知识图谱研究综述

胡光怡 冯超群 姜欣雨 郭学良

**摘 要** 知识图谱，是结构化的语义知识库，描述物理世界中的概念，物体的属性和物体与物体之间的关联，从大量的数据中抽取出知识，将数据粒度从文件级别降到数据级别，来实现知识的快速响应和推理。知识图谱是对将零散的知识组合起来，以将知识更好地应用。本文首先介绍了知识图谱的发展现状，然后从嵌入算法的角度讲解了知识图谱的嵌入构成，即知识图谱是如何变成向量表示的。接着本文讲解了知识图谱在深度学习中的应用方法，最后讲解了知识图谱的具体应用场景，以及对未来的应用进行展望。

关键词 知识图谱；嵌入；深度学习；推理；语义

**Abstract** Knowledge graph is a structured semantic knowledge base, which describes the concepts in the physical world, the attributes of objects and the relationship between objects, extracts knowledge from a large number of data, and reduces the data granularity from file level to data level, so as to realize the rapid response and reasoning of knowledge. Knowledge graph is the combination of scattered knowledge to make better use of knowledge. Firstly, this paper introduces the development status of knowledge graph, and then explains the embedding structure of knowledge graph from the perspective of embedding algorithm, that is, how knowledge graph become vector representation. Then this paper explains the application method of knowledge graph in deep learning, and finally explains the specific application scenarios of knowledge graph, and prospects for future application.

**Key words** Knowledge graph;embedding;Deep learning; reasoning;semantic

**一、引言**

2012年5月17日谷歌正式提出了“知识图谱” 这个术语。知识图谱旨在描述真实世界中存在的 各种实体或概念，以及他们之间的关联关系。其中，每个实体或概念用一个全局唯一确定的ID来标识，每个属性-值对用来刻画实体的内在属性，而关系用来连接两个实体，刻画它们之间的关联。知识图谱可以融合多种数据源丰富数据语义信息，并且可以结合推理得到的隐含信息为用户提供服务。随着信息检索、智慧城市等应用领域的发展求，将知识图谱应用于这些领域以提高用户体验和系统性能，成为学术界和工业界关注的热点。

移联动互网、人工智能等技术的迅速发展为人们的工作生活带来了很多便利，但是同时也带来了信息过载问题。搜索引擎和推荐系统是解决信息过载的代表技术。传统的搜索引擎在本质上来讲是 帮助用户过滤和筛选信息，这种方式满足了大多数人的需求，但没有提供个性化的服务。相对于传统 搜索引擎来说，推荐系统可以兼顾个性化需求和解决信息过载问题。知识图谱的出现为大数据环境下的推荐系统和搜索引擎的设计提供了一种有效途径。随着互联网上信息的爆发式增长，对于多源异构数据的挖掘是推荐系统中的关键问题。知识图谱的提出为推荐系统中多源异构数据的整合处理带来了新的契机，而基于知识图谱的推荐系统也作为一个新的研究领域被人们所关注。知识图谱的提出最初是为了优化搜索引擎，将数据组织成相互链接并且对于其中的每个实体都富含属性标签，实体间存在多种关系，以此来实现搜索引擎的优化。在对其进行深入研究后，了解到知识图谱的应用不局限于搜索，在诸如智能问答、推荐系统以及金融诈骗领域都有涉猎。

虽然知识图谱的概念较新，但它并非是一个全 新的研究领域．早在2006年，Berners就提出了 数据链接(1inked data)的思想，呼吁推广和完善相关的技术标准URL(uniform resource identifier)，RDF(resource description framework)，OWL(Web ontology language)，为迎接语义网络时代的到来做 好准备．随后掀起了一场语义网络研究热潮，知识图谱技术正是建立在相关的研究成果之上的，是对现有语义网络技术的一次扬弃和升华。

**二、知识图谱的构建过程**

在初期，构建知识图谱的方式主要是自顶向下的方式，如Freebase，将概念和关系预先定义出来，再填充知识，而现在更多的是采用自底向上的方式，从文本中抽取出抽象的概念、实体、关系以及实体的属性，如Google的Knowledge Vault和微软的Satori知识库。

知识图谱的原始数据类型一般来说有三类：结构化数据，如关系数据库；非结构化数据，如图片、音频、视频；半结构化数据 如XML、JSON、百科。存储这三类数据类型的方式一般有两种：一个是使用RDF（资源描述框架），如RDF/XML、N-Triples、RDFa；一种方法是图数据库，如FlockDB、GraphDB、Neo4j。当然也可以用关系数据库来存，但是当知识图谱变得复杂的时候，图数据库的查询效率会比关系数据库高很多，涉及到2，3度的关联查询时，图数据库的查询效率会高出几千倍甚至几百万倍。而且，基于图的存储的改动也只需要局部的改动。

在逻辑上，通常将知识图谱划分为两个层次：数据层和模式层。模式层存储经过提炼的知识，通常通过本体库来管理这一层，本体库有点类似于面向对象里“类”的概念，数据层存储真实的数据。例如，模式层：实体-关系-实体，实体-属性-性值；数据层：比尔盖茨-妻子-梅琳达·盖茨，比尔盖茨-总裁-微软。

从技术上，构建知识图谱是一个迭代更新的过程，根据知识获取的逻辑，每一轮迭代包含三个阶段：信息抽取：从各种类型的数据源中提取出实体、属性以及实体间的相互关系，在此基础上形成本体化的知识表达；知识融合：在获得新知识之后，需要对其进行整合，以消除矛盾和歧义，比如某些实体可能有多种表达，某个特定称谓也许对应于多个不同的实体等；知识加工：对于经过融合的新知识，需要经过质量评估之后，其中部分需要人工参与甄别，才能将合格的部分加入到知识库中，以确保知识库的质量。

信息抽取包括实体抽取、关系抽取和属性抽取。实体抽取，也称为命名实体识别，是指从文本中自动识别出实体。目前从面向单一领域进行实体抽取，逐步跨步到面向开放域（open domain）的实体抽取。关系抽取是从相关语料中提取出实体之间的关联关系，通过关系将实体联系起来，这样才能够形成网状的知识结构。属性抽取的目标是从不同信息源中采集特定实体的属性信息，如针对某个公众人物，可以从网络公开信息中得到其昵称、生日、国籍、教育背景等信息。

