。此外，Web上存在大量高质量的垂直领域站点(如电商网站，点评网站等)，这些站点被称为Deep Web[10]。它们通过动态网页技术将保存在数据库中的各种领域相关的结构化数据以HTML表格的形式展现给用户。各大搜索引擎公司通过收购这些站点或购买其数据来进一步扩充其知识图谱在特定领域的知识。这样做出于三方面原因：其一、大量爬取这些站点的数据会占据大量带宽，导致这些站点无法被正常访问;其二、爬取全站点数据可能会涉及知识产权纠纷;最后，相比静态网页的爬取，Deep Web爬虫需要通过表单填充(Form Filling)技术来获取相关内容，且解析这些页面中包含的结构化信息需要额外的自动化抽取算法，具体细节在下一节描述。

c) 半结构化数据挖掘AVP

虽然从Deep Web爬取数据并解析其中所包含的结构化信息面临很大的挑战，各大搜索引擎公司仍在这方面投入了大量精力。一方面，Web上存在大量长尾的结构化站点，这些站点提供的数据与最主流的相关领域站点所提供的内容具有很强的互补性，因此对这些长尾站点进行大规模的信息抽取(尤其是实体相关的属性-值对的抽取)对于知识图谱所含内容的扩展是非常有价值的。另一方面，中文百科类的站点(如百度百科等)的结构化程度远不如维基百科，能通过信息框获得AVP的实体非常稀少，大量属性-值对隐含在一些列表或表格中。一个切实可行的做法是构建面向站点的包装器(Site-specific Wrapper)。其背后的基本思想是：一个Deep Web站点中的各种页面由统一的程序动态生成，具有类似的布局和结构。利用这一点，我们仅需从当前待抽取站点采样并标注几个典型详细页面(Detailed Pages)，利用这些页面通过模式学习算法(Pattern Learning)自动构建出一个或多个以类Xpath表示的模式，然后将其应用在该站点的其他详细页面中从而实现自动化的AVP抽取。对于百科类站点，我们可以将具有相同类别的页面作为某个“虚拟”站点，并使用类似的方法进行实体AVP的抽取。自动学习获得的模式并非完美，可能会遗漏部分重要的属性，也可能产生错误的抽取结果。为了应对这个问题，搜索引擎公司往往通过构建工具来可视化这些模式，并人工调整或新增合适的模式用于抽取。此外，通过人工评估抽取的结果，将那些抽取结果不令人满意的典型页面进行再标注来更新训练样本，从而达到主动学习(Active Learning)的目的。

d) 通过搜索日志进行实体和实体属性等挖掘

搜索日志是搜索引擎公司积累的宝贵财富。一条搜索日志形如<查询，点击的页面链接，时间戳>。通过挖掘搜索日志，我们往往可以发现最新出现的各种实体及其属性，从而保证知识图谱的实时性。这里侧重于从查询的关键词短语和点击的页面所对应的标题中抽取实体及其属性。选择查询作为抽取目标的意义在于其反映了用户最新最广泛的需求，从中能挖掘出用户感兴趣的实体以及实体对应的属性。而选择页面的标题作为抽取目标的意义在于标题往往是对整个页面的摘要，包含最重要的信息。据百度研究者的统计，90%以上的实体可以在网页标题中被找到。为了完成上述抽取任务，一个常用的做法是：针对每个类别，挑选出若干属于该类的实体(及相关属性)作为种子(Seeds)，找到包含这些种子的查询和页面标题，形成正则表达式或文法模式。这些模式将被用于抽取查询和页面标题中出现的其他实体及其属性。如果当前抽取所得的实体未被包含在知识图谱中，则该实体成为一个新的候选实体。类似地，如果当前被抽取的属性未出现在知识图谱中，则此属性成为一个新的候选属性。这里，我们仅保留置信度高的实体及其属性，新增的实体和属性将被作为新的种子发现新的模式。此过程不断迭代直到没有新的种子可以加入或所有的模式都已经找到且无法泛化。在决定模式的好坏时，常用的基本原则是尽量多地发现属于当前类别的实体和对应属性，尽量少地抽取出属于其他类别的实体及属性。上述方法被称为基于Bootstrapping的多类别协同模式学习。

3. 从抽取图谱到知识图谱

上述所介绍的方法仅仅是从各种类型的数据源抽取构建知识图谱所需的各种候选实体(概念)及其属性关联，形成了一个个孤立的抽取图谱(Extraction Graphs)。为了形成一个真正的知识图谱，我们需要将这些信息孤岛集成在一起。下面我对知识图谱挖掘所涉及的重要技术点逐一进行介绍。

