北京大学

博士学位研究生学科综合考试报告

院　　系：　信息科学技术学院
姓　　名：　林　泽　琦
学　　号：　1401111333
专　　业：　计算机软件与理论
研究方向：　软件工程与软件工程环境
导　　师：　张　路　　教授

2017 年 4 月

摘　要

一个软件项目在其生命周期中往往会产生大量相关数据（包括源代码以及各种文本形式的资料等），其中蕴含着丰富的软件知识。本文作者的研究兴趣是对这些知识进行提炼、理解与利用，从而为软件开发人员提供智能辅助，以提高软件维护与复用的效率与质量。围绕这一目标，本文对特征定位、问答系统与知识表示学习这三个方面的技术进行了文献综述，并提出了博士论文的研究设想。

第一章介绍特征定位技术，这是在软件维护与复用过程中为软件开发人员提供智能辅助的一类典型工作。特征定位，指的是：对于开发人员指定的一个功能，自动从包含该功能的程序源代码中找出实现该功能的那部分代码。特征定位有助于软件开发人员进行程序理解，从而提高软件维护与复用的效率与质量。根据特征定位技术所采用的具体技术，本章将其分为静态特征定位技术、动态特征定位技术、文本特征定位技术以及它们的组合特征定位技术，并对各类特征定位技术进行了文献综述。

第二章介绍问答系统，动机是希望通过问答的形式将软件知识释放给软件开发者。问答系统是信息检索的高级形式，能够自动理解用户输入的自然语言问句的语义，并找到该问题的答案。问答系统研究兴起的主要原因是信息规模的不断增长与人们对快速、准确地获取信息的需求之间的矛盾。本章将问答系统的整体结构归纳为三部分：问题处理、段落检索与答案处理，分别进行了文献综述，并以 IBM Watson 问答系统这作为实例进行具体介绍。

第三章介绍知识表示学习技术，动机是希望能借助此类技术对软件知识进行语义分析与推理，以支持问答。知识表示学习是近年来人工智能领域新兴的一个研究热点，指的是通过机器学习的方法，将结构化的知识库映射为潜在语义空间中的表示向量的过程。对于规模庞大、结构复杂的知识库，知识表示学习技术可以有效地表示其中的实体与关联关系的潜在语义，从而提升知识融合、补全、推理等各种任务的性能。本章对目前提出的多种知识表示学习模型进行了文献综述，并以其中最受关注的翻译模型为主，介绍了知识表示学习在面对复杂知识时的处理方法。

综合上述文献综述与作者在读博过程中的研究工作，第四章提出了博士论文的研究设想：面向软件复用的智能问答。

目录

图表目录

第一章　特征定位技术

1.　特征定位技术

1.1.　背景

程序理解是软件维护、演化过程中程序员的重要活动，也是复用软件遗产系统基础。研究程序理解可以追溯到上世纪七十年代中后期。当时，大量的软件系统需要维护，而维护人员往往不是所要维护系统的开发人员；另外，由于设计文档的缺失、或因没有及时更新而导致设计文档与代码不一致，这使得源代码成为唯一可靠资源。维护人员需利用其所具有的编程经验及与问题域有关的知识，通过分析代码，获得所需知识以完成维护任务。

同时，随着计算机系统的广泛应用，软件规模、复杂性不断增加，软件维护的成本也不断增加。Rugaber [Rugaber, 1995] 指出软件维护的费用通常超过开发一个新系统的开销。Boehm[Boehm, 1984]研究表明，在软件生命周期中，系统维护所花费的资源和时间占总费用的 50%到 75%。而维护阶段的开销又集中在对系统的理解，Fjeldstad 和 Hamlen [Fjeldstad and Hamlen, 1983]指出，在完善性维护与纠正性维护任务中，分别花费约 47%与 62%的时间用于理解系统。

此外，许多研究表明，计算机应用系统之间存在大量相同或相近的功能，如 Burroughs 公司 Goodell 考察了 1300 个应用程序，仅标识出 24 个互相不同的基本功能；对 Raytheom 的商业应用程序研究中揭示：他们 60%的程序具有基本类似的功能；美国宇航局（NASA）曾研究其所用的 25 个软件项目，发现近 42%功能重复。因此，在开发新应用软件系统时，若能复用已有软件中的功能，就可能减少测试、维护所复用功能的费用，从而降低软件开发与维护的成本。软件复用思想提出后，受到产业界与学术界的广泛关注，并得到大量的实践，有着深远的影响。当今，存在大量开源项目、开源代码，软件复用对于软件开发有着重要意义；2007 年 11 月，Google 公司推出的智能手机操作系统 Android 系统堪称复用开源项目成功典范。然而，正确有效地复用开源项目或已有系统功能的关键是正确理解所复用的程序。

软件工程是一个关注提高软件生产率和软件质量的研究领域。提高软件的生产率，需要有效的软件开发方法与工具，同时，也需要改善软件维护方法和手段，因而必须关注程序理解活动。对于一个软件，程序包含了系统做什么和如何做的所有

信息。所谓程序理解是从程序出发获取完成任务所需知识的活动。鉴于其在软件维护、软件复用中所起的作用，在过去 30 多年间，学术界出现了大量有关程序理解的研究，提出了许多方法、技术，推动着程序理解研究的发展；与此同时，也出现很多辅助程序理解的工具。

为完成软件维护或复用现存软件系统任务，程序员通常只需理解与其任务相关的那部分代码，并不需要理解整个软件。那么，程序员如何知道哪些代码与其任务相关？在几十万行、几百万行、有的甚至几千万行的企业应用程序、商用软件中，找出与任务相关的几百行、几千行代码并不是一件容易的事！是否存在有效技术帮助其高效地界定与任务相关的代码？如何组织、表示所得到的代码以便程序员理解该代码？**特征定位**旨在提供界定与任务相关代码的有效技术，并合适地组织与表示所得代码以便程序员使用。

Wilde 等[Wilde et al., 1992]于 1992 年最早提出软件侦察（Software Reconnaissance）技术用于特征定位；此后，众多研究人员在软件工程重要会议（如 ICSE、ICSM、ICPC、ASE 等）与期刊（如 TSE、TOSEM、JSS、JSME 等）上，发表大量关于特征定位的研究成果；本章目的在于从中选择若干文献，归纳整理，以便了解该方向技术发展趋势。

1.2. 特征定位

本节将介绍特征定位的基本概念，并分析特征定位与其他相关研究领域的关系。

1.2.1. 特征

如前所述，特征定位旨在提供界定与任务（软件维护、现存系统复用）相关代码的技术，是程序理解的基础，服务于开发人员维护软件、复用现存系统。在纠正性维护与完善性维护任务中，开发人员改正或改进软件中一个或若干个功能，这些功能是从用户视角软件所拥有的功能，它们属于问题空间。所以，在软件维护时，开发人员任务中的功能是属于问题空间中；而软件需求文档用于描述问题空间中的功能，从而，软件维护任务中的功能，通常也是软件需求文档中所描述的软件功能。同样，在复用现存软件系统的功能于新开发的软件过程，所复用的功能也是现存软件系统需求文档所述的功能。因此，在特征定位中，特征是指软件需求文档所述的功能。以下是文献中有关特征的定义：

- Eisenbarth 等[Eisenbarth et al., 2003]定义：一个特征即是已被实现的软件系统的一个功能性需求。
- Koschke 等[Koschke and Quante, 2005]定义：一个特征是用户手册或需求规约文档中所描述的、并已被实现的一个产品功能，是用户想要系统做并且用户能观察到的事。
- Antoniol 等[Antoniol and Guéhéneuc, 2006]定义：一个特征是程序的一个需求，用户能使用它，并产生可观察的行为。
- Poshyvanyk 等[Poshyvanyk et al., 2007]定义：一个特征是由需求文档明确定义的、开发人员与用户都易于理解的、并具有可观察结果的一个软件功能。

综合这几种定义，特征的合理解释应当是：

特征是用户手册或需求规约文档中明确定义的、并已被实现的软件功能，该功能易于开发人员与用户理解、并具有可观察结果。

1.2.2. 特征定位

特征定位在文献中也称概念定位、概念分配（assignment）。 Dit 等[Dit et al., 2011]认为：特征定位是在源代码中找出实现系统给定功能代码初始位置的活动。该定义隐含前提条件是：找到实现系统给定功能代码初始位置后，借助现有工具或技术，开发人员能方便地找到实现所指定功能的其余代码。但实际情形并非如此。例如，在事件驱动且采用框架编程的应用程序中，实现某个功能的若干代码模块是由作为平台的中间件调用，这种情形下，从实现该功能代码的初始位置开始，借助现有工具或技术，很难找全其余代码。

特征定位服务于程序理解，其用途在于：对于开发人员指定一个功能，借助特征定位技术，可在包含该功能的程序源代码中，找出实现该功能的那部分代码，并合理地组织该代码以便开发人员使用。但在现有的大部分研究中，仅仅关注如何在源代码中找出与指定功能相关的代码。故在本文中，特征定位含义为：***特征定位****是从源代码中找到****实现给定特征代码****的过程。*

1.2.3. 特征定位的输入

软件过程的各种制品都可能提供对软件维护人员理解程序有益的信息；但不同时期、不同人员、具有不同文化氛围的团队，提供这些制品类型、数量、质量差异可能很大，而维护人员所能接触到的也不尽相同，所以，研究人员研究了综合利用源代码与软件过程各种其它制品的特征定位技术。

软件过程各种制品包括：用户手册、需求文档、设计文档、源代码、源代码中注释、测试用例、测试报告、错误报告、错误修复报告等。这些都可能成为特征定位的输入，其中，除***源代码***是***必需输入***外，其余的都是可选。

1.2.4. 特征定位的输出

从特征定位的定义可知：特征定位技术的输出是实现给定特征的程序元素集合。这些程序元素在不同语言背景、不同的任务下，粒度可以不相同，程序元素可以是程序的语句、函数、类、或文件等。现存的特征定位技术的输出大多是函数或类的集合。

1.2.5. 相关研究领域

特征定位与软件可追踪性链接建立（traceability link recovery）、影响分析（impact analysis）及刻面挖掘（aspect mining）等研究领域具有较强的关联性。这里需要分析一下它们之间的相似之处和差异所在。

可追踪性链接建立技术主要研究如何建立不同软件制品（例如代码和文档）之间的可追踪性链接，而特征定位技术则只关注定位实现某一功能的代码元素，并不涉及文档等其他软件制品中的定位问题。影响分析通常是特征定位的后续工作，主要分析代码中某处修改可能会影响到的其他代码。而刻面挖掘的主要目标是在代码中，发现具有贯穿特性的关注点（cross-cutting concerns），然后判定是否需要将这些关注点转换成刻面（aspect）。刻面挖掘中的刻面在挖掘之前是不知道的，而特征定位则是需要预先知道所要定位的特征，例如已知特征的描述。

区分特征定位与相关研究领域的关系，可以更好地帮助我们定位特征定位在程序理解等软件开发活动中的地位。

1.2.6. 特征定位技术分类

本章依据所采用的技术对现存特征定位进行分类。目前，特征定位中所采用的具体技术包括：静态分析（static analysis），动态分析（dynamic analysis），模式匹配（pattern matching），信息检索 IR（Information Retrieval），自然语言处理 NLP（Natural Language Processing）等。由于采用模式匹配、信息检索与自然语言处理技术的特征定位技术所基于的假设大致相同，都是将代码看作文本然后进行分析，因此可以将采用这三种技术的特征定位技术统称为文本特征定位技术。因此，可以将特征定位技术分为静态特征定位技术、动态特征定位技术、文本特征定位技术、以及它们的组合技术。具体包括：

- 静态特征定位技术
- 动态特征定位技术
- 文本特征定位技术
- 静态与动态结合特征定位技术
- 静态与文本结合特征定位技术
- 动态与文本结合特征定位技术
- 静态、动态与文本结合特征定位技术

以下将以此为纲，介绍各类特征定位技术的研究现状。

1.3. 研究现状

1.3.1. 静态特征定位技术

基于程序静态分析的特征定位技术，其输入包含两部分：1）程序源代码；2）初始软件制品集合，它通常由开发人员给出，是浏览或搜索其它相关程序元素的起始点。

基于程序静态分析的特征定位流程如下：分析程序源代码，得到程序某种依赖图，开发人员从指定的与初始软件制品集合相关的起始点开始，借助依赖图的引导，阅读理解相应的源代码，确定与任务相关的代码，直至完成任务。这个过程如图 1-1 所示。

- **步骤1.** *构造程序依赖关系图DG；*
- **步骤2.** *选择搜索起始点；*
- **步骤3.** *阅读与理解所选的代码元素；*
  判断是否已找到足够的与特征相关的的程序元素。
  - **步骤3.1.** *已找到足够的与特征相关的的程序元素，结束。*
  - **步骤3.2.** *该代码元素与特征相关，根据程序依赖关系图DG，选择新代码元素，转步骤3继续。*
  - **步骤3.3.** *该代码元素与特征不相关，回溯并根据程序依赖关系图 DG，重新选择代码元素，转步骤3继续。*

图 1-1 基于静态分析的特征定位流程

该类方法中，常用的依赖图有系统依赖图 SDG（System Dependence Graphs）、抽象系统依赖图 ASDG（Abstract System Dependence Graphs）、函数调用图 CG（Call Graph）、数据流图 DFG（Data Flow Graph）、控制关系依赖图 CFG（Control Flow Graph）。

这类代表性工作有：Chen 与 Rajlich [Chen and Rajlich, 2000]，Robillard 与 Murphy [Robillard and Murphy, 2007]，Robillard [Robillard, 2008]，Saul 等[Saul et al., 2007]，与 Trifu [Trifu, 2008, Trifu, 2009]。

Chen 与 Rajlich [Chen and Rajlich, 2000]介绍了*抽象系统调用图* ASDG 的概念，并提出了基于 ASDG 的静态特征定位技术。在 ASDG 中，结点是函数或者全局变量，而结点之间的边则是函数之间的控制依赖（control dependencies）或者变量之间的数据流（data flow）。基于 ASDG 的特征定位技术及其应用框架如图 1-2 所示。该框架明确界定程序员活动、职责与工具应具备的功能，其过程与图 1-1 所述的基本一致。

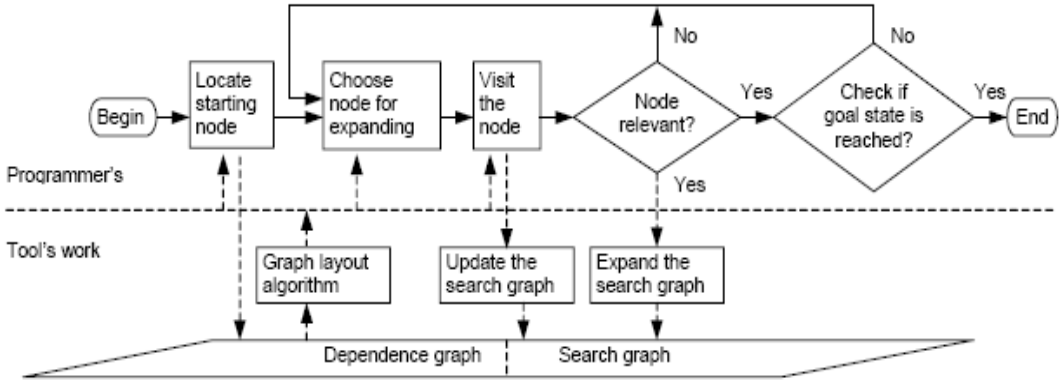

图 1-2 特征定位与代码搜索

Robillard 与 Murphy [Robillard and Murphy, 2007]开发了*关注点图*（Concern Graph），用来抽象地表示一个关注点（特征）。这种抽象方式支持建立以及保存特征与代码之间的映射关系。一个关注点图封装了程序元素（program elements）的一个子集，以及这些它们之间的关系。这种关系基于程序元素之间的静态依赖。并进一步提出了基于关注点图的静态特征定位技术，实现了名为 FEAT（Feature Exploration and Analysis Tool）的工具。

Robillard [Robillard, 2008]提出了一种基于程序中结构依赖拓扑关系（topology of structural dependencies）的静态特征定位技术。该技术中利用结构依赖的拓扑关系向开发人员推荐可能感兴趣的程序元素。与基于 ASDG 的静态特征定位技术相比，该技术可以减少开发人员的交互。具体而言，该技术允许开发人员提供一组初始的相关程序元素，进而通过分析元素之间的静态依赖关系，向开发人员推荐一组新的可能相关的程序元素。

Saul 等[Saul et al., 2007]提出了一种名为 FRAN（Finding with RANdom walks）的静态特征定位技术。该技术从开发人员提供的一个特定起点出发，建立与该起点关联的程序依赖图，发现与起点相关的程序元素，进而利用随机漫步算法计算各元素与起点之间的关联关系值，排序后将相关元素返回给开发人员。

上述方法大多同时使用程序中的控制依赖和数据依赖，Trifu[Trifu, 2008, Trifu, 2009]提出了一种仅使用数据依赖的静态特征定位技术。该技术的输入是开发人员提供的一组初始的相关变量，进而通过分析程序中的数据流，想开发人员返回更多相关的变量。

1.3.2. 动态特征定位技术

基于程序动态分析的特征定位技术，其输入包含两部分：1）程序源代码；2）测试用例集合，该集合至少包含与能执行给定特征测试用例，根据所用的分析技术也可能包含不会执行给定特征测试用例。

基于动态分析的特征定位流程如下：针对给定的特征，构造测试用例集合；插装源程序并编译成可执行代码；对每一测试用例运行程序，并收集程序运行时轨迹（Trace）；分析程序运行轨迹集合，确定与任务相关的代码。这个过程如图 1-3 所示。

- **步骤1.** *构造测试用例集合；*
- **步骤2.** *插装源程序并编译成可执行代码；*
- **步骤3.** *对每一测试用例运行程序并收集程序运行轨迹。*
- **步骤4.** *分析所得程序运行轨迹集合，确定与任务相关的代码元素。*

图 1-3 基于动态分析的特征定位流程

不同的基于动态分析的特征定位技术体现在：1）测试用例集合中除了包含与能执行给定特征测试用例外，是否还包含肯定不会执行给定特征测试用例；2）分析程序运行轨迹集合所采用的分析方法。

这类代表性工作有：Wilde [Wilde and Scully, 2006]，Wong [Wong et al., 1999]，Eisenberg [Eisenberg and De Volder, 2005]， Eisenbarth [Eisenbarth et al., 2001b]，Safyallah [Safyallah and Sartipi, 2006]，Edwards [Edwards et al., 2006]，和 Bohnet [Bohnet et al., 2008]等。

Wilde 等[Wilde and Scully, 2006]提出的*软件侦查*（Software Reconnaissance）特征定位技术是最早的特征定位技术之一。该技术的输入既包括执行某特征的测试用例，也包括不执行该特征的测试用例。该技术同时收集这两组测试用例的执行轨迹，并通过分析、比较两组测试用例的执行轨迹差异，使用确定性公式

（Deterministic Formulations）和概率公式（Probabilistic Formulations）来确定与该特征相关的程序元素。方法过程如图 1-4 所示。值得一提的是，这种通过比较执行某特征的测试用例和不执行该特征的测试用例的执行轨迹来进行特征定位的思路，被之后的动态特征定位技术研究所广泛采纳。

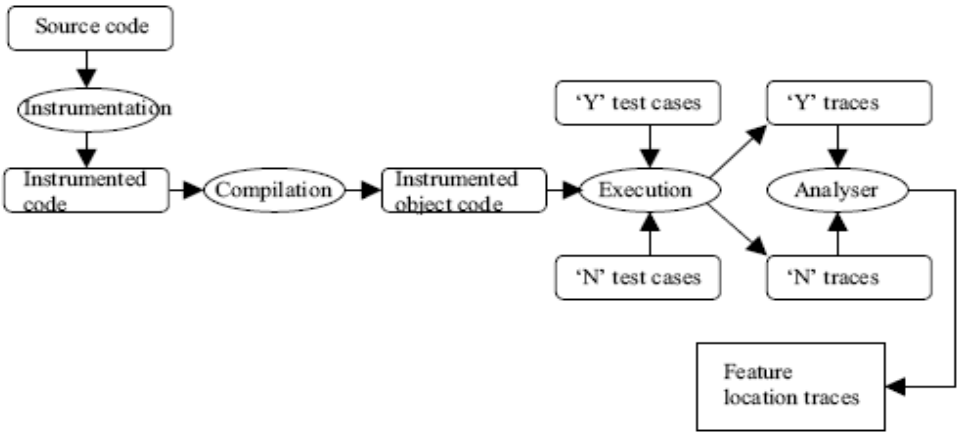

图 1-4 Wilde 的特征定位过程

Wong 等[Wong et al., 1999]提出的动态特征定位方法能细粒度地确定与特征相关的程序元素，所确定的程序元素是语句级别。其测试用例集合包含两部分：能执行给定特征的测试用例集与不会执行给定特征的测试用例集合；根据经验准则，对所得程序运行轨迹作集合交、并、差等基本运算处理；将与特征相关的程序元素划分成两部分：特定于该特征的、与非特定于该特征的。支持该方法的工具有 χVue [Agrawal et al., 1998]。

Eisenbarth [Eisenbarth 01c]提出了一种利用动态信息生成特征组件映射的技术。通过收集一组能够执行某特征的测试用例的执行轨迹，利用概念分析（concept analysis）来建立特征与组件、特征与特征之间的概念格（concept lattice）。进而通过对概念格的分析，来定位与给定特征相关联的程序元素。

Eisenberg 等 [Eisenberg and De Volder, 2005]提出动态特征轨迹 DFT（Dynamic Feature Traces）的特征定位方法。其测试用例集合只包含能执行给定特征的测试用例。分析所得程序运行轨迹集方法为：从运行轨迹中抽取函数调用者与被调用者关系对，基于三种经验值:重数（Multipliciy）、特定性（Specification）、与深度（Depth）为每个函数分配一个等级，等级越高该函数与给定特征关系越密切，其过程如图 1-5 所示。

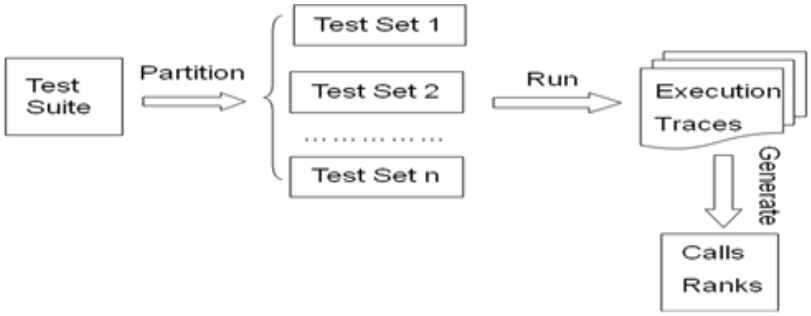

图 1-5 DFT 的特征定位过程

Safyallah 与 Sartipi [Safyallah and Sartipi, 2006]提出了利用*数据挖掘*（data mining）技术新型动态特征定位技术。该技术首先执行一组与某特征相关的测试用例，并收集其执行序列。进而，使用数据挖掘中的*序列模式挖掘*（sequential pattern mining）技术，从执行序列中挖掘出重复出现次数大于某阈值的连续的片段（continuous fragment），并将这些代码片段视为该特征关联的程序元素集合。