知识融合包括实体链接和知识合并。实体链接（entity linking）：指对于从文本中抽取得到的实体对象，将其链接到知识库中对应的正确实体对象的操作。从文本中通过实体抽取得到实体指称项，进行实体消歧、共指消解等相似度计算，在确认知识库中对应的正确实体对象之后，将该实体指称项链接到知识库中对应实体。知识合并：实体链接是处理从半结构化数据和非结构化数据里提取出来的数据。结构化数据的处理，如外部知识库和关系数据库，是知识合并的内容。主要分两种：合并外部知识库，主要处理数据层和模式层的冲突；合并关系数据库，有RDB2RDF等方法。

知识加工包括本体构建、知识推理和质量评估。本体构建： 本体可以采用人工编辑的方式手动构建，也可以以数据驱动的自动化方式构建本体。自动化本体构建过程包含三个阶段：实体并列关系相似度计算，实体上下位关系抽取，本体的生成。知识推理：完成本体构建后，群在知识图谱之间大多数关系残缺的问题，例如，如果A是B的配偶，B是C的主席，C坐落于D，那么我们就可以认为，A生活在D这个城市。知识推理的对象可以是实体间的关系，也可以是实体的属性值，本体的概念层次关系等。涉及的算法主要可以分为3大类，基于逻辑的推理、基于图的推理和基于深度学习的推理。质量评估：对知识的可信度进行量化，通过舍弃置信度较低的知识来保障知识库的质量。

构建完成后当然还有知识更新，从逻辑上看，概念层的更新是新增数据后获得了新的概念，需要自动将新的概念添加到知识库的概念层中。数据层的更新是新增或更新实体、关系、属性值，对数据层进行更新需要考虑数据源的可靠性、数据的一致性（是否存在矛盾或冗杂等问题）等，并选择在各数据源中出现频率高的事实和属性加入知识库。从方法上看，全面更新指以更新后的全部数据为输入，从零开始构建知识图谱。这种方法比较简单，但资源消耗大，而且需要耗费大量人力资源进行系统维护；增量更新指以当前新增数据为输入，向现有知识图谱中添加新增知识。这种方式资源消耗小，但目前仍需要大量人工干预（定义规则等），因此实施起来十分困难。