a) 实体对齐

实体对齐(Object Alignment)旨在发现具有不同ID但却代表真实世界中同一对象的那些实体，并将这些实体归并为一个具有全局唯一标识的实体对象添加到知识图谱中。虽然实体对齐在数据库领域被广泛研究，但面对如此多异构数据源上的Web规模的实体对齐，这还是第一次尝试。各大搜索引擎公司普遍采用的方法是聚类。聚类的关键在于定义合适的相似度度量。这些相似度度量遵循如下观察：具有相同描述的实体可能代表同一实体(字符相似);具有相同属性-值的实体可能代表相同对象(属性相似);具有相同邻居的实体可能指向同一个对象(结构相似)。在此基础上，为了解决大规模实体对齐存在的效率问题，各种基于数据划分或分割的算法被提出将实体分成一个个子集，在这些子集上使用基于更复杂的相似度计算的聚类并行地发现潜在相同的对象。另外，利用来自如LOD中已有的对齐标注数据(使用owl:sameAs关联两个实体)作为训练数据，然后结合相似度计算使用如标签传递(Label Propagation)等基于图的半监督学习算法发现更多相同的实体对。无论何种自动化方法都无法保证100%的准确率，所以这些方法的产出结果将作为候选供人工进一步审核和过滤。

b) 知识图谱schema构建

在之前的技术点介绍中，大部分篇幅均在介绍知识图谱中数据层(Data Level)的构建，而没有过多涉及模式层(Schema Level)。事实上，模式是对知识的提炼，而且遵循预先给定的schema有助于知识的标准化，更利于查询等后续处理。为知识图谱构建schema相当于为其建立本体(Ontology)。最基本的本体包括概念、概念层次、属性、属性值类型、关系、关系定义域(Domain)概念集以及关系值域(Range)概念集。在此基础上，我们可以额外添加规则(Rules)或公理(Axioms)来表示模式层更复杂的约束关系。面对如此庞大且领域无关的知识库，即使是构建最基本的本体，也是非常有挑战的。Google等公司普遍采用的方法是自顶向下(Top-Down)和自底向上(Bottom-Up)相结合的方式。这里，自顶向下的方式是指通过本体编辑器(Ontology Editor)预先构建本体。当然这里的本体构建不是从无到有的过程，而是依赖于从百科类和结构化数据得到的高质量知识中所提取的模式信息。更值得一提的是，Google知识图谱的Schema是在其收购的Freebase的schema基础上修改而得。Freebase的模式定义了Domain(领域)，Type(类别)和Topic(主题，即实体)。每个Domain有若干Types，每个Type包含多个Topics且和多个Properties关联，这些Properties规定了属于当前Type的那些Topics需要包含的属性和关系。定义好的模式可被用于抽取属于某个Type或满足某个Property的新实体(或实体对)。另一方面，自底向上的方式则通过上面介绍的各种抽取技术，特别是通过搜索日志和Web Table抽取发现的类别、属性和关系，并将这些置信度高的模式合并到知识图谱中。合并过程将使用类似实体对齐的对齐算法。对于未能匹配原有知识图谱中模式的类别、属性和关系作为新的模式加入知识图谱供人工过滤。自顶向下的方法有利于抽取新的实例，保证抽取质量，而自底向上的方法则能发现新的模式。两者是互补的。

c) 不一致性的解决

当融合来自不同数据源的信息构成知识图谱时，有一些实体会同时属于两个互斥的类别(如男女)或某个实体所对应的一个Property[11] (如性别)对应多个值。这样就会出现不一致性。这些互斥的类别对以及Functional Properties可以看作是模式层的知识，通常规模不是很大，可以通过手工指定规则来定义。而由于不一致性的检测要面对大规模的实体及相关事实，纯手工的方法将不再可行。一个简单有效的方法充分考虑数据源的可靠性以及不同信息在各个数据源中出现的频度等因素来决定最终选用哪个类别或哪个属性值。也就是说，我们优先采用那些可靠性高的数据源(如百科类或结构化数据)抽取得到的事实。另外，如果一个实体在多个数据源中都被识别为某个类别的实例，或实体某个functional property在多个数据源中都对应相同的值，那么我们倾向于最终选择该类别和该值。注：在统计某个类别在数据源中出现的频率前需要完成类别对齐计算。类似地，对于数值型的属性值我们还需要额外统一它们所使用的单位。

4. 知识图谱上的挖掘

通过各种信息抽取和数据集成技术已经可以构建Web规模的知识图谱。为了进一步增加图谱的知识覆盖率，需要进一步在知识图谱上进行挖掘。下面将介绍几项重要的基于知识图谱的挖掘技术。

a) 推理

推理(Reasoning或Inference)被广泛用于发现隐含知识。推理功能一般通过可扩展的规则引擎来完成。知识图谱上的规则一般涉及两大类。一类是针对属性的，即通过数值计算来获取其属性值。例如：知识图谱中包含某人的出生年月，我们可以通过当前日期减去其出生年月获取其年龄。这类规则对于那些属性值随时间或其他因素发生改变的情况特别有用。另一类是针对关系的，即通过(链式)规则发现实体间的隐含关系。例如，我们可以定义规定：岳父是妻子的父亲。利用这条规则，当已知姚明的妻子(叶莉)和叶莉的父亲(叶发)时，可以推出姚明的岳父是叶发。