此外，Edwards [Edwards et al., 2006]提出分布式程序的动态特征定位技术；Bohnet [Bohnet et al., 2008] 则提供采用可视化动态执行的各种信息帮助开发人员了解特征的实现特点。

1.3.3. 文本特征定位技术

基于文本分析的特征定位技术，其输入包含两部分：1）程序源代码；2）描述给定特征的查询语句（或称特征描述）。该方法适用于程序源代码中包含描述代码特征的注释，或程序代码中标识符能反映该标识符所处上下文所实现的功能。

基于文本分析的特征定位流程如下：构造能描述给定特征的查询语句；分析代码中每一程序元素与描述特征的查询语句之间关联程度；将程序元素按关联程度从高到低的顺序排序，关联程度高的程序元素与给定特征相关的可能性大。这个过程如图 1-6 所示。

- **步骤1.** *构造描述特征的查询语句；*
- **步骤2.** *计算代码中每一程序元素与描述特征的查询语句之间关联程度；*
- **步骤3.** *按关联度排序程序元素。*

图 1-6 基于文本分析的特征定位流程

不同的基于文本分析的特征定位技术体现在：1）构造描述特征的查询语句方法不同，如自动地、半自动地、还是手工构造；2）计算代码中每一程序元素与描述特征的查询语句之间关联程度方法不同，如采用 0-1 二值的、模糊的、或其它关

联等。根据建立查询语句与关联代码之间关系所采用的技术不同，文本特征定位技术主要包括基于正则表达式的技术[Petrenko et al., 2008]、基于信息检索（Information Retrieval）的技术[Cleary and Exton, 2007, Gay et al., 2009, Poshyvanyk and Marcus, 2007, Marcus et al., 2004]、和基于自然语言处理（Natural Language Processing）的技术[Hill et al., 2007, Shepherd et al., 2006]等。

基于正则表达式的特征定位技术，特别是 *grep* 的特征定位技术，是较为早期，也是最为直接的文本特征定位技术。Petrenko 等[Petrenko et al., 2008]提出了一种基于 grep 和本体片段（ontology fragments）的特征定位技术。这里本体片段中保存了关于某特征的部分知识。Wilson [Wilson, 2010] 进一步扩展了 Perenko 的方法，提出了一种利用本体片段构造查询语句的系统化方法。

信息检索技术（IR）是一种比基于 grep 进行模式匹配更为高级的技术，可以借助 IR 进行特征定位。Marcus 等 [Marcus 04][Marcus et al., 2004]提出基于潜在语义索引 LSI（Latent Semantic Indexing）[Deerwester et al., 1990]的特征定位方法。该方法首先从代码中抽取标识符和代码注释，进而对标识符按照不同的命名规则（naming conventions）进行切词等预处理，构造一个语料集（corpus）。进一步，依据语料集中词出现的位置，可以将词分配到不同的文档中，这里的文档可以是不同粒度的，例如类或者方法。接下来，通过奇异值分解（Singular Value Decomposition, SVD）可以将语料集转换到（transform）LSI 的子空间。经过 SVD，语料集中的文档均用相应的向量（vector）代替。最后，通过将用户的查询语句也转换成向量，计算该向量与各文档对应向量之间在向量空间中的夹角大小（用夹角的余弦值表示），可以确定与该查询相关联的文档集合。该过程如图 1-7 所示。LSI 和 grep 一样容易使用，但是可以取得比 grep 更好的结果。近年来，研究人员尝试使用 LSI 的改进技术 LDA[Blei et al., 2003]来进行特征定位[Asuncion et al., 2010, Bacchelli et al., 2010]，同样取得了较好的结果。

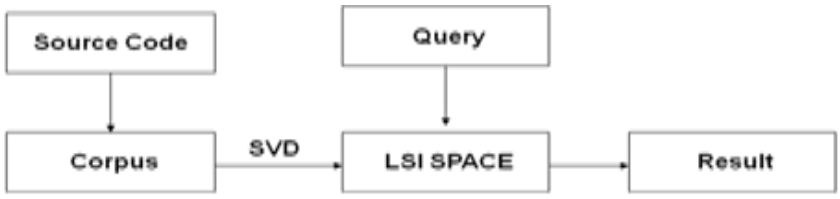

图 1-7 基于 LSI 的特征定位流程

Poshyvanyk 与 Marcus [Poshyvanyk and Marcus, 2007]提出基于 LSI 与形式化概念分析 FCA （Formal Concept Analysis）的特征定位。该方法是应用 FCA 方法分析基于 LSI 特征定位所得的程序元素，以区分特定于该特征与非特定于该特征的程序元素。

Cleary 与 Exton [Cleary and Exton, 2007]所提出的方法与[Marcus et al., 2004]原理相同，不同之处在于：Cleary 与 Exton 的方法的语料集包含了源代码以外的信息，如缺陷报告（bug report）、邮件列表、及其它外部文档。该方法具有如下特点：能找到不包含查询语句中词语但与特征相关的程序元素。

与信息检索技术类似，独立成分分析（Independent Component Analysis，ICA）[Comon, 1994]同样可以用于分析代码文本用于定位特征及其实现[Grant et al., 2008]。ICA 是一种信号分析技术，它能够将一组输入信号区分成统计上相互独立的成分。Grant 等[Grant et al., 2008]提出基于独立成分分析 ICA（Independent Component Analysis）的特征定位方法。该方法不需开发人员构造查询。它利用源代码构造如下矩阵：一行表示一个函数，一列表示代码中的一个词语，矩阵中（*i*, *j*）元素表示第 *j* 个词语在第 *i* 个函数中出现的频率。利用 ICA 对该矩阵聚类，等价于对相关的函数聚类，每一类中的函数同属同一特征，从而达到特征定位目的。

此外，可以使用自然语言处理技术对文本进行处理，进而研究特征定位技术。例如，Shepherd 等[Shepherd et al., 2006]提出基于上下文的面向行动的标识符图 AOIG（Action-Oriented Identifiers Graph）的方法。开发人员手工构造描述特征的查询语句，该查询语句是描述特征的动词及其直接宾语；程序元素与描述特征的查询语句之间关联程度为自然语言中动宾关系。该特征定位方法过程如下：首先，从源代码中抽取出所有的动宾关系对，同时，保存每一动宾关系对与程序元素的对应关系；其次，根据查询语句中的动词，找出包含该动词的动宾关系，进而找到相应的程序元素。支持该方法的工具有 Eclipse 插件 ViRMoVis [Shepherd et al., 2006]。

Hill 等[Hill et al., 2007]的方法与[Shepherd et al., 2006]类似，也是基于上下文（Context）搜索与自然语言处理技术。该方法在扩展与精化查询时使用三种短语：名词短语、动词短语、及介词短语，而不象[Shepherd et al., 2006]只使用动宾关系。

此外，Abebe 与 Tonella [Abebe and Tonella, 2010]提出了用自然语言处理技术从源代码中抽取概念的方法。Würsch 等[Würsch et al., 2010]则是利用语义网（Semantic Web）技术引导开发人员用自然语言构造高效查询语句。

1.3.4. 静态与动态结合特征定位技术

在测试及程序分析领域，静态分析技术与动态分析技术结合是一种常用且效果较好的组合方式。这一组合同样被应用于特征定位的研究中。通过动态分析技术可以有效缩小搜索空间，只关注执行序列中的相关元素；进而可以使用静态分析技术分析获得的较小元素集合中各元素之间的关系，对其进行排序或者发现额外的相关元素。

静态特征定位技术与动态特征定位技术有各自长处与不足，静态特征定位技术得到结果较保守，含较多的冗余信息，甚至，含有不相关的信息；而动态特征定位技术所得结果较精确，但往往不完整。综合利用两者于特征定位，可以取长补短得到更好的效果。研究人员尝试了这样工作。

这类技术代表性工作有：Eisenbarth 等[Eisenbarth et al., 2001a, Eisenbarth et al., 2003]，Koschke 等[Koschke and Quante, 2005]，Antoniol 等 [Antoniol and Guéhéneuc, 2006]，Rohatg [Rohatgi et al., 2009]，与 Walkinshaw [Walkinshaw et al., 2007]。

Eisenbarth 等[Eisenbarth et al., 2001a, Eisenbarth et al., 2003]提出了动态与静态结合的特征定位技术，其基本思想如下：首先，用概念格方法分析程序执行轨迹，得到特定于某给定特征的程序元素；其次，由这些程序元素作为初始集合，借助于程序的结构依赖关系，求出与该特征相关的其余程序元素。Koschke 与 Quante [Koschke and Quante, 2005]基本思想与 Eisenbarth 的方法类似，只是前者获得的程序元素粒度是语句级别。

为了解决动态数据中存在的不精确和噪音问题，Antoniol 等 [Antoniol and Guéhéneuc, 2006]提出了基于场景的概率分级 SPR（Scenario-based Probabilistic Ranking）的方法。该方法如图 1-8 所示，首先，构造与特征相关的场景集合 F 和与特征不相关的场景集合 T。其次，执行场景集合 F 中所有场景，得到区间集 I，I 中包含若干区间，而每一区间包含若干事件，此处事件即是程序元素，故可认为 I 是程序元素集；同样，执行场景集合 T 中所有场景，得到区间集 I'。接着，分别计算每一事件在 I 与 I'中的频率 f 与 f'，并据此计算每一事件的特征相关度 r。若某事件的相关度 r 大于某设定的阈值Θ，则认为该事件与特征相关；此外，SPR 通过静态分析源代码，将源代码表示成抽象对象语言 AOL，并在 AOL 中突出显示与特征相关的事件。SPR 曾用 Mozilla、Firefox、 Chimera、 ICEBrowser、JHotDraw、与 Xfig 等程序作过实验，效果明显。

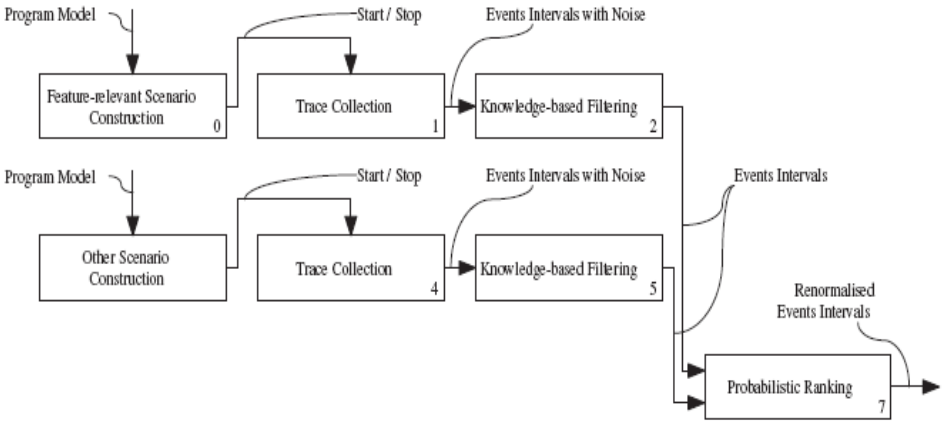

图 1-8 SPR 特征定位流程

Rohatgi 等 [Rohatgi et al., 2009]提出基于影响分析的特征定位方法。该方法基本思想如下：首先，运行一能执行给定特征的测试用例，得到程序执行轨迹，从程序执行轨迹中抽取出所涉及的所有类（class）组成集合 C。静态分析源代码得到程序类依赖图 CDG（Class Dependency Graph）。对于 C 中的每一个类 c, 根据 CDG 程序类依赖图 CDG，分析改变它对 CDG 的影响而计算它的等级值，影响越小，其等级值越高，即它越特定于给定的特征。

Walkinshaw 等 [Walkinshaw et al., 2007]提出基于函数调用图（Call Graph）切片（Slicing）的特征定位方法。该方法基本思想如下：运行一能执行给定特征的测试用例，得到程序执行轨迹，从程序执行轨迹中抽取出所涉及的所有函数，在函数调用图中将这些函数设为路标（Landmark），同时，也在函数调用图中设置路障（Barrier）；以路标及路标之间的有向路径上的函数为起点,作后向切片（Backward Slicing）,每次切片过程若遇到路障则停止，另取新起点继续作后向切片。最后，将函数调用图中所有切片以外部分移除，剩余的部分是与特征相关的函数。

1.3.5. 静态与文本结合特征定位技术

由于静态特征定位技术往往存在过拟合（overestimation）问题，而文本特征定位技术可能会遗漏掉不包含查询词语的程序元素，因此可以通过将二者结合来获得更好的效果。结合方式可以是：1）利用文本分析技术对静态分析的结果进行精化和排序；2）利用静态分析技术对文本分析得到的结果进行扩展、补充。

这类技术代表性工作有：Zhao 等 [Zhao et al., 2006]，Hill 等[Hill et al., 2007]，Ratiu 等[Ratiu and Deissenboeck, 2007]，Shao 等 [Shao and Smith, 2009]，与 Hayashi 等 [Hayashi et al., 2010]。

Zhao 等[Zhao et al., 2006]提出静态非交互式特征定位 SNIAFL 方法。该方法使用了信息检索方法与分支保留函数调用图 BRCG。图 1-9 是 SNIAFL 基本流程： 首先，利用信息检索方法 LSI 确定与给定特征相关的函数初始集合；其次，分析源代码构造程序的分支保留函数调用图 BRCG，在 BRCG 上标识与函数初始集合中函数相应的节点；从标识的节点为起点，根据依赖关系按深度优先方式遍历 BRCG 中其余节点，所有能遍历到的节点所对应的函数构成与特征相关的程序元素。在[Zhao et al., 2006]所作的实验显示该方法并单独用静态或动态的方法所得结果精度要高。

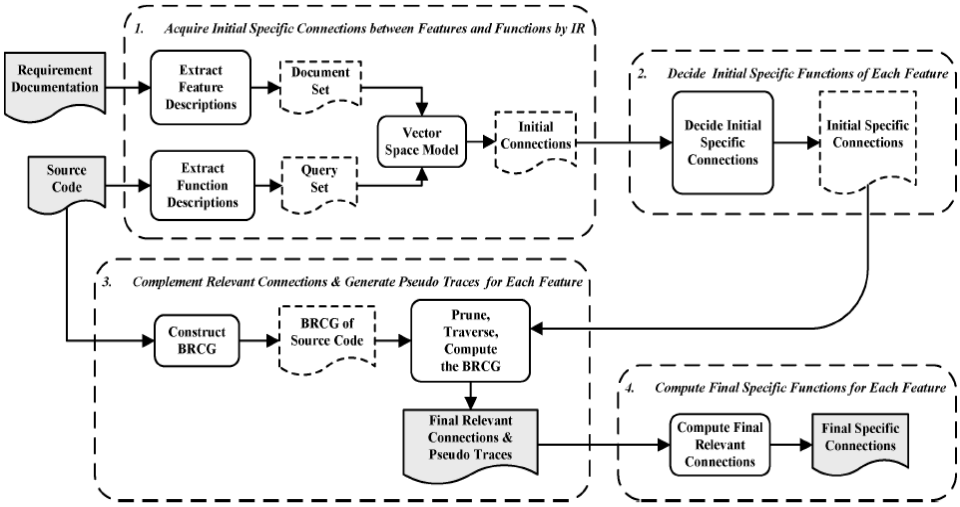

图 1-9 SNIAFL 特征定位流程

Hill 等[Hill et al., 2007]提出特征定位方法 Dora。该方法与 SNIAFL 方法类似，其基本流程为：构造查询语句；确定种子函数集合；确定特征有关的相邻函数；最后输出特征相关的函数。在确定特征有关的相邻函数时，Dora 利用文本分析与程序静态结构即函数调用者与被调用的关系，计算两函数的相关度，根据函数相关度确定有关的相邻函数。

Shao 等[Shao and Smith, 2009]提出的特征定位方法思路如下：用 LSI 方法对源代码中所有的函数按与查询相似度排队；对队中的每一函数构造函数调用图（Call Graph），并根据调用图中函数在队中出现的次数给调用图评分；对函数相关的调

用图评分和它与查询语句相似度作仿射变换得到该函数新等级，利用新的等级值重新排列函数，等级值越大的函数与特征相关越密切。

此外，Ratiu 与 Deissenboeck [Ratiu and Deissenboeck, 2007]利用通过图匹配的方法将领域知识映射到程序代码的方法实现特征定位。Hayashi 等[Hayashi et al., 2010]则是利用领域本体与有序函数调用图来实现特征定位。

1.3.6. 动态与文本结合特征定位技术

动态分析技术具有较高的准确率，而文本分析技术具有较高的召回率，因此，将两者结合起来可能会产生比单独使用其中一项技术更好的结果。由于两者均能够按照与目标特征的关联性对定位到的程序元素进行排序，因此，将两者结合对结果进行排序将是一个合理的方式。此外，可以首先利用文本分析技术获得较多的结果，然后利用动态分析技术对文本分析技术检索得到的结果进行进一步的精化。

这类技术代表性工作有：Poshyvanyk 等[Poshyvanyk et al., 2006, Poshyvanyk et al., 2007]，Liu 等[Liu et al., 2007]，Asadi 等[Asadi et al., 2010]，Revelle 等[Revelle et al., 2010]。

Poshyvanyk 等[Poshyvanyk et al., 2006, Poshyvanyk et al., 2007]提出基于执行场景（Execution Scenarios）与信息检索（Information Retrieval）进行概率分级的 PROMESIR 方法。该方法思路如图 1-10 所示：首先，分别利用 SPR[Antoniol and Guéhéneuc, 2006]技术和 LSI 技术[Marcus et al., 2004]进行特征定位；然后，利用仿射变换（affine transformation）将两种技术的排序结果进行合并，作为最终的结果。可以通过对两种方式赋予不同的权重，来影响最终的结果。

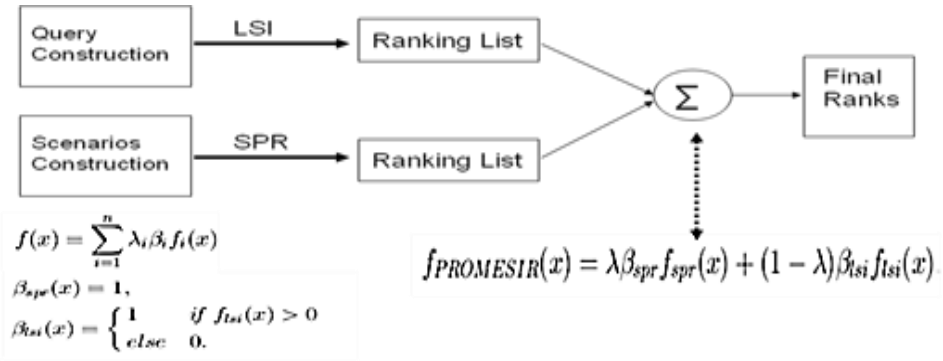

图 1-10 PROMESIR 特征定位流程

与 PROMESIR 方法类似，Liu 等[Liu et al., 2007]提出的 SITIR（Single Trace and Information Retrieval）技术也是将动态特征定位技术和文本特征定位技术相结合。

其思路是：首先，利用动态特征定位技术获得初始元素集合；然后，将初始元素集合作为输入，利用信息检索技术对输入元素进行排序返回给用户。与单纯的使用信息检索技术相比，该方法能够极大地缩小搜索空间，从而获得更加准确的结果。

此外，Asadi 等[Asadi et al., 2010]应用遗传算法分析程序执行轨迹，Revelle 等[Revelle et al., 2010]应用高级链接分析方法分析程序执行轨迹，这两项工作也属综合利用动态分析与文本分析的于特征定位方法。

1.3.7. 静态、动态与文本结合特征定位技术

Eaddy[Eaddy et al., 2008]的 Cerberus 是唯一综合利用静态分析、动态分析、与文本分析三种技术的特征定位方法。其核心技术是*依赖剪枝分析* PDA（Prune Dependency Analysis）。PDA 中假设，如果伴随着一个特征被从系统中剪掉，一个程序元素也要相应的被删减掉，那么这个程序元素与该特征之间存在关联性。Cerberus 使用 PROMESIR 技术获得初始的程序元素集合，然后利用 PDA 技术进行剪枝，获得最终的相关程序元素集合。

1.4. 参考文献

[Abebe and Tonella, 2010] ABEBE S L, TONELLA P. 2010. Natural language parsing of program element names for concept extraction[C]//Program Comprehension (ICPC), 2010 IEEE 18th International Conference on. [S.l.]: [s.n.] :156–159.

[Agrawal et al., 1998] AGRAWAL H, ALBERI J L, HORGAN J R, et al. 1998. Mining system tests to aid software maintenance[J]. Computer, 31(7):64–73.

[Antoniol and Guéhéneuc, 2006] ANTONIOL G, GUÉHÉNEUC Y G. 2006. Feature identification: An epidemiological metaphor[J]. Software Engineering, IEEE Transactions on, 32(9):627–641.

[Asadi et al., 2010] ASADI F, DI PENTA M, ANTONIOL G, et al. 2010. A heuristic-based approach to identify concepts in execution traces[C]//Software Maintenance and Reengineering (CSMR), 2010 14th European Conference on. [S.l.]: [s.n.] :31–40.

[Asuncion et al., 2010] ASUNCION H, ASUNCION A, TAYLOR R. 2010. Software traceability with topic modeling[C]//32nd ACM/IEEE International Conference on Software Engineering (ICSE). [S.l.]: [s.n.] :95–104.

[Bacchelli et al., 2010] BACCHELLI A, LANZA M, ROBBES R. 2010. Linking e-mails and source code artifacts[C]//Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering (ICSE). [S.l.]: [s.n.] :375–384.

[Blei et al., 2003] BLEI D, NG A, JORDAN M. 2003. Latent dirichlet allocation[J]. The Journal of Machine Learning Research, 3:993–1022.

[Boehm, 1984] BOEHM B W. 1984. Software engineering economics[J]. Software Engineering, IEEE Transactions on, (1):4–21.

[Bohnet et al., 2008] BOHNET J, VOIGT S, DOLLNER J. 2008. Locating and understanding features of complex software systems by synchronizing time-, collaboration-and code-focused views on execution traces[C]//Program

Comprehension, 2008. ICPC 2008. The 16th IEEE International Conference on. [S.l.]: [s.n.] :268–271.

[Chen and Rajlich, 2000] CHEN K, RAJLICH V. 2000. Case study of feature location using dependence graph[C]//8th International Workshop on Program Comprehension (IWPC). [S.l.]: [s.n.] :241–247.

[Cleary and Exton, 2007] CLEARY B, EXTON C. 2007. Assisting concept location in software comprehension[D].[S.l.]: Citeseer.

[Comon, 1994] COMON P. 1994. Independent component analysis, a new concept?[J]. Signal processing, 36(3):287–314.

[Deerwester et al., 1990] DEERWESTER S, DUMAIS S T, FURNAS G W, et al. 1990. Indexing by latent semantic analysis[J]. Journal of the American society for information science, 41(6):391–407.

[Dit et al., 2011] DIT B, REVELLE M, GETHERS M, et al. 2011. Feature location in source code: A taxonomy and survey[J]. Journal of Software Maintenance and Evolution: Research and Practice.

[Eaddy et al., 2008] EADDY M, AHO A V, ANTONIOL G, et al. 2008. Cerberus: Tracing requirements to source code using information retrieval, dynamic analysis, and program analysis[C]//Program Comprehension, 2008. ICPC 2008. The 16th IEEE International Conference on. [S.l.]: [s.n.] :53–62.

[Edwards et al., 2006] EDWARDS D, SIMMONS S, WILDE N. 2006. An approach to feature location in distributed systems[J]. Journal of Systems and Software, 79(1):57–68.