# 三、知识图谱嵌入表示

## 3.1.目的

知识图谱嵌入表示的目的是将实体和关系映射到低维连续的向量空间，从而简化知识图谱的相关计算。

## 3.2.技术分类

知识图谱现有的嵌入技术分类如图2-1所示。

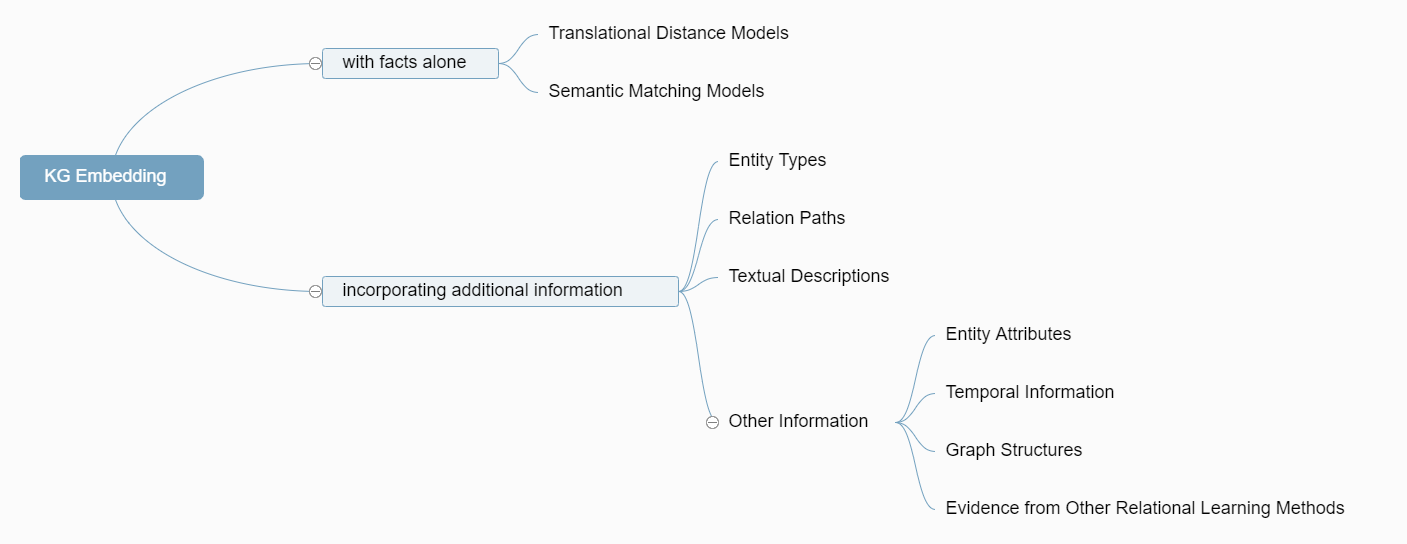


图3-1 知识图谱嵌入技术分类

其中Translational Distance Models的模型分类如图2-2所示。

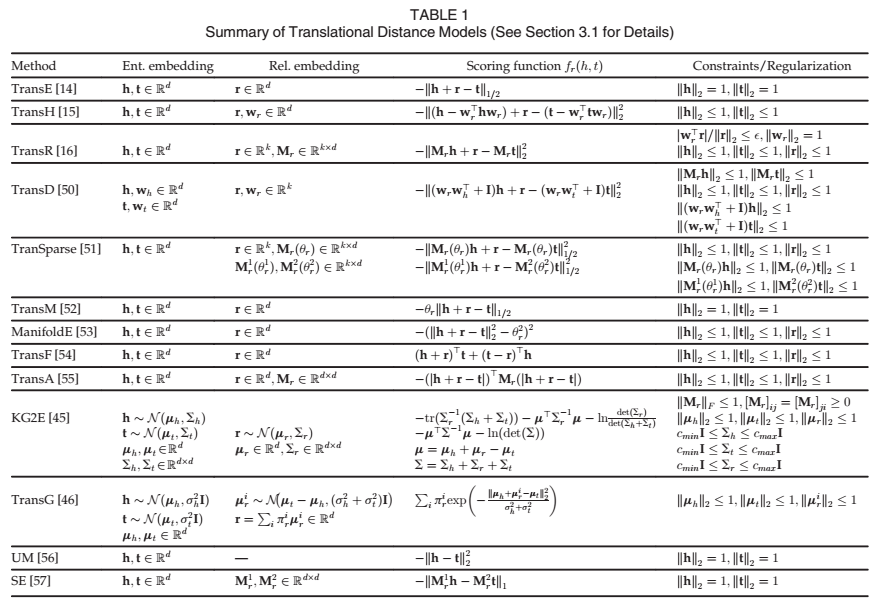
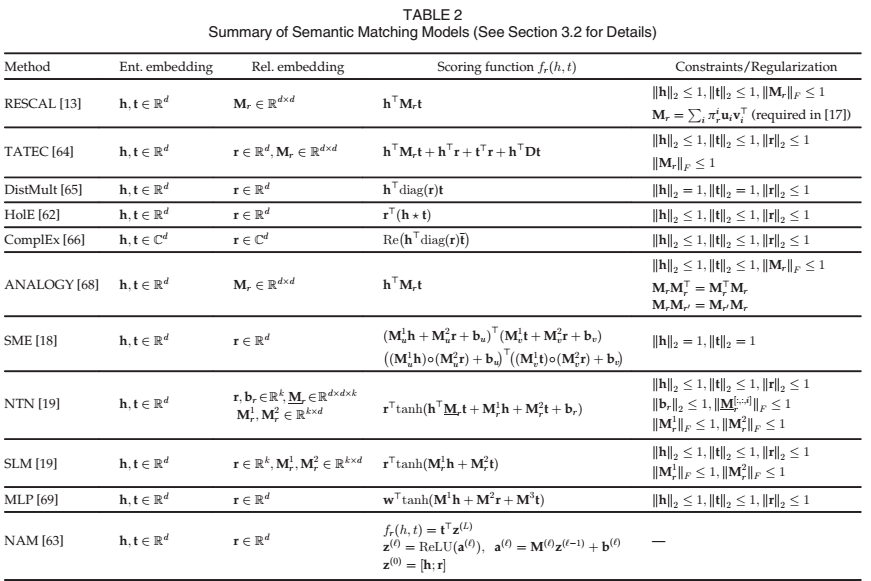


图3-2 平移距离模型分类图

Semantic Matching Models的模型分类如图2-3所示。

图3-3 语义匹配模型分类图

## 3.3.知识图谱嵌入技术的一般步骤

（1）表示图中的实体和关系

（2）定义一个打分函数（scoring function）

（3）学习实体和关系的向量表示

## 3.4.Translational Distance Models

（1）TransE

给定一个fact(h,r,t)，TransE模型将关系表示为平移向量，这样就能以较低的错误把实体的向量连接起来，即：。打分函数定义为与之间的距离，即，如果(h,r,t)存在，那么分数就比较高。TransE模型示意图如图4-1所示。

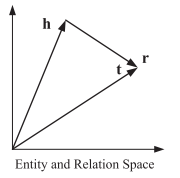


图3-3 TransE模型示意图

虽然TransE模型简单有效，但是它并不能处理1-N，N-1, N-N的问题。比如，一个导演指导了多部电影，根据头节点h（导演），关系r（指导），尾节点t（电影）进行模型训练，那么这些电影向量的距离是很近的，而事实上他们是完全不同的实体。为了克服这个问题，有效的策略是允许实体在涉及不同关系时具有不同的表示。

（2）TransH

TransH通过引入关系特定（relation-specific）的超平面，允许实体在涉及不同关系时具有不同的表示。如图4-2所示，TransH也将实体建模为向量但是每个关系r都作为超平面上的向量，其中作为法向量。给定一个fact(h,r,t)，实体表示和首先被投影到超平面上，即 ， ，如果(h,r,t)存在，这两个投影会由以较低的错误连接，也就是，得分函数。通过引入投射到关系特定超平面的机制，TransH实现了不同关系中实体由不同的表示。

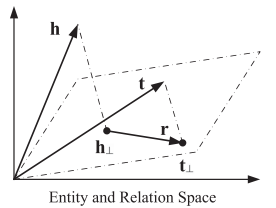


图3-4 TransH模型示意图

（3）TransR

TransR与TransH类似，但它引入了关系特定（relation-specific）的空间，而不是超平面。在TransR中，实体表示为实体空间中的向量，并且每个关系r与特定空间相关联，并在该空间中建模为平移向量。给定一个fact(h,r,t)，TransR首先将实体表示和投影到关系r特定的空间中，即，，其中是从实体空间到r的关系空间的投影矩阵。得分函数定义为。图4-3是TransR的示意图。尽管TransR在建模复杂关系方面很强大，但是它为每个关系引入了一个投影矩阵，相比TransE、TransH模型而言，没有那么简洁高效。

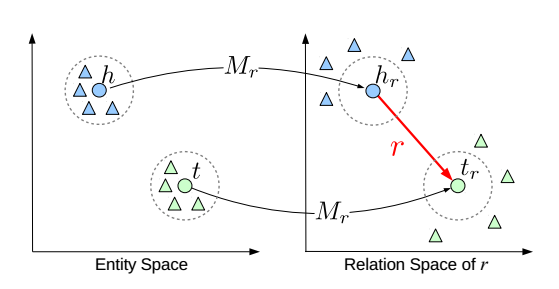


图3-5 TransR模型示意图

（4）TransD

TransD通过进一步将投影矩阵分解为两个向量的乘积来简化TransR。对于每个fact(h,r,t)，TransD除了有实体/关系的表示和外，还引入了额外的映射向量，**，，，**两个投影矩阵和相应地被定义为，然后将这两个投影矩阵分别应用于头部实体和尾部实体以获得它们的投影，即 ， ，打分函数与TransR一样：，而且比TransR更有效。