b) 实体重要性排序

搜索引擎识别用户查询中提到的实体，并通过知识卡片展现该实体的结构化摘要。当查询涉及多个实体时，搜索引擎将选择与查询更相关且更重要的实体来展示。实体的相关性度量需在查询时在线计算，而实体重要性与查询无关可离线计算。搜索引擎公司将PageRank算法[12] 应用在知识图谱上来计算实体的重要性。和传统的Web Graph相比，知识图谱中的节点从单一的网页变成了各种类型的实体，而图中的边也由连接网页的超链接(Hyperlink)变成丰富的各种语义关系。由于不同的实体和语义关系的流行程度以及抽取的置信度均不同，而这些因素将影响实体重要性的最终计算结果，因此，各大搜索引擎公司嵌入这些因素来刻画实体和语义关系的初始重要性，从而使用带偏的PageRank算法(Biased PageRank)。

c) 相关实体挖掘

在相同查询中共现的实体，或在同一个查询会话(Session)中被提到的其他实体称为相关实体。一个常用的做法是将这些查询或会话看作是虚拟文档，将其中出现的实体看作是文档中的词条，使用主题模型(如LDA)发现虚拟文档集中的主题分布。其中每个主题包含1个或多个实体，这些在同一个主题中的实体互为相关实体。当用户输入查询时，搜索引擎分析查询的主题分布并选出最相关的主题。同时，搜索引擎将给出该主题中与知识卡片所展现的实体最相关的那些实体作为“其他人还搜了”的推荐结果。

5. 知识图谱的更新和维护

a) Type和Collection的关系

知识图谱的schema为了保证其质量，由专业团队审核和维护。以Google知识图谱为例，目前定义的Type数在103-104的数量级。为了提高知识图谱的覆盖率，搜索引擎公司还通过自动化算法从各种数据源抽取新的类型信息(也包含关联的Property信息)，这些类型信息通过一个称为Collection的数据结构保存。它们不是马上被加入到知识图谱schema中。有些今天生成后第二天就被删除了，有些则能长期的保留在Collection中，如果Collection中的某一种类型能够长期的保留，发展到一定程度后，由专业的人员进行决策和命名并最终成为一种新的Type。

b) 结构化站点包装器的维护

站点的更新常常会导致原有模式失效。搜索引擎会定期检查站点是否存在更新。当检测到现有页面(原先已爬取)发生了变化，搜索引擎会检查这些页面的变化量，同时使用最新的站点包装器进行AVP抽取。如果变化量超过事先设定的阈值且抽取结果与原先标注的答案差别较大，则表明现有的站点包装器失效了。在这种情况下，需要对最新的页面进行重新标注并学习新的模式，从而构建更新的包装器。

c) 知识图谱的更新频率

加入到知识图谱中的数据不是一成不变的。Type对应的实例往往是动态变化的。例如，美国总统，随着时间的推移，可能对应不同的人。由于数据层的规模和更新频度都远超schema层，搜索引擎公司利用其强大的计算保证图谱每天的更新都能在3个小时内完成，而实时的热点也能保证在事件发生6个小时内在搜索结果中反映出来。

d) 众包(Crowdsourcing)反馈机制

除了搜索引擎公司内部的专业团队对构建的知识图谱进行审核和维护，它们还依赖用户来帮助改善图谱。具体来说，用户可以对搜索结果中展现的知识卡片所列出的实体相关的事实进行纠错。当很多用户都指出某个错误时，搜索引擎将采纳并修正。这种利用群体智慧的协同式知识编辑是对专业团队集中式管理的互补。

知识图谱在搜索中的应用

1. 查询理解

搜索引擎借助知识图谱来识别查询中涉及到的实体(概念)及其属性等，并根据实体的重要性展现相应的知识卡片。搜索引擎并非展现实体的全部属性，而是根据当前输入的查询自动选择最相关的属性及属性值来显示。此外，搜索引擎仅当知识卡片所涉及的知识的正确性很高(通常超过95%，甚至达到99%)时，才会展现。当要展现的实体被选中之后，利用相关实体挖掘来推荐其他用户可能感兴趣的实体供进一步浏览。

2. 问题回答

除了展现与查询相关的知识卡片，知识图谱对于搜索所带来的另一个革新是：直接返回答案，而不仅仅是排序的文档列表。要实现自动问答系统，搜索引擎不仅要理解查询中涉及到的实体及其属性，更需要理解查询所对应的语义信息。搜索引擎通过高效的图搜索，在知识图谱中查找连接这些实体及属性的子图并转换为相应的图查询(如SPARQL[13] )。这些翻译过的图查询被进一步提交给图数据库进行回答返回相应的答案。

总结

这篇文章比较系统地介绍了知识图谱的表示、构建、挖掘以及在搜索中的应用。通过上述介绍，大家可以看出：1)目前知识图谱还处于初期阶段;2)人工干预很重要;3)结构化数据在知识图谱的构建中起到决定性作用;4)各大搜索引擎公司为了保证知识图谱的质量多半采用成熟的算法;5)知识卡片的给出相对比较谨慎;6)更复杂的自然语言查询将崭露头角(如Google的蜂鸟算法)。

此外，知识图谱的构建是多学科的结合，需要知识库、自然语言理解，机器学习和数据挖掘等多方面知识的融合。有很多开放性问题需要学术界和业界一起解决。我们有理由相信学术界在上述方面的突破将会极大地促进知识图谱的发展。