[Eisenbarth et al., 2001a] EISENBARTH T, KOSCHKE R, SIMON D. 2001a. Aiding program comprehension by static and dynamic feature analysis[C]//Software Maintenance, 2001. Proceedings. IEEE International Conference on. [S.l.]: [s.n.] :602–611.

[Eisenbarth et al., 2001b] EISENBARTH T, KOSCHKE R, SIMON D. 2001b. Feature-driven program understanding using concept analysis of execution traces[C]//Program Comprehension, 2001. IWPC 2001. Proceedings. 9th International Workshop on. [S.l.]: [s.n.] :300–309.

[Eisenbarth et al., 2003] EISENBARTH T, KOSCHKE R, SIMON D. 2003. Locating features in source code[J]. Software Engineering, IEEE Transactions on, 29(3):210–224.

[Eisenberg and De Volder, 2005] EISENBERG A D, DE VOLDER K. 2005. Dynamic feature traces: Finding features in unfamiliar code[C]//Software Maintenance, 2005. ICSM'05. Proceedings of the 21st IEEE International Conference on. [S.l.]: [s.n.] :337–346.

[Fjeldstad and Hamlen, 1983] FJELDSTAD R K, HAMLEN W T. 1983. Application program maintenance study: Report to our respondents[J]. Proceedings Guide, 48.

[Gay et al., 2009] GAY G, HAIDUC S, MARCUS A, et al. 2009. On the use of relevance feedback in IR-based concept location[C]//Software Maintenance, 2009. ICSM 2009. IEEE International Conference on. [S.l.]: [s.n.] :351–360.

[Grant et al., 2008] GRANT S, CORDY J R, SKILLICORN D. 2008. Automated concept location using independent component analysis[C]//Reverse Engineering, 2008. WCRE'08. 15th Working Conference on. [S.l.]: [s.n.] :138–142.

[Hayashi et al., 2010] HAYASHI S, YOSHIKAWA T, SAEKI M. 2010. Sentence-to-code traceability recovery with domain ontologies[C]//Software Engineering Conference (APSEC), 2010 17th Asia Pacific. [S.l.]: [s.n.] :385–394.

[Hill et al., 2007] HILL E, POLLOCK L, VIJAY-SHANKER K. 2007. Exploring the neighborhood with Dora to expedite software maintenance[C]//Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering. [S.l.]: [s.n.] :14–23.

[Koschke and Quante, 2005] KOSCHKE R, QUANTE J. 2005. On dynamic feature location[C]//Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering. [S.l.]: [s.n.] :86–95.

[Liu et al., 2007] LIU D, MARCUS A, POSHYVANYK D, et al. 2007. Feature location via information retrieval based filtering of a single scenario execution trace[C]//Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering. [S.l.]: [s.n.] :234–243.

[Marcus et al., 2004] MARCUS A, SERGEYEV A, RAJLICH V, et al. 2004. An information retrieval approach to concept location in source code[C]//Reverse Engineering, 2004. Proceedings. 11th Working Conference on. [S.l.]: [s.n.] :214–223.

[Petrenko et al., 2008] PETRENKO M, RAJLICH V, VANCIU R. 2008. Partial domain comprehension in software evolution and maintenance[C]//Program Comprehension, 2008. ICPC 2008. The 16th IEEE International Conference on. [S.l.]: [s.n.] :13–22.

[Poshyvanyk and Marcus, 2007] POSHYVANYK D, MARCUS A. 2007. Combining formal concept analysis with information retrieval for concept location in source code[C]//Program Comprehension, 2007. ICPC'07. 15th IEEE International Conference on. [S.l.]: [s.n.] :37–48.

[Poshyvanyk et al., 2006] POSHYVANYK D, GUÉHÉNEUC Y G, MARCUS A, et al. 2006. Combining probabilistic ranking and latent semantic indexing for feature identification[C]//Program Comprehension, 2006. ICPC 2006. 14th IEEE International Conference on. [S.l.]: [s.n.] :137–148.

[Poshyvanyk et al., 2007] POSHYVANYK D, GUÉHÉNEUC Y G, MARCUS A, et al. 2007. Feature location using probabilistic ranking of methods based on execution scenarios and information retrieval[J]. Software Engineering, IEEE Transactions on, 33(6):420–432.

[Ratiu and Deissenboeck, 2007] RATIU D, DEISSENBOECK F. 2007. From reality to programs and (not quite) back again[C]//Program Comprehension, 2007. ICPC'07. 15th IEEE International Conference on. [S.l.]: [s.n.] :91–102.

[Revelle et al., 2010] REVELLE M, DIT B, POSHYVANYK D. 2010. Using data fusion and web mining to support feature location in software[C]//Program Comprehension (ICPC), 2010 IEEE 18th International Conference on. [S.l.]: [s.n.] :14–23.

[Robillard, 2008] ROBILLARD M P. 2008. Topology analysis of software dependencies[J]. ACM Transactions on Software Engineering and Methodology (TOSEM), 17(4):18.

[Robillard and Murphy, 2007] ROBILLARD M P, MURPHY G C. 2007. Representing concerns in source code[J]. ACM Transactions on Software Engineering and Methodology (TOSEM), 16(1):3.

[Rohatgi et al., 2009] ROHATGI A, HAMOU-LHADJ A, RILLING J. 2009. Approach for solving the feature location problem by measuring the component modification impact[J]. Software, IET, 3(4):292–311.

[Rugaber, 1995] RUGABER S. 1995. Program comprehension[J]. Encyclopedia of Computer Science and Technology, 35(20):341–368.

[Safyallah and Sartipi, 2006] SAFYALLAH H, SARTIPI K. 2006. Dynamic analysis of software systems using execution pattern mining[C]//Program Comprehension, 2006. ICPC 2006. 14th IEEE International Conference on. [S.l.]: [s.n.] :84–88.

[Saul et al., 2007] SAUL Z M, FILKOV V, DEVANBU P, et al. 2007. Recommending random walks[C]//Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering. [S.l.]: [s.n.] :15–24.

[Shao and Smith, 2009] SHAO P, SMITH R K. 2009. Feature location by IR modules and call graph[C]//Proceedings of the 47th Annual Southeast Regional Conference. [S.l.]: [s.n.] :70.

[Shepherd et al., 2006] SHEPHERD D, POLLOCK L, VIJAY-SHANKER K. 2006. Towards supporting on-demand virtual remodularization using program graphs[C]//Proceedings of the 5th international conference on Aspect-oriented software development. [S.l.]: [s.n.] :3–14.

[Trifu, 2008] TRIFU M. 2008. Using dataflow information for concern identification in object-oriented software systems[C]//Software Maintenance and Reengineering, 2008. CSMR 2008. 12th European Conference on. [S.l.]: [s.n.] :193–202.

[Trifu, 2009] TRIFU M. 2009. Improving the dataflow-based concern identification approach[C]//Software Maintenance and Reengineering, 2009. CSMR'09. 13th European Conference on. [S.l.]: [s.n.] :109–118.

[Walkinshaw et al., 2007] WALKINSHAW N, ROPER M, WOOD M. 2007. Feature location and extraction using landmarks and barriers[C]//Software Maintenance, 2007. ICSM 2007. IEEE International Conference on. [S.l.]: [s.n.] :54–63.

[Wilde and Scully, 2006] WILDE N, SCULLY M C. 2006. Software reconnaissance: mapping program features to code[J]. Journal of Software Maintenance: Research and Practice, 7(1):49–62.

[Wilde et al., 1992] WILDE N, GOMEZ J A, GUST T, et al. 1992. Locating user functionality in old code[C]//Software Maintenance, 1992. Proceerdings., Conference on. [S.l.]: [s.n.] :200–205.

[Wilson, 2010] WILSON L A. 2010. Using ontology fragments in concept location[C]//Software Maintenance (ICSM), 2010 IEEE International Conference on. [S.l.]: [s.n.] :1–2.

[Wong et al., 1999] WONG W E, GOKHALE S S, HORGAN J R, et al. 1999. Locating program features using execution slices[C]//Application-Specific Systems and Software Engineering and Technology, 1999. ASSET'99. Proceedings. 1999 IEEE Symposium on. [S.l.]: [s.n.] :194–203.

[Würsch et al., 2010] WÜRSCH M, GHEZZI G, REIF G, et al. 2010. Supporting developers with natural language queries[C]//Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1. [S.l.]: [s.n.] :165–174.

[Zhao et al., 2006] ZHAO W, ZHANG L, LIU Y, et al. 2006. SNIAFL: Towards a static noninteractive approach to feature location[J]. ACM Trans. Softw. Eng. Methodol., 15(2):195–226. http://doi.acm.org/10.1145/1131421.1131424.

第二章　问答系统

2.　问答系统

2.1.　背景

互联网已经成为网络用户们保存数据、交流信息以及共享知识的大平台。如今，大量数字产品网站被创建以满足用户日常生活中的各种需求，网页的数量仍然保持高速无限制的增长。Brin 曾如此定义万维网：“万维网是一个大量的完全无法控制的多种多样的文档的聚集地“[Brin 1998]。但是，在如此浩瀚的互联网中，用户如何得知网站的信息并访问这些网站，如何从相关信息服务中提取自己所需要的信息则成为了一大挑战。管理、检索、过滤海量信息的技术，成了一件亟需解决却又非常困难的事情。针对这种情况，广大学者们开始研究为人们在互联网资源中导航并快速访问所需的相关网页的技术。自此，处理海量数据的 Web 信息检索技术被提出并逐渐被计算机界所重视[Salton 1986]。信息检索，是指将信息按一定的方式组织和存储起来，并根据用户的需要找出有关的信息的过程。

搜索引擎是一种最常见的互联网信息检索技术，其基本思想是：使用自动搜索爬虫来遍历互联网，将互联网上分布的信息下载到本地文档库；然后对文档内容进行自动分析并建立索引；对于用户提出的检索请求，搜索引擎通过检查索引找出匹配的文档并返回给用户。当前，搜索引擎是用户在互联网上寻找所需信息的最主要方式。然而，从与用户交互的方面来看，用户在使用搜索引擎的时候，经常遇到如下的问题：

1) 并不清楚到底怎么样的词语序列才可以返回精确的结果；
2) 可能接收到无关的返回信息；
3) 对搜索引擎通用的布尔逻辑不太理解，更不用说使用高级检索了；
4) 新用户还需要花时间来熟悉搜索引擎的使用方法；
5) 85%的用户只看返回的第一页面，而一般的搜索引擎返回的相关性信息太多，这对排序方法提出了很高的要求。

同时，随着技术发展，用户的需求也不断提高。而关键词的逻辑组合来表达检索需求，难以处理用户复杂的检索要求。在很多情况下，用户所需的信息并不是直接显示表达在互联网文档中的。而传统以关键词为基础的索引、匹配算法停留在语言表层，没有触及语义。那么，如何满足用户进一步的需求，提高检索的准确率以

及检索结果的质量，成为现今信息检索领域研究的新课题。要让机器不仅“认识“自然语言文本，还要”理解“自然语言文本；不仅从大规模文档中找出相关文档，还要从相关文档中找出相关的部分。

问答系统是人们探索机器智能的一个应用平台，期望机器可以在一定程度上理解并运用自然语言。面对一个以自然语言表述的问题，一个问答系统的任务就是在给定的文档中找出正确的答案，并以自然语言的表达方式返回答案。从技术的角度看，它是信息检索的一种特殊方式，但这种系统与原先的信息检索系统并不相同。当前的信息检索系统可以做到返回可能含有相关信息的整篇文档，让用户自己从中去寻找答案。从功能上看，信息检索系统旨在在海量的文档中为用户大幅度地降低信息定位的范围；而问答系统则旨在进一步地从中精准地获取问题的答案。从处理技术上看，信息检索系统由于需要处理海量的文档，无法对文本进行深入分析处理，通常只能利用统计及关键词匹配等技术；而问答系统则往往需要借助自然语言处理技术，对文本中的句子或篇章进行深入的语义分析，从而从信息检索系统所返回的粗粒度结果中准确的答案。信息检索系统与问答系统之间的差异，也恰恰体现出了二者之间的互补性。在问答系统的技术发展过程中，信息检索技术发挥了重要的作用。本章将从信息检索系统的视角出发，对基于自然语言的自动问答系统进行综述。

2.2. 基本概念

问答系统 关于问答系统的内涵和外延，很多的研究者都给出了各自的定义。例如，[Mollá2007]将问答系统定义为一个能回答任意自然语言形式问题的自动机。虽然定义很多，并且各种定义之间略有不同，但是一般都认为问答系统是一种智能化的搜索引擎，其输入应该是自然语言形式的问题，输出应该是一个精确、简洁的答案或者可能答案的列表，而不是一堆相关的文档。

问题 在问答系统中，一个问题指的是一个清楚地描述了用户的信息需求的自然语言句子。问题一般是一个以疑问词开头的疑问句，有的时候也可能是一个以动词开头的祈使句。相比于信息检索系统以搜索关键词组成的简单列表作为输入，自然语言句子形式的问题中还蕴含有关键词之间的句法结构信息，并能据此体现出其间的语义关联。对于用户提出的自然语言问题，可以将它们大致分为八大种类：事实类问题、列表类问题、定义类问题、假设类问题、原因类问题、关系类问题、过程类问题、确认类问题[Kolomiyets 2011]。以下对这些种类的问题做出简单解释：

- 事实类问题：一般指要求回答实体、时间、地点、数量等事实性内容的问题。例如：“奥巴马的夫人是谁？“；”2014 年世界杯的举办地是哪里？“；“水的沸点是多少度？“；等等。
- 列表类问题：与事实类问题的目的相似，但要求的不是一个答案，而是一组答案。例如：“中国有哪些邻国？”；“列出三种原产于澳大利亚的动物”；等等。
- 定义类问题：要求对某个实体进行解释。更具体地，可以分为描述类问题和观点类问题。描述类问题指的是要求给出实体的具体定义；观点类问题则指的是要求给出对实体的看法或评价。
- 假设类问题：指的是问题中陈述了一些假设作为前提约束条件，例如：“如果核战争爆发了，我们应该怎么办？”。
- 原因类问题：要求解释事件原因的问题，一般指以“为什么”开头的问题；
- 关系类问题：要求给出两个实体之间的关系，例如：“以色列和巴勒斯坦是什么关系？”；
- 过程类问题：问题给出了一个任务，要求答案给出解决这一任务所需要执行的具体步骤的列表，例如：“怎么做西红柿炒鸡蛋？”；
- 确认类问题：可以用“是”或则“否”回答的问题，即特殊疑问句。

信息来源 问答系统的信息来源指的是用于抽取答案的信息对象的集合。

从信息的表示形态方面，问答系统的信息来源可以大致分为**非结构化数据**和**结构化数据**两种[Moens 2006]。非结构化数据指的是诸如自然语言文本、图像、音频、视频等机器难以理解与解释其语义的数据。在问答系统中，非结构化数据在大多数情况下指的是自然语言文本形式的文档集合。结构化数据则指的是数据中的实体及实体间的关联关系得到明确描述，并可以被机器识别与理解的信息来源。典型的结构化数据包括：关系型数据库、专家系统中的三元组知识库、本体网、等等，一般通过专家编辑或是从非结构化数据中进行信息抽取而形成。近年来，互联网上也逐渐出现了一些基于众包编辑的大规模的结构化知识库，如：谷歌知识图谱、DBpedia、Freebase 等。相比于非结构数据，结构化数据所表示的语义信息更加明确，更有利于机器处理，但在信息的数量与丰富度上却远远低于非结构数据。目前，从信息来源的维度上看，问答系统的相关研究工作可以划分为以非结构化数据为信息来源的问答系统和以结构化数据为信息来源的问答系统。前者立足于信息检

索，主要思想是在海量的自然语言文本文档中搜索出小范围的与问题相关的文档，再从文档中抽取出问题的答案；后者立足于知识表示与推理，主要思想是对非结构化数据中的信息进行结构化表示，并将问题转换为结构化数据上的逻辑语句，并通过逻辑推理求解出问题的答案。

从信息来源的范围方面，问答系统可以分为**限定领域问答系统**与**开放领域问答系统**。限定领域指的是系统能接受的问题只能是关于某个特定的主题的，开放领域指的是系统能接受的问题可以是任意主题的问题，没有任何限制。限定领域的问答系统与基于非结构化数据的问答系统具有很高的重合性：早期的问答系统多为限定领域的问答系统，其信息来源多是由专家编辑构造的结构化数据库或知识库。最著名的限定领域问答系统是 BASEBALL[Green 1961]和 LUNAR[Woods 1977]。BASEBALL 系统能回答关于美国棒球联赛一个赛季的相关问题；LUNAR 系统能回答关于阿波罗月球探测任务取回的岩石样本分析结果的相关问题。目前，限定领域的问答系统仍然被大量使用，典型的应用场景是生物医药领域，例如：[Zweigenbaum 2003], [Sang 2005], [Moll á2007]，等等。此类问答系统又被称为 AI 系统、专家系统，或是数据库的自然语言接口系统，其相关研究的活跃时期为 20 世纪 60 到 80 年代。近年来，此类技术没有大的突破，因此本文不对其进行详细介绍。进入 20 世纪 90 年代，由于互联网的飞速发展，产生了海量领域开放、以无结构自然语言文本为主要形式的文档数据，这为问答系统进入开放领域提供了客观条件。信息检索评测组织 TREC 自 1999 年开始每年都设立开放领域问答的评测任务[Dang 2007]，同时其它评测组织如 NTCIR 和 CLEF 也设置有开放领域问答系统的评测任务，极大地推动了开放领域问答系统的研究进展。本文将以开放领域问答系统为主线进行阐述。

2.3. 问答系统的基本结构

Moldvan 等人在 LASSO 问答系统中提出，问答系统有三个最基本的组成成分：问题处理模块、段落抽取/索引模块和答案处理模块（[Moldovan 1999], [Moldovan 2000]）。现有的问答系统大多是以这样的基本模式为基础进行拓展的。图 2-1 展示了问答系统的基本结构图。每一个问题，依次经过三个模块，得到答案返回。问题处理模块通过文本处理技术分析用户新提出的问题，以抽取用户的需求并利用某种表达方式将这种需求信息表达为查询，作为段落抽取/索引模块的输入。索引模块负责从数据库中搜集并管理所有可利用的文档，这些文档可以从互联网中获得，也可以从本地的数据源中获得。段落抽取模块则将索引得到的文档

进行分段处理；然后依据这一模块的输入，从中抽取出所有与查询相关的段落；最后，对所有得到的段落进行评估，并将最相关的若干段落作为答案处理模块的输入。答案处理模块则是通过信息抽取技术从这些相关段落中抽取出若干候选答案，通过评估排序获得针对该问题较为可信的答案。

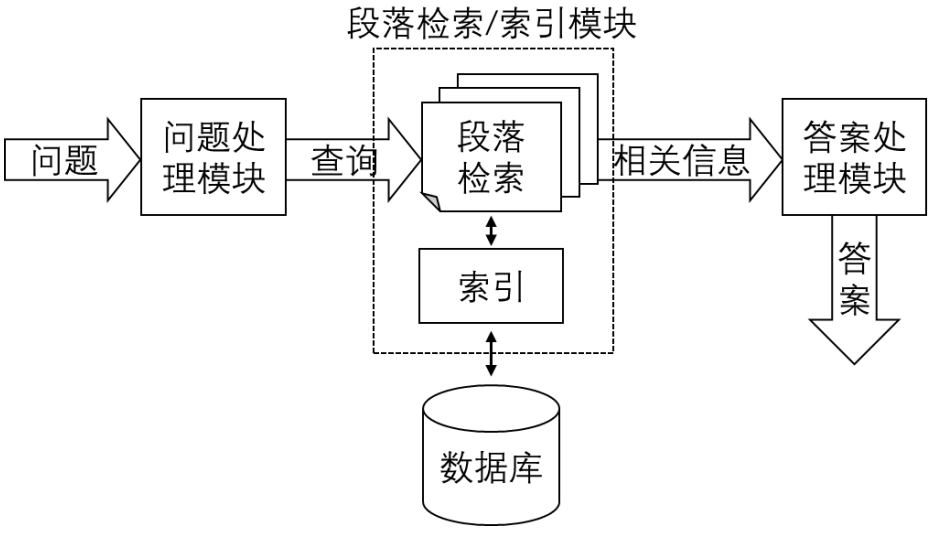

图 2-1 问答系统结构图

在接下来的三个小节中，我们分别从这三个基本模块出发，分别介绍自动问答系统中各个模块的任务、核心研究点与相关工作。

2.3.1. 问题处理模块

问题处理模块的主要目标，是找出问题句中对所需答案的约束信息，以帮助后续模块中的信息检索和信息抽取任务的开展。很多工作致力于根据问题句找出期望的答案中可能包含的关键词、答案可能的表达形式等等。例如，给定一个自然语言表达的问题作为输入，在 LASSO 中，问题处理模块需要完成以下几部分的工作：

1) 确定问题的类型；
2) 确定期望得到的答案的类型；
3) 确定问题的焦点；
4) 得到问题的关键字的集合，作为段落索引模块的查询输入。

2.3.1.1. 问题分类

问题分类指根据问题的答案类型对问题进行分类。开放领域的问题可以千变万化，但是同种类型的问题不管是从形式上看还是从内容上看，都是一个相对较小的

集合。在较小且有共性的集合中，文本处理的方法就有许多共性。因此，问题分类的制定和问题类型的识别就成为了问题处理模块中最重要的功能之一。目前，大多数问答系统都利用答案类型来指导后续的步骤，尤其是答案抽取策略。例如，对于问人物的问题，答案抽取会利用人物的各种特征来提取答案候选集合。

一般问题分类可以通过疑问词直接决定问题的类型，例如，对于含有“谁”的问题，可以认为需要的答案类型是人物。但这种方法的粒度过于粗糙，难以满足问答系统的实际需求。例如，对于“什么”、“怎么”这样的疑问词，可以对应非常多的答案类型；同时，还有一些问题不包含疑问词。针对这些问题，[Li 2002]提出了更加详细的分类。该问题分类体系中有 6 个大类（略缩语、描述、实体、人物、地点、数量），在 6 个打雷下面又分了 50 个小类。例如，在数量类中，有距离、钱等小类。图 2-1 展示了该问题分类体系，并在 TREC 数据集中的 1000 个问题中统计了各个问题类型的分布情况。在问题类别的制定中，某些问题类型的边界一直存在争论，使得一些问题的分类模棱两可。允许同一个问题有多个问题类型标记，可以部分地消除这样的模棱两可[Lampert 2004]。

Class	#	Class	#
ABBREVIATION	18	term	19
abbreviation	2	vehicle	7
expression	16	word	0
DESCRIPTION	153	**HUMAN**	171
definition	126	group	24
description	13	individual	140
manner	7	title	4
reason	7	description	3
ENTITY	174	**LOCATION**	195
animal	27	city	44
body	5	country	21
color	12	mountain	5
creative	14	other	114
currency	8	state	11
disease/medicine	3	**NUMERIC**	289
event	6	code	1
food	7	count	22
instrument	1	date	146
lang	3	distance	38
letter	0	money	9
other	19	order	0
plant	7	other	24
product	9	period	18
religion	1	percent	7
sport	3	speed	9
substance	20	temp	7
symbol	2	vol.size	4
technique	1	weight	4