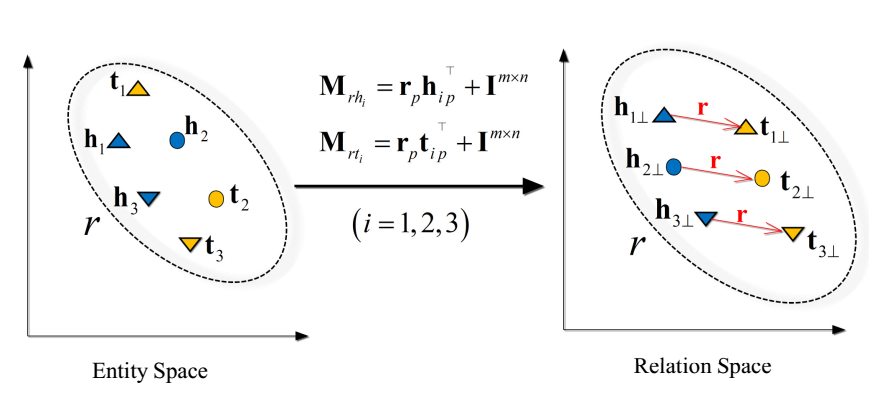


图3-6 TransD模型示意图

## 3.5. Semantic Matching Models

SME（semantic matching energy）模型，使用神经网络结构进行语义匹配，如图5-1所示。给定一个fact(h,r,t)，它首先在输入层将实体和关系投影为它们的向量，然后在隐藏层，与结合得到，与结合得到。得分函数定义为和的点积，即。其中，有两个版本：线性版本和双线性版本，线性版本为，；双线性版本为，。其中、、为权重矩阵，为偏移向量，它们在不同的关系之间共享。