图 2-2 问题分类及其在 TREC 数据集上的分布

问题分类的任务就是把一个问题自动划分到已有的分类结构中的一个或几个类。问题分类的方法主要包括模式匹配方法和机器学习方法两类。模式匹配方法为每一种问题类型建立一个模式集合，对于一个问句，只要与某种文体类型对应的模式相匹配，就被认为是这种类型的问题。机器学习方法首先定义一个问题的特征集合，然后在训练数据上得到一个分类器，就可以对新的问题进行分类了。[Zhang 2003]使用的是问题中的表层 n-gram 特征，并调研了 K-近邻算法、决策树、朴素贝叶斯等多种分类方法。在这些分类方法中，支持向量机在问题分类任务上的表现最好。[Li 2002]使用的特征更为深此次，包括语法（词性、词组）和语义（解释、近义词）信息。首先，顶层分类器把问题归类在一个大类别中；然后，根据该大类内的分类器，再将问题具体分到小类别中。这一方法的示意图如图 2-3 所示，在问题分类任务上取得了很好的分类效果。

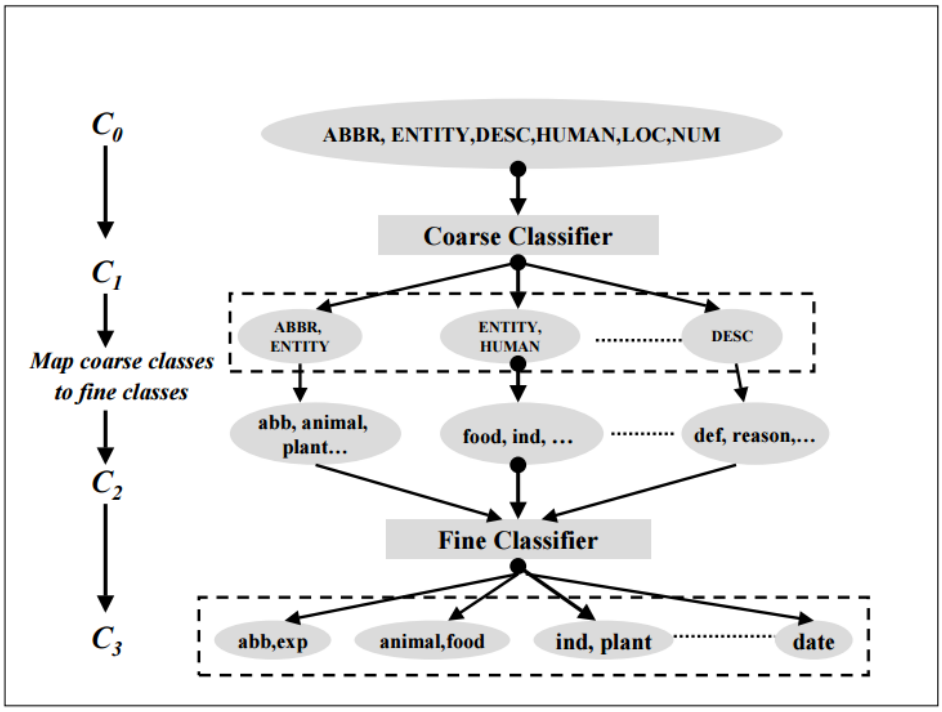

图 2-3 基于层次结构分类器的问题分类方法

2.3.1.2. 问题模版

一般来说，只知道问题的类型不足以找出正确的答案。为了从问题中抽取相关信息以表征用户的需求，在问题处理这一模块中还需要提取**问题模版**。问题模版是一种抽象化的归纳表现形式。在问题分类将具有相似语法结构和相同语义信息的问

题归为一类之后，每一类的问题将被映射到对应的模版上，从而可以通过统一的策略进行处理。因此，一个问题模版特征实际上是一种由语法语义信息到问题处理方案的映射信息，任何匹配上该模版的问题都可以依照该方案进行处理。现有系统大多通过频繁结构抽取语法及语法约束，为每一类问题构建一个问题模版，然后为该模版制定若干答案抽取模式。当系统接收到一个新问题的时候，首先会通过模版匹配搜寻到该问题的问题模版，而后得到该问题所对应的答案模版，最后利用答案模版从文档集中抽取出该问题的答案。例如，对于问题："谁是第一个实现太空行走的俄罗斯宇航员？"（"Who was the first Russian astronaut to walk in space?"），[Harabagiu 2000]首先根据疑问词 Who 识别出该问题是一个 PERSON 类的问题，随后从该问题的句法树中抽取信息，以映射到问题模版上。图 2-4 对该问题的句法树以及最终映射到的问题模版的信息进行了展示。

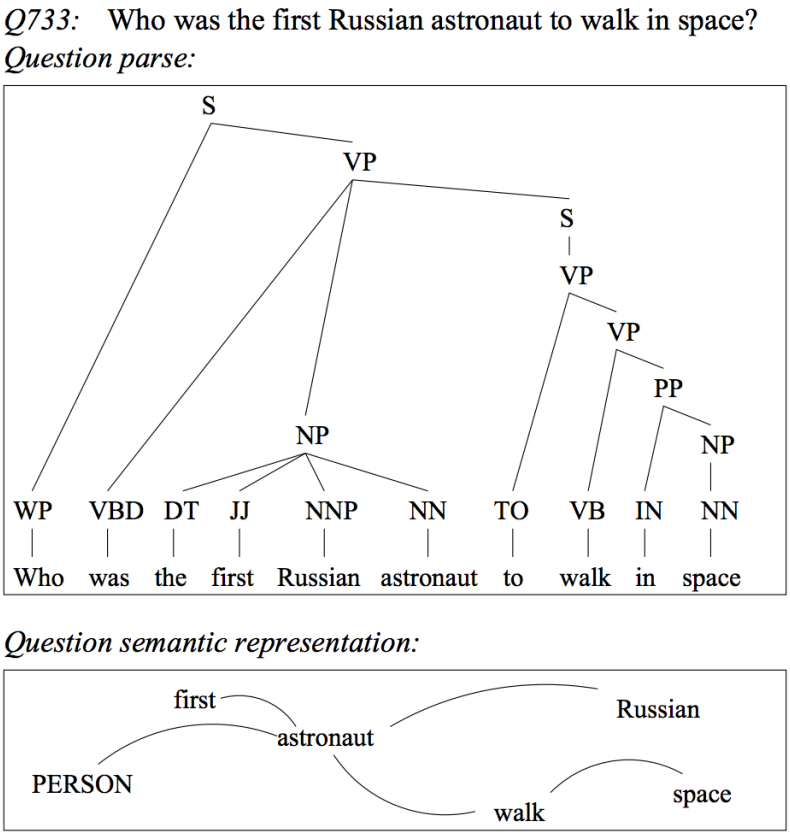

图 2-4 问题模版信息抽取示例

早期的问答系统中，问题模版多为浅层模版。浅层模版只是对语法结构作出限制，但是不具有语义约束信息，不能够区分结构相同但语义不同的问题。然而，很

多情况下具有相同结构的问题却有不同的语义，而且抽取答案的策略也截然不同。因此，随着技术的发展，问答系统逐渐使用语义模版来取代浅层模版。语义模版是在浅层模版的基础上添加语义约束信息，将浅层模版进一步细化。语义模版用一些语义标签将覆盖的问题集进一步分割成不同的子集；而后针对每一个子集，再分别制定答案模版，这样通过分情况处理的方法保证了答案的精确性。[Hao 2008] 给出了语义模版的形式化定义，如图 2-5 所示。模版中的核心元素包括：问题目标、问题类型、概念、事件、约束。图 2-6 给出了这一形式化定义下的语义模版的一个使用示例。

Semantic Pattern ::= (**<Question_Target>,** <Question_type>, [Concept], [Event], {Constraint})

(1) <Question_Target>

<Question_Target> ::= <Target: Class1\Class2>

Where "<" and ">" are two labels and must occur synchronously. "Class1\Class2" represents two level classes of the question target, which satisfy the relation SubClass (Class1, Class2).

(2) <Question_Type>

<Question_Type> ::= <Type: Question Type >

Where "<" and ">" are two labels and must occur synchronously. "Question Type" represents the type of question, such as "What", "Where", "When" and so on.

(3) [Concept]

[Concept] ::= [Concept1\Concept2]

Where "[" and "]" are two labels and must occur synchronously. "Concept1\Concept2" represents the concept relationship in the concept hierarchy, where they satisfy the relation Subcategory (Concept1, Concept2).

(4) [Event]

[Event] ::= [Event1\Event2]

Where "[" and "]" are two labels and must occur synchronously. "Event1\Event2" represents the concept relationship in the event hierarchy, where they satisfy the relation Subcategory (Event1, Event2).

(5) {Constraint}

{Constraint} ::= {O: Concept or Event; F: Facet}

Where "{", ";", "}", "O" and "F" are five labels which must occur in one constraint. "O" represents an object of constraint and "F" represents the aspect of the constraint.

图 2-5 语义模版的形式化定义

Question: *"What book did Rachel Carson write in 1962?"*
Semantic Pattern: *<Target: Entity\Product> <Type: What> [Physical_Object\Product] did [Physical_Object\Human] [Event\Action]?*
Answer Pattern: *[Entity\Product]*
Constraint: *{O: write F: Date=1962}*

图 2-6 语义模版的使用示例

有了语义模版，就需要通过**语义关系抽取**，自动地把每个问题中所蕴含的用户需求通过若干有语义信息标注的词条表达出来。经过语义信息抽取过程，每个问题从初识的语法结构约束中脱离出来，在统一的语义空间中重新定位。现有的语义关系抽取技术大都是以自然语言处理结果为基础的，例如：基于依存树分析的方法（[Liu 2007], [Bouma 2002]）；基于语义角色标注分析的方法（[Carreras 2005], [Hacioglu 2004]）；基于规则分析的方法（[Hu 2007]）；等等。

2.3.2. 段落检索/索引模块

段落检索/索引模块的主要目的是缩小答案的范围，提高下一步答案抽取的效率和精度。

缩小答案范围的最简单方法是去掉问题中的停用词和问句相关的词（如疑问词），生成查询，然后利用已有的信息检索模型进行检索，把返回的结果作为答案提取部分的输入。这种方法很难获得较好的效果，而文档检索的效果会直接影响到问答系统的整体性能[Collins 2004]。如果检索系统的相关性精度较差，那么会有大量无关文档需要后续处理，而答案提取通常采用复杂的自然语言处理技术，这必然导致系统整体效率低下。如果检索系统的召回率较低，那么很多包含答案的文档或者段落没有被返回，显然含有正确答案的文档或者段落越少，提取出正确答案的可能性也越小，这会导致系统整体性能较差。

2.3.2.1. 关键词提取

为了找到与问题相关的文本文档集合，首先需要对问题进行**关键词提取**，以作为信息检索的输入。最简单的方法是删除问题中的停用词，将其余的词作为关键词，并使用 TF-IDF 等统计学方法来度量每个关键词的重要程度。在此基础上，有一些研究工作旨在通过提高关键词提取的质量，从而提升问答系统的精度。一方面，随着 WordNet 和 HowNet 等同义词典的应用，可以从语义层面对关键词集合加以拓展。[Dave 2003]利用 WordNet 等语义词典获取与关键词集合中的词条的语

义相近的其它词条，再将这些新得到的词条加入到关键词集合中，这样可以解决同义异构的问题，有效地挖掘出潜在的答案，提高了答案抽取的召回率；另一方面，考虑关键词数量对信息检索系统的影响：当关键词数量很多时，检索返回的文档较少，召回率就会偏低；当关键词较少时，检索返回的文档偏多，相关性文档的精度就会很低。因此，针对这种关键词数量选择上的不确定性，可以动态地对关键词进行调整。[Moldovan 2000]采用这种迭代式调整技术，多次检索，根据返回文档的多少，调整关键词集合，决定是否增删关键词以及是否采用词形、句法或者语义级别的扩展形式。此外，从答案的角度看：问答系统的目的是找出一个问题的答案，而上述方法是找一个和问题相关的文档。因此，如果从一个问句推测它的答案中可能包含的关键词，用这些关键词来进行查询，会得到更好的效果。对于特定类别的问题，可以从训练数据中学习得到这类问题的回答模式，根据得到的模式从问题中生成包含答案关键词的查询[Agichtein 2004]。在从问题中抽取得合适的关键词集合之后，即可使用信息检索技术获取与问题相关的文档。信息检索领域常用的模型包括布尔模型、向量空间模型、语言模型、概率模型等。实验发现，在问答系统的文档检索中，简单的布尔模型、概率模型与改进的向量空间模型的效果相当（[Moldovan 2003], [Tellex 2003]）。

2.3.2.2. 段落检索

信息检索系统的检索粒度通常是整篇文档。这种做法往往会返回过多的信息，用户很难从中快速找出自己所需的信息。从自动的方法来看，从一片长文本文档中定位答案并抽取的过程也较为复杂。在这样的情况下，味了减少后续模块中被处理的文本文档的大小，为了方便管理、准确定位，段落检索的概念开始为人们所重视（[Liu 2002], [Cui 2005]）。该模块首先从海量数据库中检索出可能包含答案的若干相关文档；然后，从这些文档中过滤掉与问题无关的段落；最后，通过计算剩余段落与问题的相关度，仅将少数高分段落返回，作为答案抽取模块的输入。与整篇文档检索的想法相比，段落检索具有以下优势：

1) 减少了所需要处理的文档片段的长度和规模，能够快速有效地进行语法分析；
2) 提高了文档相关性确定的准确性，避免了组成查询的关键词在文档中散布而造成的检索错误，也避免了文档长度对相关度的影响；
3) 提高了相关内容呈现的有效性：在特殊情况下，用户提出的问题需要大段文字的描述才能够解决，此时也可以直接将该模块所返回的段落作为答案。

[Tellex 2003]细致地考察了八种最好的段落检索算法。实验结果表明，基于密度的算法可以获得相对较好的效果。所谓基于密度的算法，就是通过考虑关键词在段落中出现次数和接近程度来决定这个段落的相关性。目前表现比较好的段落检索算法有三个：MultiText 算法[Clarke 2000]、IBM 算法[Ittycheriah 2000]和 SiteQ 算法[Lee 2001]。以下方便对这三个段落检索算法进行介绍。

MultiText 算法

MultiText 算法将问题与文档都视为一个词序列。记问题的词序列为 Q，文档的词序列为 D。

考虑 Q 中出现的词的一个子集 T。若 D 的一个连续子序列 S 中的词都是 T 中的词，则称 S **满足** T。若 S 满足 T 且 S 的任何连续子序列 S'都不满足 T，则称 S 是 T 的一个**覆盖**。MultiText 算法的基本思想是计算 Q 的所有子集在所有段落上的覆盖，并据此检索出与 Q 相关的段落。

对于一个词 t，MultiText 算法使用一种类似 IDF 思想的方式来计算其权重：

$$w_t = \log(N/f_t)$$

其中，f_t指 t 在所有段落中出现的总次数，N 则指整个语料库的总词数。

随后，一个词集 T 的权重被定义为其中的所有词的权重之和：

$$W(T) = \sum_{t \in T} w_t$$

如果词序列$S = d_u d_{u+1} \dots d_v$是 T 的一个覆盖，则 S 和 T 之间的分值可以被定义为：

$$C(T,S) = W(T) - |T|\log(v-u+1)$$

依据 C(T,S)对所有可能的(T,S)对进行排序，并取出排在前列的若干个 S。对于其中的每一个连续子序列 S，MultiText 算法选取其中间位置，并在原文档中向前向后各延伸 100 个单词，截取出一个总长度为 200 的段落，加入到段落检索结果列表中。

IBM 算法

IBM 算法从多方面特征来计算问题与文本段落的相关性。这些特征包括：

- 词匹配特征：即问题与段落共现的词的 IDF 值之和；（ + ）
- 领域匹配特征：基于 WordNet 同义词典，若段落中有些词是问题中的某些词的同义词，则将这些词的 IDF 值进行加和；（ + ）
- 词失配特征：即出现在问题中但未出现在答案中的词的 IDF 值之和；（ - ）

- 散布特征：将段落视为一个词序列，计算该词序列中最先匹配到问题中的词的位置与最后匹配到问题中的词的位置之间有多少个词无法匹配到问题中的词；（ - ）
- 词聚集特征：将段落视为一个词序列，统计其中有多少对相邻的词能够匹配到问题中的词。（ + ）

其中，带（ + ）标记的特征意为该特征的值越大，则 IBM 算法越趋于给这个段落一个较高的得分；反之，（ - ）标记意味着该特征的值越小，则 IBM 算法越趋于给这个段落一个较高的得分。

对这五个特征进行带符号的求和，即得到了问题与文档之间的相关性分值。

SiteQ 算法

SiteQ 算法将段落细分为句子。一个段落的得分被定义为该段落中所有句子的得分的总和。句子的得分是基于其中包含的问题中的词的稠密程度计算的。更具体地，一个句子的得分的计算公式为：

$$Score_{sent} = Score_1 + Score_2$$

$$Score_1 = \sum_i wgt(qw_{\mathrm{i}})$$

$$Score_2 = \frac{\sum_{\mathrm{j}=1}^{\mathrm{k}-1} \frac{wgt(dw_{\mathrm{j}}) + wgt(dw_{\mathrm{j}+1})}{\alpha \times dist(j, j+1)^2}}{k-1} \times matched\_cnt$$

其中，$wgt(qw_{\mathrm{i}})$意为问题中的第 i 个词的权重。在[Lee 2001]中，该权重取决于词性、WordNet 等多方面因素，但在后续应用中，一般基于 IDF 值来设置此权重。即：如果该词在句子中出现，则其权重为 IDF 值；否则，权重为 0。此外，$wgt(dw_{\mathrm{j}})$指的是句子中与问题中的词能够匹配上的第 j 个词的权重；dist(j , j+1)指的是匹配到的第 j+1 个词与第 j 个词之间的距离；matched_cnt 指问题中有多少个词能够被该句子所匹配到；α是一个超参数，用于调节词的匹配度（$Score_1$）与稠密度（$Score_2$）之间的权重。

综合来看，MultiText 算法、IBM 算法、SiteQ 算法这三个算法虽然在设计和实现细节上有很大的差异，但是都使用了 IDF 值的总和，且都考虑了邻近关键词之间距离的因素。然而，基于密度的算法只考虑了独立的关键词及其位置信息，没有

考虑关键词在问句中的先后顺序，也没有考虑语法和语义信息。针对这一问题，[Cui 2005]提出了一种基于模糊依赖关系匹配的算法。这种算法把问题和答案都解析称为语法树，并且从中得到词与词之间的依赖关系，然后通过依赖关系匹配的程度来进行排序。实验结果表明，这种方法的检索效果比基于密度的算法好。

传统检索方法一般只需要处理关键词，而问答系统需要处理更多的语法、语义信息。因此，部分问答系统也把语法、语义等信息添加到索引中，丰富了传统的索引，以提高检索效果。[Radev 2000]把一些关键词或者词组的属性放入索引，这样构造的查询包含关键词和答案的属性要求。例如，对于一个问时间的问题，构造的查询包含关键词和时间的属性，返回的段落中要求包含时间。[Bilotti 2007]把问题变成一个结构化的查询，表达查询词和段落中应该包含的某些词的属性。为了解决问题关键词的顺序问题，[Katz 2003]把句子解析为<主，谓，宾>三元组的形式，然后加入索引。另外，[Chu 2006]还索引了句子中词和词组的语义关系。

2.3.3. 答案处理模块

段落索引模块输出的有序的段落集合被输入到答案处理模块中，以提炼答案。在答案处理模块中，系统需要对段落中的答案进行识别、抽取、验证和表达等一系列的处理。首先，利用相应的语法及语义分析规则，对每个输入的段落进行语义分析；而后，根据这些语义信息，以及问题处理模块的到的关于答案的信息推断出答案的准确位置，并将其抽取出来作为候选答案。为了检验答案的正确性，系统还将对候选答案进行评估以及排序，并将最可能正确的答案返回用户。我们将答案处理模块视为两个步骤的组合：

1) 候选答案集合的生成
2) 答案提取

2.3.3.1. 候选答案集合的生成

通过问题分析，已经获得问题的类型目前，问答系统的答案处理模块一般都是面向事实类型的问题的。对于非事实类型的问题，可以直接将段落检索的结果返回给用户作为答案。大多数事实类型的问题对应的答案比较短，可能是实体名，如人名、地点等；可能是抽象名词，如人类、学科、树木、植物等；也可能是数字，如距离、速度等。对于这类问题，可以通过找到相应类型的词、词组或者片段来回答。目前，自然语言处理领域命名实体的识别已经能够达到非常好的效果，如隐马尔可夫模型（hidden Markov model, HMM）活着条件随机场模型（conditional

random field, CRF）。对于实体名词词表，除了利用 WordNet 等字典之外，还可以采用一些概念性名词和具体名词作为训练的种子，用 bootstrap 方法从文档集活着互联网中找到这种连接概念性名词和具体名词的模式，再根据这种模式提取更多的具体名词，多次迭代可以发现更多的<概念名词，具体名词>词对和相应的模式[Mann 2002]。另一种简单的方法是直接利用 Web 资源（如 Wikipedia）中具体名词列表。对于抽象名词，通常是构建一个名词列表，若片段中含有这个列表中的词，就作为答案返回。对于数字度量，可以通过正则表达式来获取。例如，距离的一个模式是数字跟上距离单位，如 5 米。通过在文本中匹配相应问题类型的短语，就构成了候选答案集合。

2.3.3.2. 答案提取

与问题处理模块相类似，答案处理模块会使用相应的信息抽取策略。这些方法从语义描述能力的不同来分，包括：基于表层特征的答案提取，基于语法结构匹配的答案提取，基于模版的答案提取，基于统计模型的答案提取，等等。下面来具体介绍分析一下这些方法。

基于表层特征的答案提取 常用的表层特征是答案周围段落的一些特征，如段落和问题关键词的相关程度、问题关键词之间的距离、问题关键词和候选答案的距离等。一般来说，段落相关程度越高，问题关键词之间以及问题关键词和候选答案之间的距离越接近，则该候选答案越可能是问题的答案。另一个常用的特征是该候选答案出现的次数。对于一个比较大的文档集，一个问题的答案可能反复出现，出现的次数越多，则它越可能是正确答案[Lin 2003]。

基于语法结构匹配的答案提取 表层特征没有考虑语法、语义的因素，容易出错，特别是词相同，但词序不同的情况。[Light 2001]指出，这种基于实体识别和表层特征的方法的性能上限是 70%。味了客服基于表层特征抽取答案的缺陷，基于语法结构匹配的答案提取方法被提出。其中一种方法是基于依存关系分析，将问题和候选段落转换成<主，谓，宾>三元组，删除句子中的修饰成分，就可以从文本三元组中获得答案而不产生混淆[Katz 2003]。另一种改进方法是建立问题到答案的逻辑表示[Moldovan 2002]。逻辑表示是介于句法解析表示和深层语义表示之间的一种表示形式，它可以通过对解析获得的句法树进行一些规则计算获得、表达主语、宾语、前置词、复杂的名词性短语、附属的形容词或副词之间的关系。