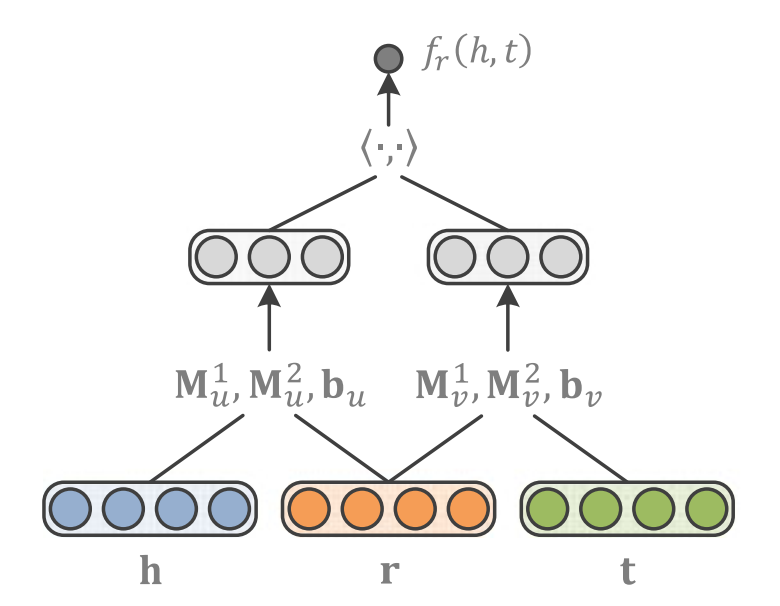


图3-7 SME模型示意图

# 四、知识图谱在深度学习中的使用方法

## 4.1 关系抽取中的知识图谱

关系抽取任务相信很多人都听说过，作为一种生成知识图谱的新方法。但是由于其精度不高，导致其目前仍然处于尝试阶段，无法实际应用。

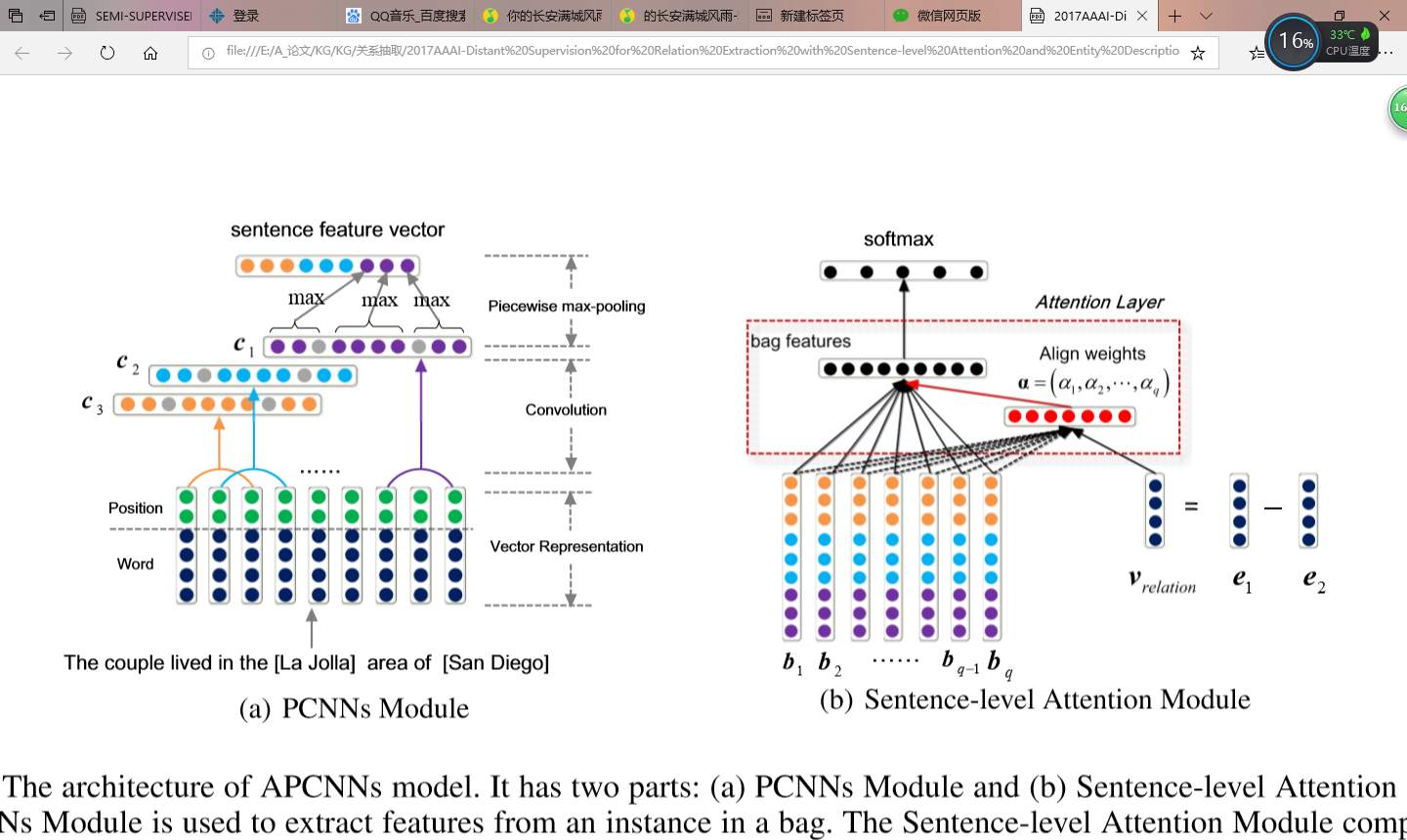


图4-1

实际上这篇文章并没有使用知识图谱，我将其放在这里的原因在于，本文使用了知识图谱的一个非常重要的知识点：知识图谱嵌入方法。本文使用了Trans E思想。

下面具体讲述这篇文章是怎么在关系抽取任务中应用Trans E的。

关系抽取最著名的一个方法就是远程监督方法。提出这个方法是因为有标注数据的稀疏性。仅仅有少量句子具备标注，大多数句子没有标注内容。如果仅仅使用有标注数据进行训练，效果一定会差强人意。因此有人提出了远程监督方法：假设拥有同样实体对的句子共享同一标注。这样将会大大增加标注句子的数量。

但是这样的方法也会有一定的问题，拥有同一实体对的句子中，一定包含不表示这样关系的实体。处理这样问题的方法，最普遍的是应用attention机制。通过机器学习的方式，学习到哪些句子是真正包含这样的关系，哪些句子不包含这样的关系。包含这样关系的句子attention将会比较高。

上面提出的这篇文章同样使用了 远程监督方法，同样也应用了attention机制来解决噪声的问题。

这篇文章最大的亮点在于使用了Trans E的方法来计算句子集合中每个句子attention的值。本文提出了一个假设，即假设如果某个句子包含给定的这个关系，那么这个句子的向量表示将会和这个关系的向量表示相似。而这个关系向量就是使用知识图谱嵌入方法Trans E得到的。

最开始看这篇文章的时候我对这个Trans E的引入持有怀疑的态度。因为我们要求的就是句子集合的向量表示，那么只要知道每个句子的attention就好了。每个句子的attention将会在训练中学习到，为何要引入Trans E来计算每个句子的attention呢？

后来随着对文章了解的不断深入，我了解到这篇文章实际上做了两件事。第一个是学习到每个句子的attention，第二个是通过引入关系向量表示，对每个单词、每个实体向量进行微调，最终可以得到最合适的attention以及最合适的词汇向量表示。