基于模版的答案提取 基于模版的答案提取方法通常被应用于将问题模版作为问题特征的问答系统中。在问题被映射到问题模版之后，可以直接获取该问题模版对

应的答案模版，最后再利用答案模版从候选段落中抽取答案。味了能够保证答案的正确性，每个答案模版只能和一个问题模版相对应。该方法虽然不能分析候选段落的语义信息，但是通过这种模版间的一一对应关系，保证了所抽取到的信息的准确性。例如，TREC-10 中的系统 INSIGHT[Soubbotin 2001]通过人工分析的方法，将定义类问题的解答方式分为六种，并构建了六个答案模版，以从信息来源中抽取答案。利用这些答案模版，INSIGHT 在该届 TREC 的比赛中赢得了冠军。淡然，模版的粒度和涵盖程度，对系统得到正确答案的成功率和答案质量都有决定性影响。于是，这种方法也带来了繁重的模版制定工作。为了解决这一问题，可以通过在训练数据上自动学习以得到模版。例如，[Cui 2004]提出了一种软模式的方法，来处理定义类问题答案的抽取。

基于统计模型的答案提取 在上述答案提取方法之外，研究人员还尝试用统计模型对答案提取进行建模。目前两个有代表性的建模方法为：一种是噪音信道模型，该模型把问题看成目标信息，把答案看成源信息，假设源信息需要通过一个包含噪音的信道，则转换概率可以通过一组训练数据（问题答案对集合）训练获得[Echihabi 2003]；另一种是用无向图模型对答案提取过程进行建模 [Ko 2007]。

2.4. IBM Watson

前文介绍了问答系统的研究背景、基本概念与基本结构。本节将具体介绍 IBM Watson，近年来最负盛名的开放领域问答系统。

Watson 是由 IBM 公司研发的一个开放领域问答系统[Ferrucci 2012]。该系统在 2011 年 2 月的美国问答节目《Jeopardy!》上一战成名。在这次节目中，Watson 战胜了这一节目的两位冠军选手，这被和 1996 年同样来自 IBM 的“深蓝”战胜国际象棋大师卡斯帕罗夫相提并论，被认为是人工智能历史上的一个里程碑。从技术角度来看，2011 年参加“Jeopardy！”电视问答挑战赛时，Watson 做了一件事——用自然语言进行深度问答。近年来，IBM 公司已经对 Watson 在外延上进行了极大的扩展。如今，问答只是 Watson 具备的众多能力之一，截至 2015 年 10 月，Watson 已经拥有包括问答在内的 28 项能力，并形成了一系列数字服务或 API。在本文中，我们所说的 Watson 特指其中的自然语言问答功能。

我们可以通过 2000 次 Jeopardy 竞赛的数据来分析人类在回答这些问题时的表现。平均上看，一个人类选手会对 40%到 50%的问题给出答案，答案的正确率在 85%到 95%之间。为了击败人类选手，Watson 的目标是：以至少 85%的正确率回答 70%以上的问题，且回答每个问题的反应时间只有数秒。

在 Watson 的处理流程中，共有数百个具体算法合作执行，以获得问题的答案。这些算法按照如图 2-7 所示的整体架构[Ferrucci 2012]进行组织。

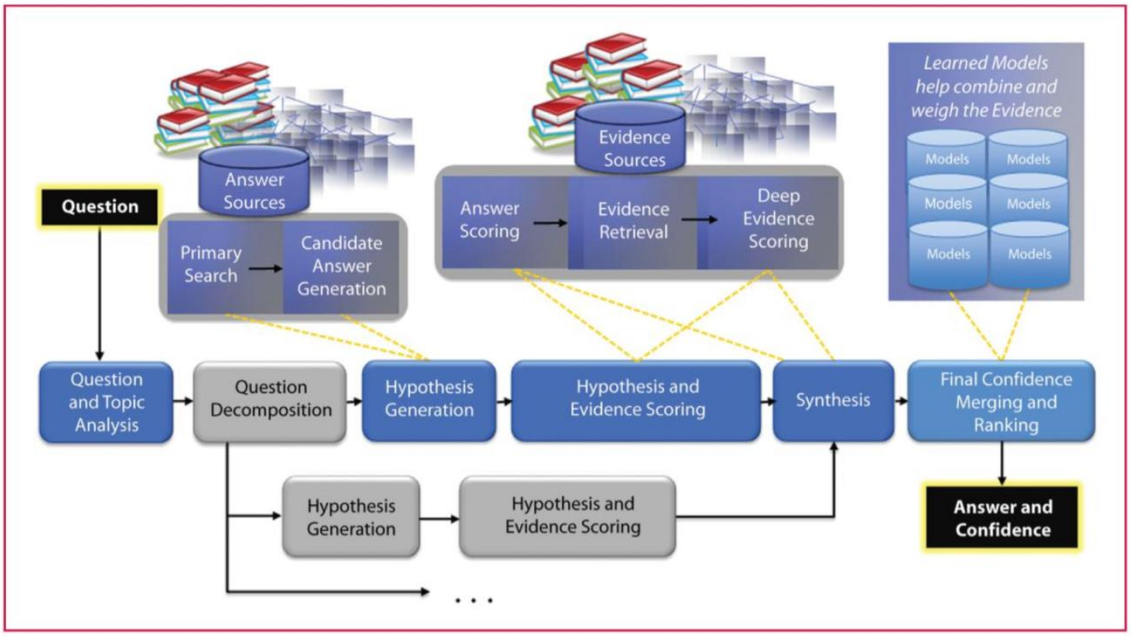

图 2-7 IBM Watson 的整体架构

Watson 中涉及到大量的技术细节，本文并不一一对它们进行列举，而是从下述四个方面，概要性地介绍 Watson 系统的主干：

1) 信息来源；
2) 问题分析；
3) 假设生成（即生成候选答案集合）；
4) 最终答案生成。

2.4.1. 信息来源

在信息来源方面，Watson 主要做了三项处理：信息来源获取、信息来源转换、信息来源扩展[Chu 2012]。

信息来源获取，指的是获取能够对问题与答案有很好的覆盖度的数据集。Watson 所采用的初始数据集为 Wikipedia 上的半结构化网页数据，一共包含超过 3,500,000 个页面，合计 13GB。对于 Jeopardy 竞赛的历史问答集，这一数据集大约能够覆盖其中的 77.1%的问题。在此基础上，Watson 采用一种迭代式的失配分析方法来获取更多的信息来源，以覆盖更多的问题。在这一方法中，首先需要找出最经常发生失配的问题类型。这些问题类型包括：逆定义类问题（给出定义描述，要

求答案给出相应的实体），引用类问题（可以是对引用中缺失的词/句子进行补充完整，也可以是要求给出引用内容的来源），圣经的细节，书籍和电影的情节等等。针对这些问题类型，Watson 研究组搜集与之对应的数据源来覆盖它们，包括：维基词典、维基语录、各种不同版本的圣经、古滕堡项目中搜集的各种热门书籍，等等。通过这样的方式，Watson 系统性地对主要失配的问题类型进行了信息来源补充。然而，还有一些特殊的问题并没有被覆盖到。为了覆盖这些问题，Watson 课题组又加入了一些额外的开放领域数据集合，例如：辞典、新闻报道等等。

信息来源转换，指的是对信息来源中的原始数据进行一定的预处理，使其更符合后续的句子处理与答案抽取等工作的要求。图 2-8 和图 2-9 给出了一个简单的示例。图 2-8 是一段定义类型的语料，其中的句子大多没有主语（如“A set of structured activities.”），不利于后续处理。Watson 将利用文档的标题，对这些句子进行重构。例如，对于前述的句子，转换后的句子是：“A program is a set of structured activities”。

- Program
  - A set of structured activities.
  - A leaflet listing information about a play, a game, or other activities.
  - A performance of a show or other broadcasts on radio or television.
  - . . .

图 2-8 定义类型的语料的一个示例

- Program
  - *A program is* a set of structured activities.
  - *A program is* a leaflet listing information about a play, a game, or other activities.
  - *A program is* a performance of a show or other broadcasts on radio or television.
  - . . .

图 2-9 对定义类型的语料的重构示例

信息来源扩展，针对的是辞典、字典等类型的信息来源。这些数据条目清晰，但这也意味着缺乏冗余。对于一个概念或实体，没有更多的数据从不同的侧面对其进行补充描述，这样的信息量很可能是不够的，导致有一些问题无法从中找到答案。因此，需要利用海量的互联网网页数据对已有的信息来源进行扩展，增加信息的冗余程度与丰富程度，以提高对问题的覆盖率。对于原信息来源中的一个文档，首先为其生成候选的扩展文档集。候选的扩展文档集一方面包含原文档中的链接所指向的互联网页面，另一方面包含以原文档中的关键词进行互联网搜索之后得到的高相关性网页。之后，对候选的扩展文档集进行段落切分与评分，选取得分较高的若干个段落，整合在一起作为原文档的扩展语料。图 2-10 给出了这一流程的示例。

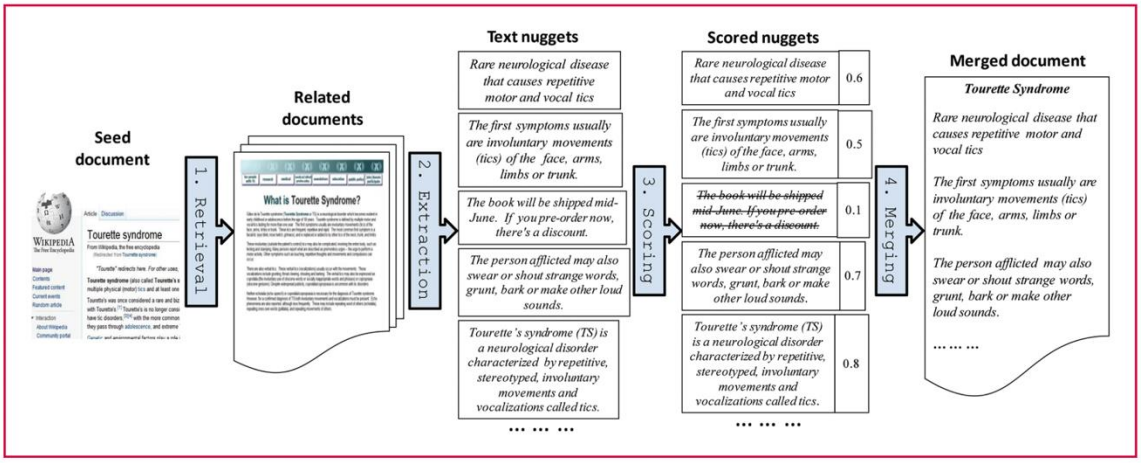

图 2-10 信息来源扩展的流程示例

经过这三个步骤，Watson 系统对 Jeopardy 的问题覆盖率从 77.1%提升到了 87.5%。

2.4.2. 问题分析

通过问题分析，Watson 从无结构的自然语言问题中识别出了句法与语义层面上的元素，并将它们编码为结构化的信息，从而在 Watson 的后续工作流程中进行使用。在 Watson 中，几乎所有组件都在某种程度上需要用到问题分析产生的结果。在 Watson 中，有许多检测规则和分类器被用于从问题中识别出各种特定的元素。不同的元素在后续处理中被不同的组件所使用。[Lally 2012]对 Watson 中的问题分析进行了概述。

在 Watson 的问题分析中，最重要的元素包括：问题焦点（focus）、词法答案类型（lexical answer types, LATs）、问题分类（Question Classification），以及问题部件（Question Section, QSection）。以下面这个 Jeopardy 问题为例：

POETS & POETRY: He was a bank clerk in the Yukon before he published "Songs of a Sourdough" in 1907.

问题的焦点指的是问题中指代答案的部分，在这个问题中，问题的焦点是"he"。在后续处理中，问题的焦点被用于与检索到的段落中的词进行对齐，从而抽取出答案。

LATs 指的是问题中能够体现答案实体应该是什么类型的词的集合。在这个问题中，LATs 为："he", "clerk"和"poet"。在后续处理中，LATs 用于验证一个候选答案是否能用于回答这个问题。

问题分类指的是问题是否属于某个或多个较为宽泛的类型。本问题属于一个事实型的问题。对于 Jeopardy 的问题，还有其它的问题分类，包括：定义型问题、多选型问题、填空型问题、缩写型问题，等等。在后续处理中，问题分类的不同将导致 Watson 从不同的答案提取技术中选取最适合该问题类型的答案提取技术来进行处理。

问题部件指的是问题中需要特殊处理的片段，一般是一些约束。例如："4-letter"或者"3-word"。在后续处理中，问题部件用于将问题分解为多个子问题的组合。

Watson 实现问题分析的基本方法为：

首先解析问题句中的依存关系。Watson 使用英文的槽文法（English Slot Grammar, ESG）解析器来实现依存关系的解析。

随后，基于这一解析结果，Watson 的研究组用 Prolog 逻辑编程语言构造模版，从而可以通过模版规则匹配提取出问题中的各种元素。

2.4.3. 假设生成

假设生成，指为一个问题生成候选答案集合。相比于最终给出的答案，假设生成所得到的候选答案集合并不十分看重候选答案的准确率，其重点在于尽量保证正确答案的召回率。当然，候选答案集合的规模也不宜过大，否则后续对各个候选答案进行评分的过程的代价将会过大，且容易被噪音所干扰。Watson 的假设生成的概述可见[Chu 2012 a]。

在 Watson 中，假设生成主要由两部分构成：搜索和候选集合生成。

在搜索步骤中，Watson 使用一种多因素的方法来获取与问题具有较高相关性的段落。相比于其它问答系统，Watson 采用的段落检索策略在两方面进行了改进：首先，对于标题是概念或实体的问答，Watson 采用了特殊的搜索策略以对标题与文本内容之间的关系加以利用；其次，Watson 利用了从问题中提取出的句法和语义层面上的关系信息。

候选集合生成步骤中包含了多重候选集合生成策略，本文取其重的两个具有代表性的策略作为示例：

基于文档标题的候选集合生成策略 若搜索到的某个文档的标题是概念或者实体，则这个概念或实体就可以被作为一个候选答案。有的时候，文档标题具有消歧信息。例如，对于来自 Wikipedia 的文档，一个文档的标题是“Naomi(Bible)”，这说明该文档所述的 Naomi 是一个人；另一个文档的标题是“Naomi(band)”，这说明该文档所述的 Naomi 是一个乐队。Watson 对这种情况进行了特殊处理，通过检查文档标题中的消歧信息是否与问题分析结果中的 LATs 相匹配来决定该概念或实体是否应该被加入到候选答案集合中。

基于命名实体识别的候选集合生成策略 从搜索到的文档中识别出若干命名实体，并将这些命名实体关联到 Wikipedia 中的条目上去。依照 Wikipedia 条目内的信息，可以知道该命名实体的类型。之后，比对该命名实体的类型与问题分析结果中的 LATs 是否匹配，来决定该命名实体是否应该被加入到候选答案集合中。由于 Watson 处理的最主要的文档是 Wikipedia 文档，而 Wikipedia 文档中的命名实体往往都带有超链接标签，因此这一策略中的命名实体识别主要指的是基于对超链接标签的解析找到对应的 Wikipedia 文档。

2.4.4. 最终答案生成

在得到候选答案集合之后，Watson 为每个候选答案生成证据，并根据证据是支持这一候选答案还是拒绝这一候选答案对证据进行打分。随后，这些证据的得分被综合起来，用于对候选答案进行排序，并给出答案的可信度。在 Watson 中，涉及到的证据多达上千种，人工设定这些证据的分值应该被如何综合起来是不现实的。针对这一问题，Watson 使用机器学习算法，从已知的问题答案对中训练分值综合模型。在这一框架中，用于学习和预测的实例指的是一个问题-答案对，特征指的是对一个问题-答案对从不同场面上打出的一系列分值。为了构造机器学习的训练集，Watson 从 Jeopardy 节目中的约 25,000 个问题出发，采集了 5,700,000 个问

题-答案对样本。对于每个样本，Watson 共计算了 550 个特征。[Gondek 2012]介绍了这一基于机器学习的答案综合与排序框架。

对于生成最终答案这一任务，Watson 面临的主要挑战包括：

- 一些候选答案可能是等同的，或者它们紧密相关联。在这种情况下，一个答案的证据可能也能够用来影响另一个答案的可信度。
- 对于不同的问题和问题类型，各种不同的特征的影响力可能会有非常大的区别。同时，对于某些类型的问题，训练数据可能非常有限。
- 一些特征在进行答案排序的不同阶段可能会有不一样的影响力。例如，在进行初期排序时，更看重能够提高答案召回率的特征；而在后期的精化排序时，更看重能够提高答案准确率的特征。
- 不同的特征可能具有很大的异构性。它们是采用不同的算法计算出来的，且这些算法是独立开发的。因此，很多特征并没有归一化。同时，在问题数据集中，很多特征都是稀疏的。
- 如果将最终答案生成问题看成是候选答案集合上的一个二分类问题，那么这个二分类问题是很不平衡的。候选答案集合中可能包含很多候选答案，但只有一个答案能成为最终的正确答案。

为了解决这些挑战，Watson 用到的技术包括：

答案融合技术 该技术能够识别出哪些答案是相同的。例如，“John F. Kennedy”，“J.F.K”和“Kennedy”指的都是美国总统肯尼迪。答案融合技术由多个相互独立的构件组成，代表性的构件有包括：一个基于英文词形的答案融合构件；一个基于模式的答案融合构件；一个基于结构化知识库消歧的答案融合构件；等等。

特征选取技术 从词法、句法、语义等各个层面上，Watson 共选取了 550 个特征。然而，为了避免发生过拟合，不能直接对这些特征进行机器学习，而是应该对这些特征进行预处理，对于每一种类型的问题，选取出合适的一个特征子集来进行机器学习。Watson 使用了一种自动化方法来选取特征子集。首先，使用最好优先一致性子集搜索技术（best first consistency subset technique, [Liu 1996]）生成一个初始的特征子集；随后，基于正确相关性（correlation with correctness），为这个特征子集补充更多特征。

特征融合技术 一个答案可能会有多个来源，比如，有多个被检索到的段落都能抽取出这个答案，或者是答案融合技术将来自不同段落的多个答案融合在了一

起。而问题-答案对的很多特征都需要通过答案的来源来计算，这就导致了根据答案的不同来源可能会算出多个不同的特征向量。因此，需要使用特征融合技术，将多个特征向量融合为单一的特征向量。Watson 在进行特征融合时采用的基本思想是：特征值越高的来源，其可信度越高。因此，在特征融合公式中，不同的特征值首先从高到低进行排序，而后按照排序进行指数式的权重衰减，最后通过求和生成单一的特征值。具体公式如下所示：

$$\mathrm{decay}(p_0, \ldots, p_k) = \sum_{i=0}^{K} \frac{p_i}{2^i}$$

其中，p 代表同一个特征在不同的答案来源上的不同的取值，并按照从高到低的顺序进行排序。

特征标准化技术 特征值在候选答案中的分布是不同的，Watson 认为其分布会影响特征的重要性。例如，如果有一个特征，它的值在某个候选答案上很高，而在其它候选答案上很低，则这个特征很可能是一个较为重要的特征；反之，如果一个特征在所有候选答案上的值都很高，那么这个特征就不是一个重要的特征。基于这一思想，Watson 对特征值进行了标准化。特征x_{ij}将按照下面的公式被标准化为x_{ij}^{std}：

$$x_{ij}^{std} = \frac{x_{ij} - \mu_j}{\sigma_j}, \quad \mu_j = \frac{1}{|Q|}\sum_{k=1}^{|Q|} x_{kj},$$

$$\sigma_j = \sqrt{\frac{1}{|Q|}\sum_{k=1}^{|Q|}(x_{kj} - \mu_j)^2}$$

基于分类器的答案排序技术 Watson 使用机器学习中的有监督分类器将不同的特征综合为一个数值，从而能够对不同的候选答案进行排序。更具体地，Watson 使用的分类算法为正则化的逻辑回归分类器。该分类器依照如下公式为每个候选答案生成一个 0 到 1 之间的分值：

$$f(x) = \frac{1}{1 + e^{-\beta_0 - \sum_{m=1}^{M} \beta_m x_m}}$$

其中，候选答案 x 共有 M 个特征，β则为一系列超参数。基于这一公式，Watson 在训练时会使正确的答案的 f 值尽可能大，错误答案的 f 值尽可能小，从而训练生成一个分类器。对于一个新问题，该分类器使用 f 值从高到低地对候选答案进行排序。

迁移学习技术 Watson 对于不同类型的问题分别进行机器学习。这种方法对于很多类型的问题，例如定义类问题和翻译类问题，是很有效的。然而，有些类型的问题的训练数据较少，则效果就会很受限。因此，Watson 采用迁移学习技术，使得这些训练数据不足的问题类型可以从其它问题类型中借鉴模型参数。

迭代精化技术 Watson 的机器学习不是一次完成的，而是反复执行、持续迭代精化的。每次迭代都将分值明显偏低的候选答案筛除，然后进行下一次迭代，最终生成精化的排序结果。

证据扩散技术 证据扩散，指的是候选答案集合中的证据可以作为背景知识来互相影响。例如，对于下面这个 Jeopardy 问题：WORLD TRAVEL: If you want to visit this country, you can fly into Sunan International Airport or … or not visit this country. (Correct answer: "North Korea")。问题中提到的"Sunan International Airport"位于平壤。对于绝大部分提到这个实体的答案来源段落，它们都会提到平壤。因此，我们可以将其与平壤的关系视为背景知识，进而依靠平壤与朝鲜的语义关联关系而找到正确答案：朝鲜。

2.5. 参考文献

[Agichtein 2004] Agichtein, E., Lawrence, S., & Gravano, L. (2004). Learning to find answers to questions on the Web. *ACM Transactions on Internet Technology (TOIT)*, *4*(2), 129-162.

[Bilotti 2007] Bilotti, M. W., Ogilvie, P., Callan, J., & Nyberg, E. (2007, July). Structured retrieval for question answering. In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 351-358). ACM.

[Bouma 2002] Bouma, G., & Kloosterman, G. (2002). Querying Dependency Treebanks in XML. In *LREC*.

[Brin 1998] Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. Computer networks and ISDN systems, 30(1), 107-117.

[Carreras 2005] Carreras, X., & Màrquez, L. (2005, June). Introduction to the CoNLL-2005 shared task: Semantic role labeling. In Proceedings of the Ninth Conference on Computational Natural Language Learning (pp. 152-164). Association for Computational Linguistics.

[Chu 2006] Chu-Carroll, J., Prager, J., Czuba, K., Ferrucci, D., & Duboue, P. (2006, August). Semantic search via XML fragments: a high-precision approach to IR. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 445-452). ACM.

[Chu 2012] Chu-Carroll, J., Fan, J., Schlaefer, N., & Zadrozny, W. (2012). Textual resource acquisition and engineering. IBM Journal of Research and Development, 56(3.4), 4-1.

[Chu 2012 a] Chu-Carroll, J., Fan, J., Boguraev, B. K., Carmel, D., Sheinwald, D., & Welty, C. (2012). Finding needles in the haystack: Search and candidate generation. IBM Journal of Research and Development, 56(3.4), 6-1.

[Clarke 2000] Clarke, C. L., Cormack, G. V., Kisman, D. I., & Lynam, T. R. (2000, November). Question Answering by Passage Selection (MultiText Experiments for TREC-9). In TREC.

[Collins 2004] Collins-Thompson, K., Callan, J., Terra, E., & Clarke, C. L. (2004, July). The effect of document retrieval quality on factoid question answering performance. In Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 574-575). ACM.

[Cui 2004] Cui, H., Kan, M. Y., & Chua, T. S. (2004, May). Unsupervised learning of soft patterns for generating definitions from online news. In Proceedings of the 13th international conference on World Wide Web (pp. 90-99). ACM.

[Cui 2005] Cui, H., Sun, R., Li, K., Kan, M. Y., & Chua, T. S. (2005, August). Question answering passage retrieval using dependency relations. In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 400-407). ACM.

[Dang 2007] Dang, H. T., Kelly, D., & Lin, J. J. (2007, November). Overview of the TREC 2007 Question Answering Track. In Trec (Vol. 7, p. 63).

[Dave 2003] Dave, K., Lawrence, S., & Pennock, D. M. (2003, May). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In Proceedings of the 12th international conference on World Wide Web (pp. 519-528). ACM.

[Echihabi 2003] Echihabi, A., & Marcu, D. (2003, July). A noisy-channel approach to question answering. In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1 (pp. 16-23). Association for Computational Linguistics.

[Ferrucci 2012] Ferrucci, D. A. (2012). Introduction to “this is watson”. IBM Journal of Research and Development, 56(3.4), 1-1.

[Gondek 2012] Gondek, D. C., Lally, A., Kalyanpur, A., Murdock, J. W., Dubou é, P. A., Zhang, L., ... & Welty, C. (2012). A framework for merging and ranking of answers in DeepQA. IBM Journal of Research and Development, 56(3.4), 14-1.

[Green 1961] Green Jr, B. F., Wolf, A. K., Chomsky, C., & Laughery, K. (1961, May). Baseball: an automatic question-answerer. In Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference (pp. 219-224). ACM.

[Hacioglu 2004] Hacioglu, K., Pradhan, S., Ward, W. H., Martin, J. H., & Jurafsky, D. (2004, May). Semantic Role Labeling by Tagging Syntactic Chunks. In CoNLL (pp. 110-113).

[Hao 2008] Hao, T., Hu, D., Wenyin, L., & Zeng, Q. (2008). Semantic patterns for user-interactive question answering. Concurrency and Computation: Practice and Experience, 20(7), 783-799.

[Harabagiu 2000] Harabagiu, S. M., Moldovan, D. I., Paşca, M., Mihalcea, R., Surdeanu, M., Bunescu, R., ... & Morărescu, P. (2000). Falcon: Boosting knowledge for answer engines.

[Hu 2007] Hu, D., Li, H., Hao, T., Chen, E., & Wenyin, L. (2007, June). Heuristic learning of rules for information extraction from web documents. In Proceedings of

the 2nd international conference on Scalable information systems (p. 61). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[Ittycheriah 2000] Ittycheriah, A., Franz, M., Zhu, W. J., Ratnaparkhi, A., & Mammone, R. J. (2000, November). IBM's Statistical Question Answering System. In TREC.

[Katz 2003] Katz, B., & Lin, J. (2003, April). Selectively using relations to improve precision in question answering. In Proceedings of the workshop on Natural Language Processing for Question Answering (EACL 2003) (pp. 43-50).

[Ko 2007] Ko, J., Nyberg, E., & Si, L. (2007, July). A probabilistic graphical model for joint answer ranking in question answering. In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 343-350). ACM.

[Kolomiyets 2011] Kolomiyets, O., & Moens, M. F. (2011). A survey on question answering technology from an information retrieval perspective. Information Sciences, 181(24), 5412-5434.

[Lally 2012] Lally, A., Prager, J. M., Mccord, M. C., Boguraev, B. K., Patwardhan, S., & Fan, J., et al. (2012). Question analysis: how watson reads a clue. Ibm Journal of Research & Development, 56(56), 2:1-2:14.

[Lampert 2004] Lampert, A. (2004). A quick introduction to question answering. Dated December.

[Lee 2001] Lee, G. G., Seo, J., Lee, S., Jung, H., Cho, B. H., Lee, C., ... & Kim, H. (2001, November). SiteQ: Engineering High Performance QA System Using Lexico-Semantic Pattern Matching and Shallow NLP. In TREC.

[Li 2002] Li, X., & Roth, D. (2002, August). Learning question classifiers. In Proceedings of the 19th international conference on Computational linguistics-Volume 1 (pp. 1-7). Association for Computational Linguistics.

[Light 2001] Light, M., Mann, G. S., Riloff, E., & Breck, E. (2001). Analyses for elucidating current question answering technology. Natural Language Engineering, 7(04), 325-342.

[Lin 2003] Lin, J., & Katz, B. (2003, November). Question answering from the web using knowledge annotation and knowledge mining techniques. In Proceedings of the twelfth international conference on Information and knowledge management (pp. 116-123). ACM.

[Liu 1996] Liu, H., & Setiono, R. (1996, July). A probabilistic approach to feature selection-a filter solution. In ICML (Vol. 96, pp. 319-327).

[Liu 2002] Liu, X., & Croft, W. B. (2002, November). Passage retrieval based on language models. In Proceedings of the eleventh international conference on Information and knowledge management (pp. 375-382). ACM.

[Liu 2007] Xiaoli, L., Hu, D., Feng, M., & Wenyin, L. (2007, October). Semantic Pattern based Dependency Matching for Exact Answer Retrieval. In Semantics, Knowledge and Grid, Third International Conference on (pp. 262-265). IEEE.

[Mann 2002] Mann, G. S. (2002, September). Fine-grained proper noun ontologies for question answering. In Proceedings of the 2002 workshop on Building and using semantic networks-Volume 11 (pp. 1-7). Association for Computational Linguistics.

[Moens 2006] Moens, M. F. (2006). Information extraction: algorithms and prospects in a retrieval context (Vol. 21). Springer Science & Business Media.

[Moldovan 1999] Moldovan, D. I., Harabagiu, S. M., Paşca, M., Mihalcea, R., Goodrum, R. A., G îrju, C. R., & Rus, V. (1999). Lasso: A tool for surfing the answer net.

[Moldovan 2000] Moldovan, D., Harabagiu, S., Pasca, M., Mihalcea, R., Girju, R., Goodrum, R., & Rus, V. (2000, October). The structure and performance of an open-domain question answering system. In Proceedings of the 38th Annual Meeting on Association for Computational Linguistics (pp. 563-570). Association for Computational Linguistics.

[Moldovan 2002] Moldovan, D. I., Harabagiu, S. M., Girju, R., Morarescu, P., Lacatusu, V. F., Novischi, A., ... & Bolohan, O. (2002, November). LCC Tools for Question Answering. In TREC.

[Moldovan 2003] Moldovan, D., Paşca, M., Harabagiu, S., & Surdeanu, M. (2003). Performance issues and error analysis in an open-domain question answering system. ACM Transactions on Information Systems (TOIS), 21(2), 133-154.

[Moll á2007] Moll á, D., & Vicedo, J. L. (2007). Question answering in restricted domains: An overview. Computational Linguistics, 33(1), 41-61.

[Radev 2000] Radev, D. R., Prager, J., & Samn, V. (2000, April). Ranking suspected answers to natural language questions using predictive annotation. In Proceedings of the sixth conference on Applied natural language processing (pp. 150-157). Association for Computational Linguistics.

[Salton 1986] Salton, G., & McGill, M. J. (1986). Introduction to modern information retrieval.

[Sang 2005] Sang, E. T. K., Bouma, G., & de Rijke, M. (2005, July). Developing offline strategies for answering medical questions. In Proceedings of the AAAI-05 Workshop on Question Answering in Restricted Domains, Pittsburgh, PA, USA (pp. 41-45).

[Soubbotin 2001] Soubbotin, M. M., & Soubbotin, S. M. (2001, November). Patterns of Potential Answer Expressions as Clues to the Right Answers. In TREC.

[Tellex 2003] Tellex, S., Katz, B., Lin, J., Fernandes, A., & Marton, G. (2003, July). Quantitative evaluation of passage retrieval algorithms for question answering. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval (pp. 41-47). ACM.

[Woods 1977] Woods, W. A., & Kaplan, R. (1977). Lunar rocks in natural English: Explorations in natural language question answering. Linguistic structures processing, 5, 521-569.

[Zhang 2003] Zhang, D., & Lee, W. S. (2003, July). Question classification using support vector machines. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval (pp. 26-32). ACM.

[Zweigenbaum 2003] Zweigenbaum, P. (2003, April). Question answering in biomedicine. In Proceedings Workshop on Natural Language Processing for Question Answering, EACL (Vol. 2005, pp. 1-4).

第三章　知识表示学习技术

3.　知识表示学习技术

3.1.　背景

知识库将人类知识组织成结构化的知识系统。人们花费大量精力构建了各种结构化的知识库，如语言知识库 WordNet、通用事实知识库 Freebase 等。知识库是推动人工智能学科发展和支撑智能信息服务应用（如智能搜索、智能问答、个性化推荐等）的重要基础技术。为了改进信息服务的质量，国内外互联网公司（特别是搜索引擎公司）纷纷推出知识库产品，如谷歌知识图谱、微软 Bing Satori、百度知心以及搜狗知立方等。著名的 IBM Watson 问答系统和苹果 Siri 语音助理的背后，知识库也扮演着重要角色。如谷歌在介绍知识图谱时所说的“构成这个世界的是实体，而非字符串”。可以说，知识库的兴起拉开了智能信息检索从字符串匹配跃迁至智能理解的序幕。

知识库描述现实世界中实体（entity）间的关系（relation）。这些知识蕴藏在无（半）结构的互联网信息中，而知识库则是有结构的。因此，知识库的主要研究目标是：从无（半）结构的互联网信息中获取有结构知识，自动融合构建知识库、服务知识推理等相关应用。知识表示是知识获取与应用的基础，因此知识表示学习问题是贯穿知识库的构建与应用全过程的关键问题。

人们通常以网络的形式组织知识库中的知识，网络中每个节点代表实体（人名、地名、机构名、概念等），而每条连边则代表实体间的关系。因此，大部分知识往往可以用三元组（实体 1，关系，实体 2）来表示，对应着知识库网络中的一条连边及其连接的 2 个实体。这是知识库的通用表示方式，例如万维网联盟（W3C）发布的资源描述框架（resource description framework, RDF）技术标准，就是以三元组表示为基础的。特别是在谷歌提出知识图谱的概念之后，这种网络表示形式更是广受认可。然而，基于网络形式的知识表示面临诸多挑战性难题，主要包括如下 2 个方面：

1) 计算效率问题。基于网络的知识表示形式中，每个实体均用不同的节点表示。当利用知识库计算实体间的语义或推理关系时，往往需要人们设计专门的图算法来实现，存在可移植性差的问题。更重要的是，基于图的算法

计算复杂度高、可扩展性差，当知识库达到一定规模时，就很难较好地满足实时计算的需求。

2) 数据稀疏问题。与其它类型的大规模数据类似，大规模知识库也遵守长尾分布，在长尾部分的实体和关系上，面临严重的数据稀疏问题。例如，对于长尾部分的罕见实体，由于只有极少的知识或路径涉及它们，对这些实体的语义或推理关系的计算往往准确率极低

近年来，以深度学习[Bengio 2009]为代表的表示学习技术[Bengio 2013]异军突起，在语音识别、图像分析和自然语言处理领域获得广泛关注。表示学习旨在将研究对象的语义信息表示为稠密低维实值向量。在该低维向量空间中，2 个对象距离越近则说明其语义相似度越高。

顾名思义，知识表示学习是面向知识库中的实体和关系进行表示学习。该方向最近取得了重要进展，可以在低维空间中高效计算实体和关系的语义联系，有效解决数据稀疏问题，使知识获取、融合和推理的性能得到显著提升。由于上述优点，知识表示学习引起了广泛关注和研究兴趣，但该方向仍然面临着诸多挑战。

3.2. 基本概念

3.2.1. 表示学习

如前所述，表示学习的目标是通过机器学习将研究对象的语义信息表示为稠密低维实值向量。例如，在图像处理领域，表示学习将为每张图片生成低维实值向量表示；在自然语言处理领域，表示学习将为每个单词（或每个短语、句子、段落、文档）生成低维实值向量表示。对于结构化的知识库，表示学习将会把知识库中的实体表示为低维实值向量，实体之间的关联关系则会被表示为对应向量之间的数学约束。在表示学习产生的低维实值向量空间中，我们可以通过欧式距离或余弦距离等简单方式，计算任意 2 个对象之间的语义相似度。

实际上，在表示学习之外，有更简单的数据表示方案，即独热表示（one-hot representation）[Turian 2010]。该方案也将研究对象表示为向量，只是该向量只有某一维非零，其它维度上的值均为零。显而易见，味了将不同对象区分开，有多少个不同的对象，独热表示向量就有多长。独热表示是信息检索和搜索引擎中广泛使用的词袋模型的基础。以中文为例，假如网页中共有 W 个不同的词，词袋模型中的每个词都被表示为一个 W 维的独热表示向量。在此基础上，词袋模型将每个文档表示为一个 W 维向量，每一维表示对应的词在该文档中的重要性。

与表示学习相比，独热表示无需学习过程，简单高效，在信息检索和自然语言处理中得到广泛应用。但是独热表示的缺点也非常明显。独热表示方案假设所有对象都是相互独立的。也就是说，在独热表示空间中，所有对象的向量都是相互正交的，通过余弦距离或欧式距离计算的语义相似度均为 0。这显然是不符合实际情况的，会丢失大量有用信息。例如，“苹果”和“香蕉”虽然是 2 个不同的词，但由于它们都属于水果，因此应当具有较高的语义相似度。显然，独热表示无法有效利用这些对象间的语义相似度信息。这也是词袋模型无法有效表示短文本、容易受到数据稀疏问题影响的根本原因。

与独热表示相比，表示学习的向量维度较低，有助于提高计算效率，同时能够充分利用对象间的语义信息，从而有效缓解数据稀疏问题。由于表示学习的这些优点，最近出现了大量关于单词[Turian 2010]、短语[Mikolov 2013] [Zhao 2015]、实体[Zhao 2015 a]、句子[Hu 2014] [Le 2014] [Blunsom 2014]、文档[Le 2014]和社会网络[Perozzi 2014] [Tang 2015] [Yang 2015]等对象的表示学习研究。特别是在词表示方面，针对一词多义[Huang 2012] [Reisinger 2010] [Tian 2014]、语义组合[Zhao 2015] [Socher 2013] [Socher 2013 a] [Socher 2012]、语素或字母信息[Luong 2013] [Botha 2014] [Chen 2015]、跨语言[AP 2014] [Shi 2015]、可解释性[Luo 2015] [Murphy 2012] [Fyshe 2014]等特点提出了相应表示方案，展现出分布式表示灵活的可扩展性。

3.2.2. 知识表示学习

知识表示学习是面向知识库中实体和关系的表示学习。通过将实体或关系投影到低维向量空间，我们能够实现对实体和关系的语义信息的表示，可以高效地计算实体、关系及其之间的复杂语义关联。这对知识库的构建、推理与应用均有重要意义。

首先，我们定义知识表示学习的输入。知识表示学习的输入是一个结构化的知识库（knowledge base），或者可以称之为知识图（knowledge graph）。一个知识库 G 可以用$G = (E, R, Y)$来表示，其中：

$E = \{e_1, \dots, e_{N_E}\}$ 是实体集合；

$R = \{r_1, \dots, r_{N_R}\}$ 是实体间的关联关系类型的集合；

$Y \in \{0，1\}^{N_E \times N_E \times N_R}$ 是一个三维张量，用于表示两个实体之间是否存在某种类型的关联关系。更具体地，对于 Y 中的一个项y_{ijk}，其语义为：

$$y_{ijk} = \begin{cases} 1 & 若实体e_i与e_j之间存在类型为r_k的关系 \\ 0 & 其它情况 \end{cases}$$

当$y_{ijk} = 1$时，也可以说该知识库中存在三元组(e_i, r_k, e_j)。

对于$y_{ijk} = 0$的情况，在不同的假设下有不同的解释。例如，在封闭世界假设中，$y_{ijk} = 0$的语义为：三元组(e_i, r_k, e_j)在现实世界中不成立；而在开放世界假设中，$y_{ijk} = 0$的语义为：目前暂无法确定三元组(e_i, r_k, e_j)在现实世界中是否成立。对于目前互联网上的大规模知识图，由于其构造方法多是基于众包编辑或是自动抽取生成的，其中往往会出现大量知识缺失的情况。因此，多数情况下我们采用的是开放世界假设。

知识表示学习的基本假设是：存在一个低维、连续、稠密的向量空间，使得知识库中的每个实体可以被映射为该向量空间中的一个向量，且知识库中的每种类型的关联关系可以被映射为该向量空间中的某种形式的数学约束。如此，知识库中蕴含的语义信息就在该向量空间中得到了分布式的表示。知识表示学习的基本流程框架如下所示：

1. 假设存在一个 K 维的向量空间 ***S***，我们的目的是将知识库 G=(E, R, Y) 中蕴含的语义信息在 ***S*** 中进行分布式的表示。其中，K 是一个超参数，其取值一般为 100 到 200。
2. 对于知识库中的每一个实体e ∈ E，记其在 ***S*** 中对应的向量为 ***e***，并在 ***S*** 中随机选取一个长度不大于 1 的向量作为 ***e*** 的初始化取值。
3. 对于知识库中的每一种类型的关系r ∈ R，将其建模为数学约束$f_r(h,t): |E| \times |E| \to \mathcal{R}_*$。其含义为：对于知识库中的两个实体 h 和 t，若知识库中存在三元组 (h, r, t)，则$f_r(h,t)$的值应当趋于 0；若知识库中不存在三元组 (h, r, t)，则$f_r(h,t)$的值应当偏大。$f_r(h,t)$中会有一些模型参数，这些参数在初始情况下是随机选取的。
4. 以知识库中存在的三元组作为正样本，以知识库中不存在的三元组作为负样本，通过机器学习优化算法（如随机批量梯度下降法）进行训练，从而估计出模型参数：每个实体 e 对应的向量 ***e***，以及每种关系类型 r 所对应的函数$f_r(h,t)$中的参数。

通过这一流程，我们将知识库中的每个实体表示为了低维语义向量空间中的一个向量，且知识库中的每一个三元组(h,r,t)的信息在该空间中都体现为实体 h 和 t 对应的向量能够满足$f_r(h,t) \to 0$ 这一约束。

知识表示学习得到的分布式具有如下主要优点：

- 显著提升计算效率。知识库的三元组表示实际就是基于独热表示的。如前所分析的，在这种表示方式下，需要设计专门的图算法计算实体间的语义和推理关系，计算复杂度高、可扩展性差。而表示学习得到的分布式表示，则能够高效地实现语义相似度计算等操作，显著提升计算效率。
- 有效缓解数据稀疏。由于知识表示学习将实体投影到统一的低维空间中，使每个实体均对应一个稠密向量，从而有效缓解数据稀疏问题。这主要体现在两个方面：一方面，每个实体的向量均为稠密有值得，因此可以度量任意实体之间的语义相似程度。而基于独热表示的图算法，由于受到大规模知识库稀疏特性的影响，往往无法有效计算很多实体之间的语义相似度；另一方面，将大量实体投影到统一空间的过程，也能够将高频实体的语义信息用于帮助低频实体的语义表示，提高低频实体的语义表示的精确性。

知识表示学习得到的分布式表示有以下典型应用：

- 相似度计算。利用实体的分布式表示，我们可以快速计算实体间的语义相似度，这对于自然语言处理和信息检索的很多任务具有重要意义。
- 知识图谱补全。构建大规模知识图谱，需要不断补充实体间的关系。利用知识表示学习模型，可以自动推理、预测两个实体的关系，从而实现知识图谱中的关系的补全。
- 其他应用。知识表示学习已被广泛应用于关系抽取、自动问答、实体链指等各种任务中，展现出了巨大的应用潜力。随着深度学习在自然语言处理各项重要任务重得到广泛应用，这将为知识表示学习带来更广阔的应用空间。

3.3. 知识表示学习的主要方法

前文已经纯属了知识表示学习的基本流程框架。在这一框架中，最为关键的是如何将知识库中的关联关系类型建模为数学约束$f_r(h,t)$。针对这一问题，研究者提出了多种模型。接下来，我们介绍其中的几个代表性的模型，包括距离模型、单层神经网络模型、能量模型、双线性模型、张量神经网络模型、矩阵分解模型和翻译模型等。

3.3.1. 距离模型

结构表示（structured embedding, SE）[Bordes 2011]是较早的几个知识表示方法之一。SE 为每种关系类型 r 定义了 2 个矩阵$\boldsymbol{M}_{r,1}, \boldsymbol{M}_{r,2} \in \boldsymbol{R}^{d\times d}$，用于三元组中头实体和尾实体的投影操作。对应的约束函数为：

$$f_r(h,t) = |\boldsymbol{M}_{r,1}\boldsymbol{h} - \boldsymbol{M}_{r,2}\boldsymbol{t}|_{L1}$$

我们可以理解为，SE 将头实体向量 $\boldsymbol{h}$ 和尾实体向量 $\boldsymbol{t}$ 通过关系 r 的两个矩阵投影到 r 的对应空间中，然后在该空间中计算两个投影向量之间的距离。这个距离反映了两个实体在关系 r 下的语义相关度，它们的距离越小，说明这两个实体存在这种关系。

然而，SE 模型有一个重要缺陷：它对头、尾实体使用两个不同的矩阵进行投影，协同性较差，往往无法精确刻画两个实体与关系之间的语义联系。

3.3.2. 单层神经网络模型

单层神经网络模型（single layer model, SLM）[Socher 2013 b]尝试采用单层神经网络的非线性操作，来减轻 SE 无法协同精确刻画实体与关系的语义联系的问题。SLM 为每个三元组(h,r,t)定义了如下评分函数：

$$f_r(h,t) = \boldsymbol{u}_r^T g(\boldsymbol{M}_{r,1}\boldsymbol{l}_h + \boldsymbol{M}_{r,2}\boldsymbol{l}_t)$$

其中，$\boldsymbol{M}_{r,1}$，$\boldsymbol{M}_{r,2} \in \mathbb{R}^{d\times k}$为投影矩阵，$\boldsymbol{u}_r^T \in \mathbb{R}^k$为关系 r 的表示向量，g()是 tanh 函数。

虽然 SLM 是 SE 模型的改进版本，但是它的非线性操作仅提供了实体和关系之间比较微弱的联系。与此同时，却引入了更高的计算复杂度。

3.3.3. 能量模型

语义匹配能量模型（semantic matching energy, SME）[Bordes 2012]提出更复杂的操作，寻找实体和关系之间的语义联系。SME 为每个三元组(h,r,t)定义了 2 种评分函数，分别是线性形式：

$$f_r(h,t) = (\boldsymbol{M}_1\boldsymbol{l}_h + \boldsymbol{M}_2\boldsymbol{l}_r + \boldsymbol{b}_1)^T(\boldsymbol{M}_3\boldsymbol{l}_t + \boldsymbol{M}_4\boldsymbol{l}_r + \boldsymbol{b}_2)$$

和双线性形式：

$$f_r(h,t) = (\boldsymbol{M}_1\boldsymbol{l}_h \otimes \boldsymbol{M}_2\boldsymbol{l}_r + \boldsymbol{b}_1)^T(\boldsymbol{M}_3\boldsymbol{l}_t \otimes \boldsymbol{M}_4\boldsymbol{l}_r + \boldsymbol{b}_2)$$

其中，$\boldsymbol{M}_1, \boldsymbol{M}_2, \boldsymbol{M}_3, \boldsymbol{M}_4 \in \mathbb{R}^{d\times k}$为投影矩阵；⊗表示按位相乘（即 Hadamard 积）；$\boldsymbol{b}_1, \boldsymbol{b}_2$为偏置向量。此外，也有研究工作用三阶张量代替 SME 的双线性形式[Bordes 2014]。