## 4.2 推荐中的知识图谱

推荐是知识图谱的重要应用领域。这一篇文章就是将知识图谱应用在推荐上。下面介绍推荐的具体步骤：

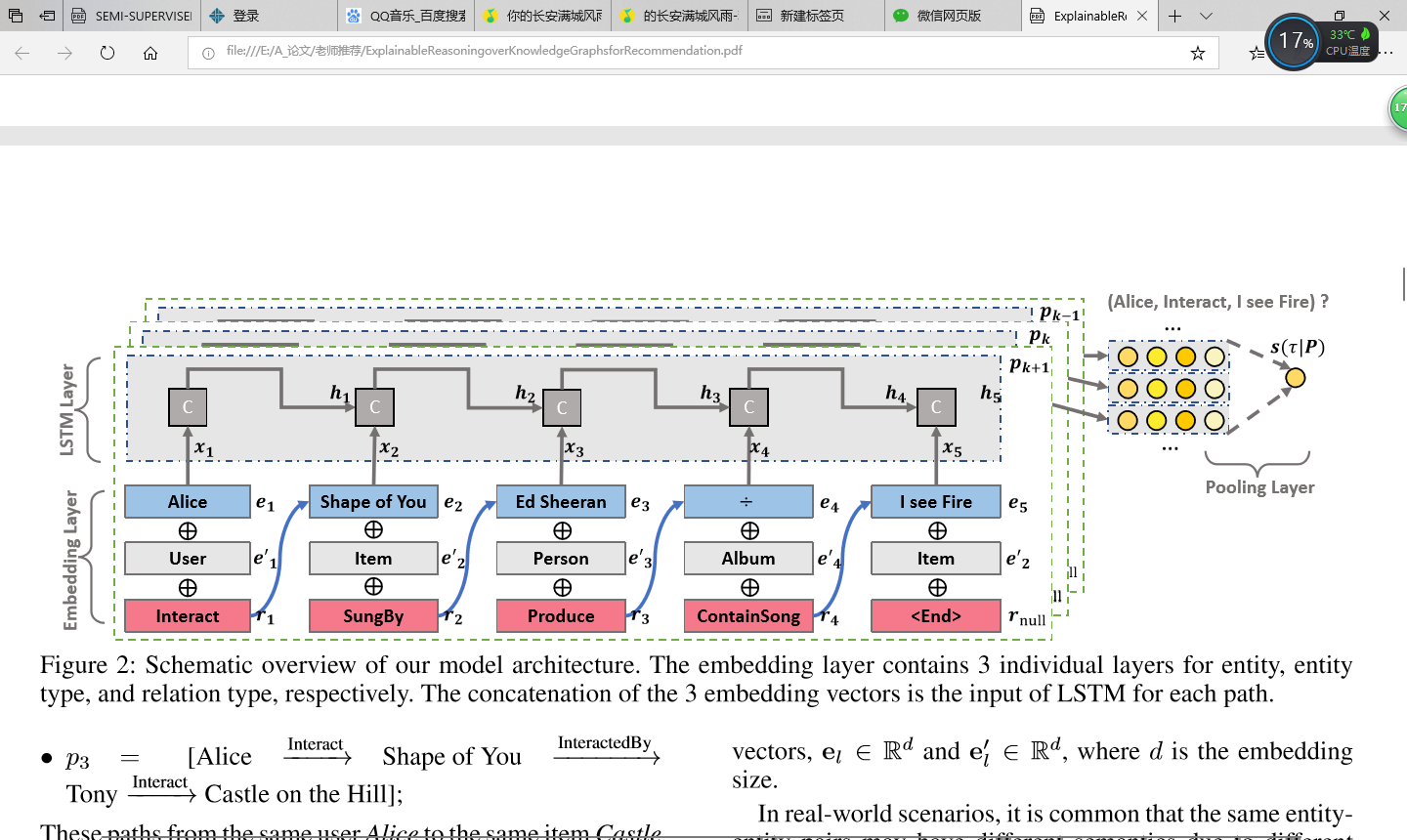


图4-2

1、首先有一个知识图谱，所有的歌曲和歌曲的专辑、歌手、作曲者等，构成一个庞大的知识图谱。当某个用户A收听了某一首歌曲的时候，就将A加入到知识图谱中。下一步就是为其推荐他可能喜欢的歌曲。

2、找到一些候选推荐的歌曲，对每一首歌，找到从用户A到这首歌B的所有可能路径。由于路径过多，因此只考虑路径长度小于6的路径。

3、对于每一条路径，提取出路径中的实体，对于每个实体，随机初始化他的向量表示a，同样随机初始化他的类型向量表示b和路径中后一个实体与他关系的向量表示c。将a、b、c简单拼接在一起，输入到LSTM中，最终预测用户喜欢这首歌曲的可能性，最终可能性区间为（0，1），趋近于1则用户会喜欢这首歌。

需要注意的是，这篇文章提出实际上做了两件事。第一件为推荐，即预测用户是否会喜欢某件商品。第二件为知识图谱嵌入，因为本文使用的所有向量都没有与训练，说明他们全部都是随机初始化的，因此这篇文章做的第二件事是代替Trans E等方法的知识图谱嵌入算法。

## 4.3 QA中的知识图谱

QA需要匹配问题向量和答案向量，问题向量很简单，就使用word2vec或者卷积的方法就可以得到，关键在于答案的向量表示。

本文使用三种方法得到答案向量表示，从三个角度解析知识图谱内容。

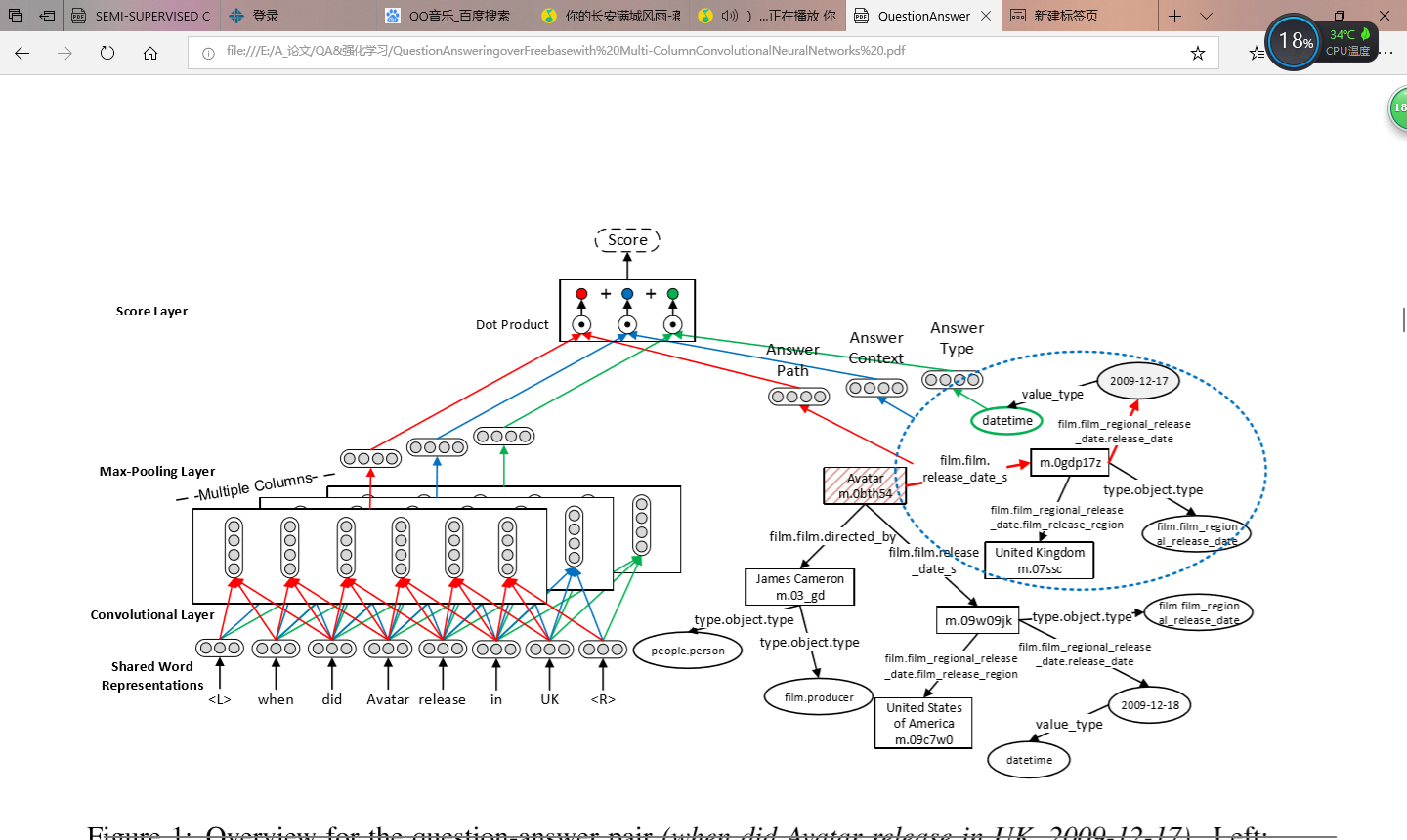


图4-3

1.首先定位问题中所有的实体，在知识图谱中定位到这些实体。

2.将定位到的实体一跳或者两跳的实体都作为候选答案。

3.对每个候选答案，用三个角度表示这个候选答案。第一个是候选答案的类型，比如人、物品等等。第二个是候选答案的邻居，将其邻居实体向量表示加和取平均值。第三个是路径依赖，即问题实体和这个候选答案的路径，将路径上的实体关系向量加和取平均值。

4.三种候选答案表示向量分别与答案做相似度匹配，最终得到这个候选答案是真正答案的可能性。

个人认为三种表示方法中，第二种貌似用处不大，因为所有的候选答案的邻居实体都差不多，他们都聚在一堆，所以这个无法让候选答案产生实质性的区别，因此个人觉得用处不大。第三种表示方法应该是用处最大的表示方法，因为比如问题是“美国总统是谁？”问题中实体为“美国”，关系为“总统”，知识图谱中“特朗普”和“美国”的连线一定有一个是“总统”，因此这样的路径依赖方法得到的向量一定和问题向量相似度最高，毕竟单词都一样“美国”“总统”。

这样的QA应用一般是用在信息检索方面，因为这是基于知识的问答，而不是和机器人亲切友好的闲聊。

# 五、知识图谱的应用

Google为了提升搜索引擎返回的答案质量和用户查询的效率，于2012年5月16日发布了知识图谱(Knowledge Graph)。有知识图谱作为辅助，搜索引擎能够洞察用户查询背后的语义信息，返回更为精准、结构化的信息，更大可能地满足用户的查询需求。Google知识图谱的宣传语“things not strings”给出了知识图谱的精髓，即，不要无意义的字符串，而是获取字符串背后隐含的对象或事物。因此知识图谱作为辅助计算机获取语义层次知识的辅助技术被广泛应用。

5.1 知识图谱在搜索引擎上的应用

搜索引擎（Search Engine）是指根据一定的策略、运用特定的计算机程序从互联网上搜集信息，在对信息进行组织和处理后，为用户提供检索服务，将用户检索相关的信息展示给用户的系统。搜索引擎包括全文索引、目录索引、元搜索引擎、垂直搜索引擎、集合式搜索引擎、门户搜索引擎与免费链接列表等。

搜索系统一般有两种场景。第一种场景是，用户知道某个事物的名字，但是只知道这个事物的名字，用户想知道有关这个事物的其他相关信息，即了解这个东西。那么搜索引擎需要将这个事物，尽可能多的信息返还给用户。第二种场景是，用户不知道某个事物的名字，但是能够通过自己的语言简单描述这个事物。搜索引擎需要将这个事物的名字，以及其他相关信息返还给用户。

目前为止传统的搜索引擎，大部分只是单纯的对字符串进行处理，对语义的理解能力不是很强。下图为搜索引擎的三个处理层次的示意图。如图所示，搜索引擎一般分为语言层、语义层、执行层三个层次。语言层：主要用于剔除语言中的噪音，杂质，以及时间、数值识别、描述一致性等问题。处理成计算机能识别的语言。语义层：主要处理自然语言中的语义问题，这一块决定了整个自然语言处理流的描述能力。知识图谱主要解决的也是这一层次的问题。执行层则是根据语义层的处理结果判断信息的相关程度，对候选信息进行排序并输出优先的几个结果。

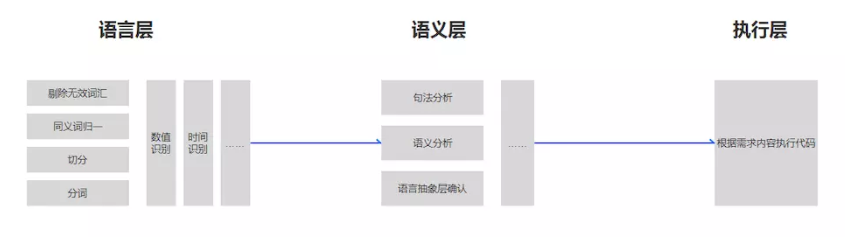


图5-1 搜索引擎的三层结构

以百度为例，当我们输入相关内容后，搜索引擎只是单纯的对输入的字符串进行简单的语义分析。即从海量URL中找到标题和正文经过NLP语言层处理后匹配度最高的一个URL group。可以这么理解，语言层将语言处理成了标准形式，然后使用一个排序算法，得到各种URL的得分，根据得分高低将URL展示给用户。这一过程中，语义层起到的作用甚微，可能就仅仅只在于将语言归一化上。

在这种情况下，搜索引擎是不知道我们实际上输入了什么了的，他只是对字符串进行了处理，并没有联系其他的语义知识，用户到底在讲什么，搜索引擎不知道也不关心。而如果借助知识图谱的帮助，搜索引擎就会通过知识图谱将表层的理解变为深层次的翻译。机器会根据用户的输入联想到其他相关知识，而不只是字符串上单纯的联系。这就是应用知识图谱与不应用知识图谱在搜索系统上的本质区别。

下图即为应用了知识图谱的google搜索对马云关键字的搜索结果，可以看出除了含有马云关键词的搜索结果，还包括了马云的各种信息、家人、朋友、相关企业家。



图5-2 google对“马云”的搜索结果

5.2 知识图谱在推荐系统上的应用

推荐系统的应用场景与搜索系统不同。一般是，用户不会告诉机器想要什么，甚至用户自己都不知道想要什么，这时候就需要机器对用户的历史行为进行分析，然后推测出用户想要的事物。

传统的推荐系统一般是基于内容或者基于协同过滤等相关算法来实现的，但是基于这些算法的推荐系统往往存在着冷启动问题以及不能充分理解user和user之间、item和item之间、以及user和item之间的关系的问题。知识图谱可以通过对知识库中实体与实体之间的联系，更好的分析上述关系以及解决因为数据量不够的冷启动问题。