3.3.4. 隐变量模型

隐变量模型（latent factor model, LFM）[Jenatton 2012] [Sutskever 2009]提出利用基于关系的双线性变换，刻画实体和关系之间的二阶联系。LFM 为每个三元组(h,r,t)定义了如下双线性评分函数：

$$f_r(h,t) = \boldsymbol{l}_h^T \boldsymbol{M}_r \boldsymbol{l}_t$$

其中，$\boldsymbol{M}_r \in \mathbb{R}^{d\times d}$是关系 r 对应的双线性变换矩阵。与以往模型相比，LFM 取得巨大突破：通过简单有效的方法刻画了实体和关系的语义联系，协同性较好，计算复杂度低。

后来的 DISTMULT 模型[Yang 2014]还探索了 LFM 的简化形式：将关系矩阵$\boldsymbol{M}_r$设置为对角阵。实验表明，这种简化不仅极大降低了模型复杂度，模型效果反而得到了显著提升。

3.3.5. 张量神经网络模型

张量神经网络模型（neural tensor network, NTN）[Socher 2013 b]的基本思想是用双线性张量取代传统神经网络中的线性变换层，在不同的维度下将头、尾实体向量联系起来。其基本思想如图 3-1 所示。

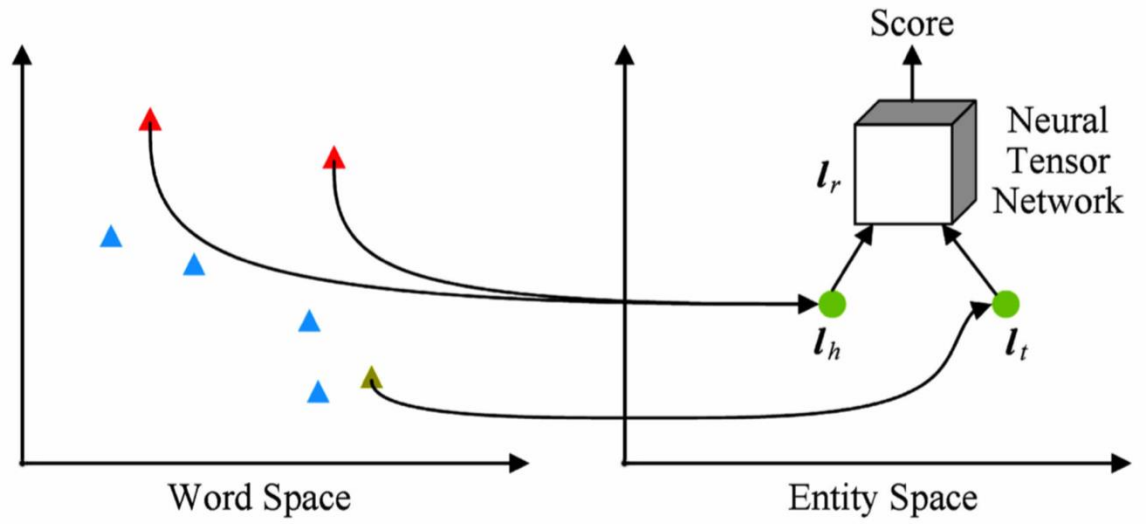

图 3-1 张量神经网络模型

NTN 为每个三元组(h,r,t)定义了如下评分函数，评价 2 个实体之间存在某个特定关系 r 的可能性：

$$f_r(h,t) = \boldsymbol{u}_r^T g(\boldsymbol{l}_h \boldsymbol{M}_r \boldsymbol{l}_t + \boldsymbol{M}_{r,1}\boldsymbol{l}_h + \boldsymbol{M}_{r,2}\boldsymbol{l}_t + \boldsymbol{b}_r)$$

其中$\boldsymbol{u}_r^T$是一个与关系相关的线性层，g()是 tanh 函数，$\boldsymbol{M}_r \in \mathbb{R}^{d\times d\times k}$是一个三阶张量，$\boldsymbol{M}_{r,1}, \boldsymbol{M}_{r,2} \in \mathbb{R}^{d\times k}$是与关系 r 有关的投影矩阵。可以看出，前述 SLM 是 NTN 的简化版本，是 NTN 将其中张量的层数设置为 0 时的特殊情况。

值得注意的是，与以往模型不同，NTN 中的实体向量是该实体中所有单词向量的平均值。这样做的好处是，实体中的单词数量远小于实体数量，可以充分重复利用单词向量构建实体表示，降低实体表示学习的稀疏性问题，增强不同实体的语义联系。

由于 NTN 引入了张量操作，虽然能够更精确地刻画实体和关系的复杂语义联系，但是计算复杂度非常高，需要大量三元组样例才能得到充分学习。实验表明，NTN 在大规模稀疏知识图谱上的效果较差[Bordes 2013]。

3.3.6. 矩阵分解模型

矩阵分解是得到低维向量表示的重要途径。因此，也有研究者提出采用矩阵分解进行知识表示学习。这方面的代表方法是 RESACL 模型[Nickel 2011] [Nickel 2012]。

在该模型中，知识库三元组构成一个大的张量 $\boldsymbol{X}$，如果三元组(h,r,t)存在，则：

$$X_{hrt} = 1$$

否则为 0。

张量分解旨在将每个三元组(h,r,t)对应的张量值X_{hrt}分解为实体和关系表示，使得X_{hrt}尽量地接近于$\boldsymbol{l}_h \boldsymbol{M}_r \boldsymbol{l}_t$。

可以看到 RESACL 的基本思想与前述 LFM 类似。不同之处在于，RESACL 会优化张量中的所有位置，包括值为 0 的位置；而 LFM 只会优化知识库中存在的三元组。

3.3.7. 翻译模型

表示学习在自然语言处理领域受到广泛关注起源于 Mikolov 等人在 2013 年研发的 word2vec 词表示学习模型和工具包[Mikolov 2013] [Mikolov 2013 a]。利用该模型，Mikolov 等人发现词向量空间存在有趣的平移不变现象。例如，他们发现：

$$\boldsymbol{C}(king) - \boldsymbol{C}(queen) \approx \boldsymbol{C}(man) - \boldsymbol{C}(woman)$$

这里$\boldsymbol{C}(w)$表示利用 word2vec 学习的到的单词 w 的词向量。也就是说，词向量能够捕捉到单词 king 和 queen 之间、man 和 woman 之间的某种相同的隐含语义关系。Mikolov 等人通过类比推理实验[Mikolov 2013] [Mikolov 2013 a]发现，这种平

移不变现象普遍存在于词汇的语义关系和句法关系中。有研究者还利用词表示的这种特性寻找词汇之间的上下位关系[Fu 2014]。

受到该现象的启发，Bordes 等人提出了 TransE 模型[Bordes 2013]，将知识库中的关系看作实体间的某种平移向量。对于每个三元组(h,r,t)，TransE 用关系 r 的向量$\boldsymbol{l}_r$作为头实体向量$\boldsymbol{l}_h$和尾实体向量$\boldsymbol{l}_t$之间的平移。我们也可以将$\boldsymbol{l}_r$看作从$\boldsymbol{l}_h$到$\boldsymbol{l}_t$的翻译，因此 TransE 也被称为翻译模型。

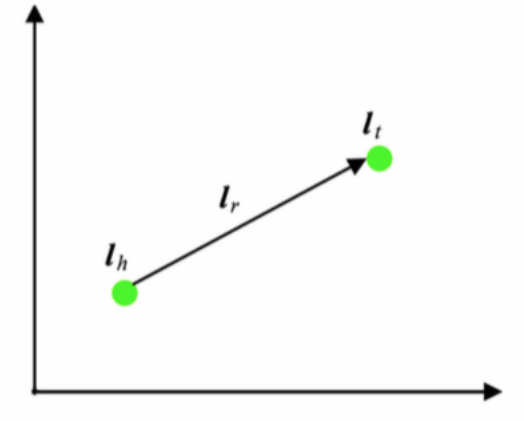

图 3-2 TransE 模型

如图 3-2 所示，对于每个三元组(h,r,t)，TransE 希望：

$$\boldsymbol{l}_h + \boldsymbol{l}_r \approx \boldsymbol{l}_t$$

TransE 模型定义了如下损失函数：

$$f_r(h,t) = |\boldsymbol{l}_h + \boldsymbol{l}_r - \boldsymbol{l}_t|_{L_1/L_2}$$

即向量$\boldsymbol{l}_h + \boldsymbol{l}_r$和$\boldsymbol{l}_t$的$L_1$或$L_2$距离。

在时机学习过程中，为了增强知识表示的区分能力，TransE 采用最大间隔方法，定义了如下优化目标函数：

$$L = \sum_{(h,r,t)\in S,(h',r',t')\in S^-} \max(0, f_r(h,t) + \gamma - f_{r'}(h',t'))$$

其中，S 是合法三元组的集合，S^-为错误三元组的集合，max(x,y)返回 x 和 y 中较大的值，γ为合法三元组得分与错误三元组得分之间的间隔距离。

错误三元组并非随机产生的，为了选取有代表性的错误三元组，TransE 将 S 中每个三元组的头实体、关系和尾实体其中之一随机替换成其他实体或关系来得到S^-，即：

$$S^- = \{(h',r,t)\} \cup \{(h,r',t)\} \cup \{(h,r,t')\}$$

在基于最大间隔的优化目标函数之外，在使用 TransE 模型时，还可以使用基于贝叶斯统计学的优化目标函数：

$$L = \sum_{(h,r,t)\in S,(h',r',t')\in S^-} \frac{1}{1+e^{-(f_r(h,t)-f_{r'}(h',t'))}}$$

与以往模型相比，TransE 模型参数较少，计算复杂度低，却能直接建立实体和关系之间的复杂语义联系。Bordes 等人在 WordNet 和 Freebase 等数据集伤进行链接预测等评测任务，实验表明 TransE 的性能较以往模型有显著提升。特别是在大规模稀疏知识图谱上，TransE 的性能尤其惊人。

由于 TransE 简单有效，自提出以来，有大量研究工作对 TransE 进行扩展和应用。可以说，TransE 已经成为知识表示学习的代表模型。在后面的章节中，我们将以 TransE 为例，介绍知识表示学习的主要挑战与解决方案。

3.3.8. 其他模型

在 TransE 提出之后，大部分知识表示学习模型是以 TransE 为基础的扩展。在 TransE 扩展模型以外，这里主要介绍全息表示模型（holographic embeddings, Hole）[Nickel 2015]。

Hole 提出使用头、尾实体向量的“循环相关”操作来表示该实体对。这里，循环相关*：$\mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$操作如下：

$$[\boldsymbol{l}_h * \boldsymbol{l}_t]_k = \sum_{i=0}^{d-1} \boldsymbol{l}_{h_i} \boldsymbol{l}_{t_{(i+k)\ mod\ d}}$$

循环相关操作可以看作张量乘法特殊形式，具有较强的表达能力，具有以下 3 个优点：1）不可交换性。循环相关是不可交换的，即$\boldsymbol{l}_h * \boldsymbol{l}_t \neq \boldsymbol{l}_t * \boldsymbol{l}_h$。而知识库中很多关系是不可交换的，因此该特点具有重要意义。2）相关性。循环相关操作的到的向量每一维都衡量了向量$\boldsymbol{l}_h$和$\boldsymbol{l}_t$的某种相似性。例如，循环相关的第一位$[\boldsymbol{l}_h * \boldsymbol{l}_t]_0 = \sum_{i=0}^{d-1} \boldsymbol{l}_{h_i} \boldsymbol{l}_{t_i}$相当于向量$\boldsymbol{l}_h$和$\boldsymbol{l}_t$的内积。该性质处理头、尾实体比较相似的关系（例如“夫妻”关系）时具有重要意义。3）计算效率高。循环相关操作还可以使用如下公式进行优化：

$$\boldsymbol{l}_h * \boldsymbol{l}_t = F^{-1}(\bar{F}(\boldsymbol{l}_h) \odot F(\boldsymbol{l}_t))$$

这里$F(x), F^{-1}(x)$为傅立叶变换与逆傅立叶变换，可以用快速傅立叶变换加速计算。

对于每个三元组(h,r,t)，Hole 定义了如下评分函数：

$$f_r(h,t) = \sigma(\boldsymbol{l}_r^T(\boldsymbol{l}_h * \boldsymbol{l}_t))$$

这里$\sigma(x) = \frac{1}{1+e^{-x}}$为 sigmoid 函数。

由于该模型刚刚提出，尚未验证其效果，单身无疑为知识表示学习提供了全新的视角，值得关注。

3.4. 复杂关系建模

TransE 由于模型简单，计算效率高，在大规模知识图谱上的效果很好。但是，也正是由于模型过于简单，导致了 TransE 在面对知识库中的复杂关系时效果并不好。

根据知识库中关系两端连接的实体的数目，可以将关系划分为 1-1，1-N，N-1 和 N-N 四种类型。例如，r 是一个 N-1 类型的关系，指的是：若知识库中存在三元组(h,r,t)，则知识库中可能存在三元组(h',r,t)，但不可能存在三元组(h,r,t')。我们将 1-N，N-1 和 N-N 称为复杂关系。

研究发现，各种知识获取算法在处理 4 种类型的关系时的性能差异较大[Bordes 2013]。以 TransE 为例，在处理复杂关系时性能显著降低，这与 TransE 的模型假设有密切关系。根据 TransE 的优化目标，面向 1-N，N-1 和 N-N 三种类型关系，我们可以推出以下结论：如果关系 r 是 N-1 关系，我们将会得到$\boldsymbol{l}_{h_0} \approx \boldsymbol{l}_{h_1} \approx \cdots \approx \boldsymbol{l}_{h_m}$。同样，这样的问题在关系 r 是 N-1 关系时也会发生，得到$\boldsymbol{l}_{t_0} \approx \boldsymbol{l}_{t_1} \approx \cdots \approx \boldsymbol{l}_{t_m}$。

例如，加入知识库中有 2 个三元组，分别是（美国，总统，奥巴马）和（美国，总统，布什）。这里的关系“总统”是典型的 1-N 的复杂关系。如果用 TransE 从这 2 个三元组学习知识表示，如图 3-3 所示，将会使奥巴马和布什的向量变得相同。

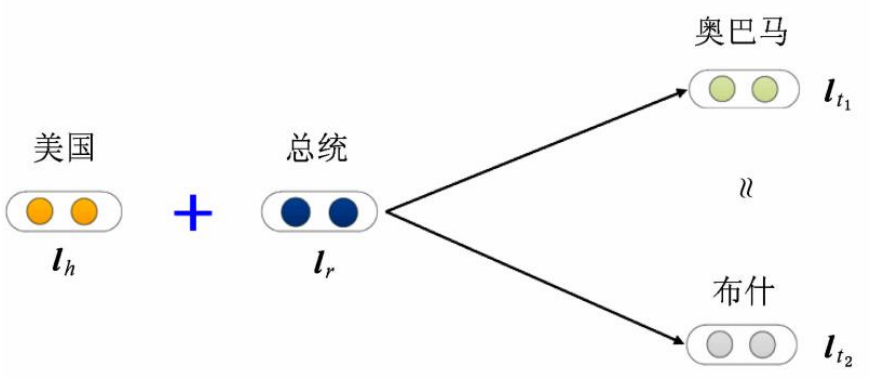

图 3-3 复杂关系示例

这显然不符合事实：奥巴马和布什除了作为美国总统这个身份上比较相似之外，其他很多方面都不尽相同。因此，由于这些复杂关系的存在，导致 TransE 学习得到的实体表示区分性较低。

因此，为了实现表示学习对复杂关系的建模，有大量关于 TransE 的扩展模型尝试解决这一挑战问题。这里我们简要介绍其中的几个代表模型。

3.4.1. TransH 模型

为了解决 TransE 模型在处理 1-N，N-1，N-N 复杂关系时的局限性，TransH 模型提出让一个实体在不同的关系下拥有不同的表示[Wang 2014]。

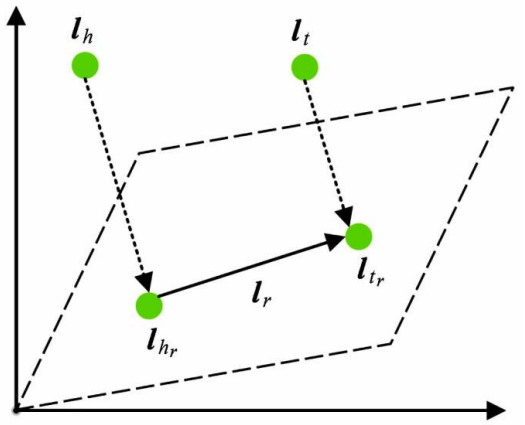

图 3-4 TransH 模型

如图 3-4 所示，对于关系 r，TransH 模型同时使用平移向量$\boldsymbol{l}_r$和超平面的法向量$\boldsymbol{w}_r$来表示它。对于一个三元组(h,r,t)，TransH 首先将头实体向量$\boldsymbol{l}_h$和尾实体向量$\boldsymbol{l}_t$沿法线$\boldsymbol{w}_r$投影到关系 r 对应的超平面上，用$\boldsymbol{l}_{h_r}$和$\boldsymbol{l}_{t_r}$表示如下：

$$\boldsymbol{l}_{h_r} = \boldsymbol{l}_h - \boldsymbol{w}_r{}^T \boldsymbol{l}_h \boldsymbol{w}_r$$
$$\boldsymbol{l}_{t_r} = \boldsymbol{l}_t - \boldsymbol{w}_r{}^T \boldsymbol{l}_t \boldsymbol{w}_r$$

因此，TransH 定义了如下损失函数：

$$f_r(h,t) = ||\boldsymbol{l}_{h_r} + \boldsymbol{l}_r - \boldsymbol{l}_{t_r}||_{L_1/L_2}$$

需要注意的是，由于关系 r 可能存在无限个超平面，TransH 简单地令$\boldsymbol{l}_r$与$\boldsymbol{w}_r$近似正交来选取某一个超平面。

3.4.2. TransR 模型

虽然 TransH 模型使每个实体在不同关系下拥有了不同的表示，它仍然假设实体和关系处于相同的语义空间$\mathbb{R}^d$中，这一定程度上限制了 TransH 的表示能力。TransR 模型则认为，一个实体是多种属性的综合体，不同关系关注实体的不同属

性。TransR 认为不同的关系拥有不同的语义空间。对每个三元组，首先应将实体投影到对应的关系空间中，然后再建立从头实体到尾实体的翻译关系[Lin 2015]。

如图 3-5 所示是 TransR 模型的简单示例。对于每个三元组(h,r,t)，我们首先将实体向量向关系 r 空间投影。原来在实体空间中与头、尾实体（用圆圈表示）相似的实体（用三角形表示），在关系 r 空间中被区分开了。

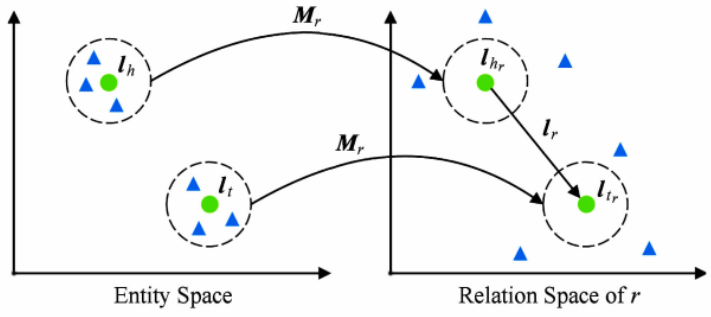

图 3-5 TransR 模型

具体而言，对于每一个关系 r，TransR 定义投影矩阵$\boldsymbol{M}_r \in \mathbb{R}^{d\times k}$，将实体向量从实体空间投影到关系 r 的子空间，用$\boldsymbol{l}_{h_r}$和$\boldsymbol{l}_{t_r}$表示如下：

$$\boldsymbol{l}_{h_r} = \boldsymbol{l}_h \boldsymbol{M}_r$$
$$\boldsymbol{l}_{t_r} = \boldsymbol{l}_t \boldsymbol{M}_r$$

然后使$\boldsymbol{l}_{h_r} + \boldsymbol{l}_r \approx \boldsymbol{l}_{t_r}$，因此，TransR 定义了如下损失函数：

$$f_r(h,t) = ||\boldsymbol{l}_{h_r} + \boldsymbol{l}_r - \boldsymbol{l}_{t_r}||_{L_1/L_2}$$

3.4.3. TransD 模型

虽然 TransR 模型较 TransE 和 TransH 有显著改进，它仍然有很多缺点：

1) 在同一个关系 r 下，头、尾实体共享相同的投影矩阵。然而，一个关系的头、尾实体的类型或属性可能差异巨大。例如，对于三元组（美国，总统，奥巴马），美国和奥巴马的类型完全不同，一个是国家，一个是人物。
2) 从实体空间到关系空间的投影是实体和关系之间的交互过程，因此 TransR 让投影矩阵仅与关系有关是不合理的。
3) 与 TransE 和 TransH 相比，TransR 由于引入了空间投影，使得 TransR 模型参数急剧增加，计算复杂度大大提高。

因此，研究者提出了 TransD 模型以解决这些问题，如图 3-6 所示：

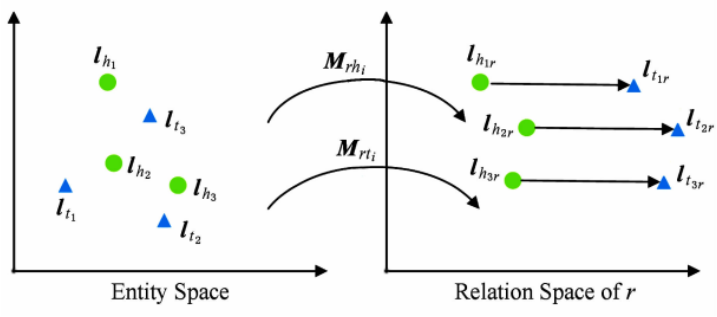

图 3-6 TransD 模型

给定三元组(h,r,t)，TransD 模型设置了 2 个分别将头实体和尾实体投影到关系空间的投影矩阵$\boldsymbol{M}_{rh}$和$\boldsymbol{M}_{rt}$，具体定义如下：

$$\boldsymbol{M}_{rh} = \boldsymbol{l}_{r_p}\boldsymbol{l}_{h_p} + \boldsymbol{I}^{d\times k}$$
$$\boldsymbol{M}_{rt} = \boldsymbol{l}_{r_p}\boldsymbol{l}_{t_p} + \boldsymbol{I}^{d\times k}$$

这里$\boldsymbol{l}_{h_p}, \boldsymbol{l}_{t_p} \in \mathbb{R}^d$，$\boldsymbol{l}_{r_p} \in \mathbb{R}^k$，下标 p 代表该向量为投影向量。显然，$\boldsymbol{M}_{rh}$和$\boldsymbol{M}_{rt}$与实体和关系均相关。而且，利用 2 个投影向量构建投影矩阵，解决了原来 TransR 模型参数过多的问题。最后，TransD 模型定义了如下损失函数：

$$f_r(h,t) = ||\boldsymbol{l}_h\boldsymbol{M}_{rh} + \boldsymbol{l}_r - \boldsymbol{l}_t\boldsymbol{M}_{rt}||_{L_1/L_2}$$