知识图谱的推荐主要是通过实体与实体之间的关系，将用户搜索实体的相关内容根据一定的逻辑推荐给用户，以北京大学为例，搜索北京大学后，神马搜索会出现“知名校友”，“相关院校”的推荐栏目，通过知识图谱，能够准确的知道哪些人从北京大学毕业的，然后通过一系列的热点排序算法，将用户最关心的毕业生选出来，知名校友这一栏，就可以得出以下结果。相关院校这一块，可以采用对比学校的相似度，相似度高的排前面，例如：清华、复旦、浙大、北大都是985院校，其他属性相似度越高。



图5-3 神马搜索对“北京大学的搜索结果”

将知识图谱作为辅助信息引入到推荐系统中可以有效地解决传统推荐系统存在的稀疏性和冷启动问题。目前，将知识图谱特征学习应用到推荐系统中主要通过三种方式——依次学习、联合学习、以及交替学习。

1、依次学习（one-by-one learning）

首先使用知识图谱特征学习得到实体向量和关系向量，然后将这些低维向量引入推荐系统，学习得到用户向量和物品向量；

依次学习的优势在于知识图谱特征学习模块和推荐系统模块相互独立。在真实场景中，特别是知识图谱很大的情况下，进行一次知识图谱特征学习的时间开销会很大，而一般而言，知识图谱远没有推荐模块更新地快。因此我们可以先通过一次训练得到实体和关系向量，以后每次推荐系统模块需要更新时都可以直接使用这些向量作为输入，而无需重新训练。

依次学习的缺点也正在于此，因为两个模块相互独立，所以无法做到端到端的训练。通常来说，知识图谱特征学习得到的向量会更适合于知识图谱内的任务，比如连接预测、实体分类等，并非完全适合特定的推荐任务。在缺乏推荐模块的监督信号的情况下，学习得到的实体向量是否真的对推荐任务有帮助，还需要通过进一步的实验来推断。

2、联合学习（joint learning）

将知识图谱特征学习和推荐算法的目标函数结合，使用端到端（end-to-end）的方法进行联合学习；联合学习的优劣势正好与依次学习相反。联合学习是一种端到端的训练方式，推荐系统模块的监督信号可以反馈到知识图谱特征学习中，这对于提高最终的性能是有利的。但是需要注意的是，两个模块在最终的目标函数中结合方式以及权重的分配都需要精细的实验才能确定。联合学习潜在的问题是训练开销较大，特别是一些使用到图算法的模型。

3、交替学习（alternate learning）

将知识图谱特征学习和推荐算法视为两个分离但又相关的任务，使用多任务学习（multi-task learning）的框架进行交替学习。交替学习是一种较为创新和前沿的思路，其中如何设计两个相关的任务以及两个任务如何关联起来都是值得研究的方向。从实际运用和时间开销上来说，交替学习是介于依次学习和联合学习中间的：训练好的知识图谱特征学习模块可以在下一次训练的时候继续使用（不像联合学习需要从零开始），但是依然要参与到训练过程中来（不像依次学习中可以直接使用实体向量）。

除了上述介绍的，知识图谱在搜索结果上的推荐以外，知识图谱在推荐系统上的应用还有：旅游推荐、电子商务推荐、电影或者音乐推荐、职位推荐等

5.3 知识图谱在其他方向的应用

除了搜索引擎和推荐系统这两个最典型的应用外，知识图谱还可以应用在QA系统、情感分析、不良信息发现、金融分析等方向。

利用知识图谱的知识相关性的联系，机器可以使用和人的推理联想类似的能力，使得机器对语义的理解能力大大提升。通过对语义的充分理解，机器可以在QA系统、情感分析、不良信息发现等方面做得更好。而借助知识图谱，我们可以构建与企业相关的知识图谱，这样可以将混乱的信息整理清晰，对一些企业的经营状况进行更加准确的分析，从而可以得到更加准确的结果。同样的方法也可以应用于公安部门对于犯罪嫌疑人的调查中。

5.4 本章总结

本章介绍了知识图谱的应用领域，主要介绍了知识图谱在搜索引擎、推荐系统当中的应用。知识图谱的使用主要是通过使用知识图谱增强对知识库的语义分析，增强知识库中实体与实体之间的联系，从表层的字面分析到深层的知识推理，从而得到更加准确的分析结果。简单介绍了知识图谱在QA系统、情感分析、不良信息发现、金融分析和公安部门的应用。

# 六、总结与展望

本文主要对知识图谱的应用过程进行了详细的考察。讲解了知识图谱的构建、知识图谱的嵌入方法、知识图谱在深度学习中的使用以及知识图谱的应用四个方面。知识图谱是一个全局知识库，是支撑搜索引擎、问答系统、推荐系统等智能应用的基础。知识图谱不仅是一项技术，更是一种知识的财富。

由于知识图谱只有在达到一定规模时才能发挥较好的效果，因此在应用知识图谱时还存在许多难点问题有待解决。典型的问题包括多源信息融合、知识图谱中用户特征的挖掘及潜在需求的分析、多维度推荐、与时间空间推理的结合、用户隐私保护等等。此外，近年来兴起的深度学习、表示学习、位置感知等也为基于知识图谱的具体应用提供了许多可能的改进方向，值得进一步探索与发现。

**参考文献：**

[1] XIONG H, LIU Z. A situation information integrated personalized travel package recommendation approach based on TD-LDA model[C]//Behavioral, Economic and Socio-cultural Computing (BESC), International Conference on. IEEE, Nanjing, China,2015:32-37.0

[2] 常亮, 曹玉婷, 孙文平,等. 旅游推荐系统研究综述[J]. 计算机科学, 2017, 44(10):1-6.

[3] 孟祥武, 纪威宇, 张玉杰. 大数据环境下的推荐系统[J]. 北京邮电大学学报, 2015, 38(2):1- 15.

[4] 刘峤, 李杨, 段宏,等.知识图谱构建技术综述[J].计算机研究与发展,2016, 53(3):582-600.

[5] LUBERG A, TAMMET T, JARV P. Smart City: A Rulebased Tourist Recommendation System[M]// Information and Communication Technologies in Tourism 2011. Springer Vienna, 2011:51-62.

[6] 漆桂林, 高桓, 吴天星. 知识图谱研究进展[J]. 情报工程, 2017, 3(1):4-25.