3.4.4. TranSparse 模型

知识库中实体和关系的异质性和不平衡性是制约知识表示学习的难题：

1) 异质性。知识库中某些关系可能会与大量的实体有连接，而某些关系则可能仅仅与少量实体有连接。
2) 不均衡性。在某些关系中，头实体和尾实体的种类和数量可能差别巨大。例如，“国籍”这个关系的头实体是成千上万不同的人物，而尾实体只有几百个国家。

为了解决实体和关系的异质性，TranSparse 提出使用稀疏矩阵代替 TransR 模型中的稠密矩阵[Ji 2016]，其中矩阵$\boldsymbol{M}_r$的稀疏度由关系 r 连接的实体对数量决定。这里头、尾实体共享同一个投影矩阵$\boldsymbol{M}_r$。投影矩阵$\boldsymbol{M}_r(\theta_r)$的稀疏度$\theta_r$定义如下：

$$\theta_r = 1 - (1-\theta_{min})N_r/N_{r^*}$$

其中，$0 \ll \theta_{min} \ll 1$为计算稀疏度的超参数，N_r表示关系 r 连接的实体对数量，r^*表示连接实体对数量最多的关系。这样，投影向量可以定义为：

$$\boldsymbol{l}_{h_r} = \boldsymbol{l}_h\boldsymbol{M}_r^h(\theta_r^h)$$
$$\boldsymbol{l}_{t_r} = \boldsymbol{l}_t\boldsymbol{M}_r^t(\theta_r^t)$$

TransSparse 对于以上 2 种形式，均定义如下损失函数：

$$f_r(h,t) = ||\boldsymbol{l}_{h_r} + \boldsymbol{l}_r - \boldsymbol{l}_{t_r}||_{L_1/L_2}$$

3.4.5. KG2E 模型

知识库中的关系和实体的语义本身具有不确定性，因此，KG2E 模型[He 2015]使用高斯分布来表示实体和关系。其中，高斯分布的均值表示的是实体或关系在语义空间中的中心位置，而高斯分布的协方差则表示该实体或关系的不确定度。

图 3-7 为 KG2E 模型示例，每个圆圈代表不同实体与关系的表示，它们分别与“比尔 克林顿”构成三元组，其中圆圈大小表示的是不同实体或关系的不确定度，可以看到“国籍”的不确定度远远大于其他关系。

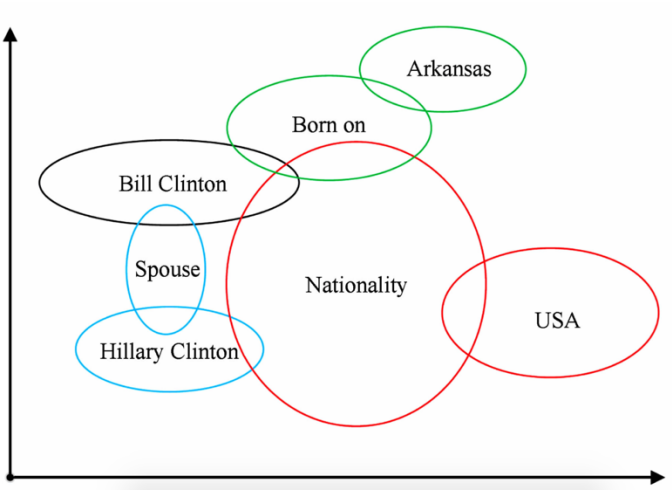

图 3-7 KG2E 模型

KG2E 使用$\boldsymbol{l}_h - \boldsymbol{l}_r$来表示头、尾实体之间的关系。这里$\boldsymbol{l}_h - \boldsymbol{l}_r$可以用一个概率分布来表示：

$$P_e \sim N(\boldsymbol{\mu}_h - \boldsymbol{\mu}_t, \Sigma_h + \Sigma_t)$$

而关系 r 同样是一个高斯分布$P_r \sim N(\boldsymbol{\mu}_r, \Sigma_r)$。因此，可以根据 2 个概率分布$P_e$和$P_r$的相似度来估计三元组的评分。KG2E 考虑 2 种计算概率相似度的办法：KL 距离和期望距离。需要注意的是，为了防止过拟合，KG2E 使用了对参数的强制限制：

$$\forall\, l \in E \cup R, c_{min}\boldsymbol{I} \le \Sigma_l \le c_{max}\boldsymbol{I}, c_{min} > 0$$

3.5. 多源信息融合

知识表示学习面临的另外一个重要挑战，是如何实现多源信息融合。现有的知识表示学习模型如 TransE 等，仅利用知识图谱的三元组结构信息进行表示学习，尚有大量与知识有关的其他信息没有得到有效利用，例如：

1) 知识库中的其他信息，如实体和关系的描述信息、类别信息等。

2) 知识库外的海量信息，如互联网文本蕴含了大量与知识库实体和关系有关的信息。

这些海量的多源异质信息可以帮助改善数据稀疏问题，提高知识表示的区分能力。如何充分融合这些多源异质信息，实现知识表示学习，具有重要意义。

在融合上述信息进行知识表示学习方面，已经有一些研究工作。这里简单介绍其中的几个代表性工作。

3.5.1. 融合语义类别标签信息

在很多知识库中，实体带有额外的语义类别标签信息。例如，实体“姚明”、“刘翔”、“乔丹”等会带有“运动员”标签；实体“中国”、“美国”、“俄罗斯”等实体会带有“国家标签”。SSE（Semantically Smooth Knowledge Graph Embedding）模型[Guo 2015]将这些语义类别标签信息融合进了知识表示学习技术中，从而能够更为精确地对知识库进行表示。

在 SSE 模型中，我们定义实体 e 有且仅有一个语义类别标签c_e。

SSE 模型的基本假设是：如果两个实体拥有同样的语义类别标签信息，则这两个实体对应的向量应该是相近的。

SSE 模型用$\mathcal{R}_1$来刻画这一假设：

$$\mathcal{R}_1 = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}||\boldsymbol{l}_i - \boldsymbol{l}_j||_2^2\, w_{ij}^{(1)}$$

其中，$w_{ij}^{(1)}$是矩阵$\boldsymbol{W}_1 \in \mathbb{R}^{n\times n}$中的项，$\boldsymbol{W}_1$的定义如下：

$$w_{ij}^{(1)} = \begin{cases} 1 & if\ c_{e_1} = c_{e_2} \\ 0 & otherwise \end{cases}$$

因此，总体损失函数可以定义为：

$$\mathrm{L} = L_1 + \lambda_1\mathcal{R}_1$$

其中，L_1为 TransE 的整体损失函数，λ_1为语义类别标签信息在损失函数中所占的比重，是一个超参数。

3.5.2. 融合实体描述信息

在很多知识库中，实体是带有自然语言文本描述的。DKRL（description-embodied knowledge representation learning）模型[Xie 2016]将这些实体描述信息融合进了知识表示学习技术中，从而能够更为精确地对知识库进行表示。在文本表示方面，DKRL 考虑了 2 种模型：一种是 CBOW[Mikolov 2013] [Mikolov 2013 a]，即将文本中的词向量简单相加作为文本表示；另一种是卷积神经网络（convolutional

neural network, CNN）[Collobert 2008] [Collobert 2011]，能够考虑文本中的词序信息。

如图 3-8 和图 3-9 所示，DKRL 可以利用 CBOW 和 CNN 根据实体描述文本得到实体表示，然后将该实体表示用于 TransE 的目标函数学习。

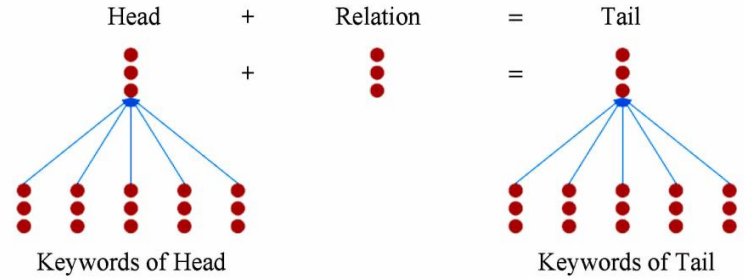

图 3-8 DKRL(CBOW)模型

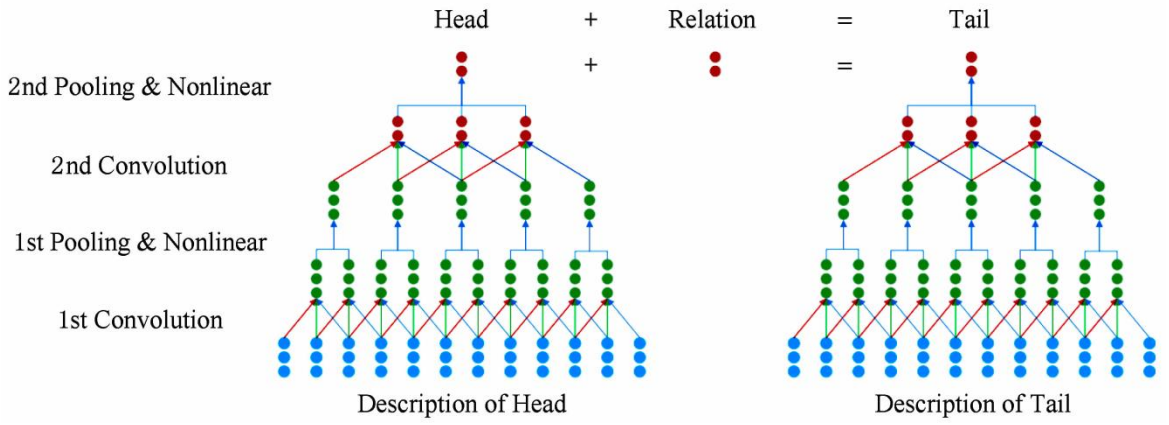

图 3-9 DKRL(CNN)模型

DKRL 的优势在于，除了能够提升实体表示的区分能力外，还能实现对新实体的表示。当新出现一个未曾在知识库中的实体时，DKRL 可以根据它的简短描述产生它的实体表示，用于知识图谱补全等任务。这对于不断扩充知识图谱具有重要意义。

3.5.3. 融合知识库外部的文本信息

DKRL 模型将实体描述文本信息融合进了知识表示学习。在此之外，知识库外部还存在海量的文本信息。例如，互联网网页中的文本就蕴含有大量实体信息以及它们之间的关联关系。[Wang 2014 a]将这些知识库外部的文本信息融合进知识表示学习，从而提高知识表示的性能。该模型利用 word2vec 学习维基百科正文中的词表示，利用 TransE 学习知识库中的知识表示。同时，利用维基百科正文中的链接

信息（锚文本与实体的对应关系），让文本中的实体对应的词表示与知识库中的实体表示尽可能接近，从而实现文本与知识库融合的表示学习。

3.6. 关系路径建模

在知识图谱中，多步的关系路径也能够反映实体之间的语义关系。Lao 等人曾提出路径约束的随机游走（Path-Constraint Random Walk，[Lao 2010]）、路径排序（Path Ranking Algorithm，[Lao 2011]）等算法，利用两实体间的关系路径信息预测它们的关系，取得显著效果，说明关系路径蕴含着丰富的信息。

为了突破 TransE 等模型孤立学习每个三元组的局限性，Lin 等人提出考虑关系路径的表示学习方法，以 TransE 作为扩展基础，提出 Path-based TransE（PTransE）模型[Lin 2015 a]。

图 3-10 展示的是 PTransE 考虑 2 步关系路径的示例。PTransE 模型面临的挑战在于：

1) 并不是所有的实体间的关系路径都是可靠的。为此，PTransE 提出 Path-Constraint Resource Allocation 图算法度量关系路径的可靠性。
2) PTransE 需要建立关系路径的向量表示，参与从头实体到尾实体的翻译过程。这是典型的组合语义问题，需要对路径上所有关系的向量进行语义组合产生路径向量。PTransE 尝试了 3 种代表性的语义组合操作，分别是相加、按位相乘和循环神经网络。相关数据实验表明，相加的组合操作效果最好。

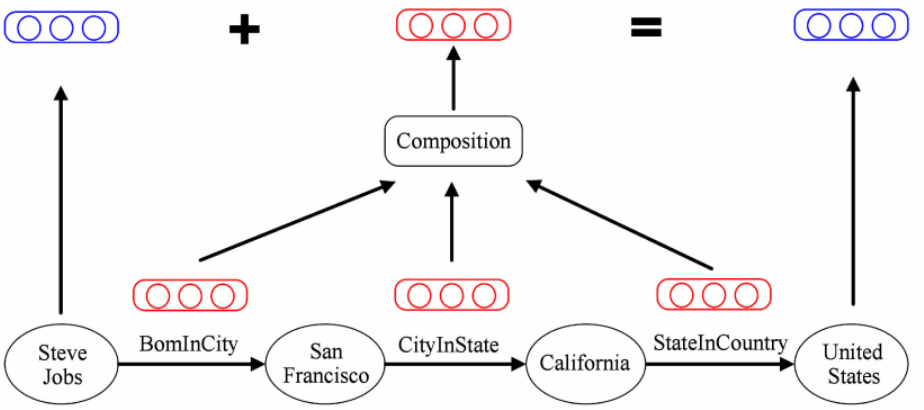

图 3-10 PTransE 模型

考虑了关系路径建模的工作还有[Garcia-Duran 2015]。关系路径的表示学习也被用来进行基于知识库的自动问答[Guu 2015]。

PTransE 等研究的实验表明，考虑关系路径能够极大提升知识表示学习的区分性，提高在知识图谱补全等任务上的性能。关系路径建模方面的工作还比较初步，在关系路径的可靠性计算、关系路径的语义组合操作等方面，还有很多细致的考察工作需要完成。

3.7. 参考文献

[AP 2014] AP, S. C., Lauly, S., Larochelle, H., Khapra, M., Ravindran, B., Raykar, V. C., & Saha, A. (2014). An autoencoder approach to learning bilingual word representations. In Advances in Neural Information Processing Systems (pp. 1853-1861).

[Bengio 2009] Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and trends® in Machine Learning*, *2*(1), 1-127.

[Bengio 2013] Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence, 35(8), 1798-1828.

[Blunsom 2014] Blunsom, P., Grefenstette, E., & Kalchbrenner, N. (2014). A convolutional neural network for modelling sentences. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics.

[Bordes 2011] Bordes, A., Weston, J., Collobert, R., & Bengio, Y. (2011). Learning structured embeddings of knowledge bases. In Conference on artificial intelligence (No. EPFL-CONF-192344).

[Bordes 2012] Bordes, A., Glorot, X., Weston, J., & Bengio, Y. (2012, April). Joint Learning of Words and Meaning Representations for Open-Text Semantic Parsing. In AISTATS (Vol. 22, pp. 127-135).

[Bordes 2013] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In Advances in neural information processing systems (pp. 2787-2795).

[Bordes 2014] Bordes, A., Glorot, X., Weston, J., & Bengio, Y. (2014). A semantic matching energy function for learning with multi-relational data. Machine Learning, 94(2), 233-259.

[Botha 2014] Botha, J. A., & Blunsom, P. (2014, June). Compositional Morphology for Word Representations and Language Modelling. In ICML (pp. 1899-1907).

[Chen 2015] Chen, X., Xu, L., Liu, Z., Sun, M., & Luan, H. B. (2015, July). Joint Learning of Character and Word Embeddings. In IJCAI (pp. 1236-1242).

[Collobert 2008] Collobert, R., & Weston, J. (2008, July). A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th international conference on Machine learning (pp. 160-167). ACM.

[Collobert 2011] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. Journal of Machine Learning Research, 12(Aug), 2493-2537.

[Fu 2014] Fu, R., Guo, J., Qin, B., Che, W., Wang, H., & Liu, T. (2014). Learning Semantic Hierarchies via Word Embeddings. In ACL (1) (pp. 1199-1209).

[Fyshe 2014] Fyshe, A., Talukdar, P. P., Murphy, B., & Mitchell, T. M. (2014, June). Interpretable semantic vectors from a joint model of brain-and text-based meaning. In Proceedings of the conference. Association for Computational Linguistics. Meeting (Vol. 2014, p. 489). NIH Public Access.

[Garcia-Duran 2015] Garcia-Duran, A., Bordes, A., & Usunier, N. (2015). Composing relationships with translations (Doctoral dissertation, CNRS, Heudiasyc).

[Guo 2015] Guo, S., Wang, Q., Wang, B., Wang, L., & Guo, L. (2015, September). Semantically Smooth Knowledge Graph Embedding. In ACL (1) (pp. 84-94).

[Guu 2015] Guu, K., Miller, J., & Liang, P. (2015). Traversing knowledge graphs in vector space. arXiv preprint arXiv:1506.01094.

[He 2015] He, S., Liu, K., Ji, G., & Zhao, J. (2015, October). Learning to represent knowledge graphs with gaussian embedding. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (pp. 623-632). ACM.

[Hu 2014] Hu, B., Lu, Z., Li, H., & Chen, Q. (2014). Convolutional neural network architectures for matching natural language sentences. In Advances in neural information processing systems (pp. 2042-2050).

[Huang 2012] Huang, E. H., Socher, R., Manning, C. D., & Ng, A. Y. (2012, July). Improving word representations via global context and multiple word prototypes. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1 (pp. 873-882). Association for Computational Linguistics.

[Jenatton 2012] Jenatton, R., Roux, N. L., Bordes, A., & Obozinski, G. R. (2012). A latent factor model for highly multi-relational data. In Advances in Neural Information Processing Systems (pp. 3167-3175).

[Ji 2016] Ji, G., Liu, K., He, S., & Zhao, J. (2016, February). Knowledge Graph Completion with Adaptive Sparse Transfer Matrix. In AAAI (pp. 985-991).

[Lao 2010] Lao, N., & Cohen, W. W. (2010). Relational retrieval using a combination of path-constrained random walks. Machine learning, 81(1), 53-67.

[Lao 2011] Lao, N., Mitchell, T., & Cohen, W. W. (2011, July). Random walk inference and learning in a large scale knowledge base. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (pp. 529-539). Association for Computational Linguistics.

[Le 2014] Le, Q. V., & Mikolov, T. (2014, June). Distributed Representations of Sentences and Documents. In ICML (Vol. 14, pp. 1188-1196).

[Lin 2015] Lin, Y., Liu, Z., Sun, M., Liu, Y., & Zhu, X. (2015, January). Learning Entity and Relation Embeddings for Knowledge Graph Completion. In AAAI (pp. 2181-2187).

[Lin 2015 a] Lin, Y., Liu, Z., Luan, H., Sun, M., Rao, S., & Liu, S. (2015). Modeling relation paths for representation learning of knowledge bases. arXiv preprint arXiv:1506.00379.

[Luo 2015] Luo, H., Liu, Z., Luan, H. B., & Sun, M. (2015). Online Learning of Interpretable Word Embeddings. In EMNLP (pp. 1687-1692).

[Luong 2013] Luong, T., Socher, R., & Manning, C. D. (2013, August). Better word representations with recursive neural networks for morphology. In CoNLL (pp. 104-113).

[Perozzi 2014] Perozzi, B., Al-Rfou, R., & Skiena, S. (2014, August). Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 701-710). ACM.

[Mikolov 2013] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).

[Mikolov 2013 a] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[Murphy 2012] Murphy, B., Talukdar, P. P., & Mitchell, T. (2012, January). Learning effective and interpretable semantic models using non-negative sparse embedding. Association for Computational Linguistics.

[Nickel 2011] Nickel, M., Tresp, V., & Kriegel, H. P. (2011). A three-way model for collective learning on multi-relational data. In Proceedings of the 28th international conference on machine learning (ICML-11) (pp. 809-816).

[Nickel 2012] Nickel, M., Tresp, V., & Kriegel, H. P. (2012, April). Factorizing yago: scalable machine learning for linked data. In Proceedings of the 21st international conference on World Wide Web (pp. 271-280). ACM.

[Nickel 2015] Nickel, M., Rosasco, L., & Poggio, T. (2015). Holographic embeddings of knowledge graphs. arXiv preprint arXiv:1510.04935.

[Reisinger 2010] Reisinger, J., & Mooney, R. J. (2010, June). Multi-prototype vector-space models of word meaning. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (pp. 109-117). Association for Computational Linguistics.

[Shi 2015] Shi, T., Liu, Z., Liu, Y., & Sun, M. (2015). Learning Cross-lingual Word Embeddings via Matrix Co-factorization. In ACL (2) (pp. 567-572).

[Socher 2012] Socher, R., Huval, B., Manning, C. D., & Ng, A. Y. (2012, July). Semantic compositionality through recursive matrix-vector spaces. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (pp. 1201-1211). Association for Computational Linguistics.

[Socher 2013] Socher, R., Bauer, J., Manning, C. D., & Ng, A. Y. (2013, August). Parsing with Compositional Vector Grammars. In ACL (1) (pp. 455-465).

[Socher 2013 a] Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013, October). Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the conference on empirical methods in natural language processing (EMNLP) (Vol. 1631, p. 1642).

[Socher 2013 b] Socher, R., Chen, D., Manning, C. D., & Ng, A. (2013). Reasoning with neural tensor networks for knowledge base completion. In Advances in neural information processing systems (pp. 926-934).

[Sutskever 2009] Sutskever, I., Tenenbaum, J. B., & Salakhutdinov, R. R. (2009). Modelling relational data using bayesian clustered tensor factorization. In Advances in neural information processing systems (pp. 1821-1828).

[Tang 2015] Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., & Mei, Q. (2015, May). Line: Large-scale information network embedding. In Proceedings of the 24th International Conference on World Wide Web (pp. 1067-1077). ACM.

[Tian 2014] Tian, F., Dai, H., Bian, J., Gao, B., Zhang, R., Chen, E., & Liu, T. Y. (2014, August). A Probabilistic Model for Learning Multi-Prototype Word Embeddings. In COLING (pp. 151-160).

[Turian 2010] Turian, J., Ratinov, L., & Bengio, Y. (2010, July). Word representations: a simple and general method for semi-supervised learning. In Proceedings of the 48th annual meeting of the association for computational linguistics (pp. 384-394). Association for Computational Linguistics.

[Wang 2014] Wang, Z., Zhang, J., Feng, J., & Chen, Z. (2014, July). Knowledge Graph Embedding by Translating on Hyperplanes. In AAAI (pp. 1112-1119).

[Wang 2014 a] Wang, Z., Zhang, J., Feng, J., & Chen, Z. (2014, October). Knowledge Graph and Text Jointly Embedding. In EMNLP (Vol. 14, pp. 1591-1601).

[Xie 2016] Xie, R., Liu, Z., Jia, J., Luan, H., & Sun, M. (2016, February). Representation Learning of Knowledge Graphs with Entity Descriptions. In AAAI (pp. 2659-2665).

[Yang 2014] Yang, B., Yih, W. T., He, X., Gao, J., & Deng, L. (2014). Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575.

[Yang 2015] Yang, C., Liu, Z., Zhao, D., Sun, M., & Chang, E. Y. (2015, June). Network Representation Learning with Rich Text Information. In IJCAI (pp. 2111-2117).

[Zhao 2015] Zhao, Y., Liu, Z., & Sun, M. (2015, January). Phrase Type Sensitive Tensor Indexing Model for Semantic Composition. In AAAI (pp. 2195-2202).

[Zhao 2015 a] Zhao, Y., Liu, Z., & Sun, M. (2015, July). Representation Learning for Measuring Entity Relatedness with Rich Information. In IJCAI (pp. 1412-1418).

第四章　研究设想

4. 研究设想

4.1. 研究背景

软件复用是解决软件危机、提高软件质量和开发效率的有效途径之一。一个软件项目在其生命周期中往往会产生大量数据，包括但不限于：源代码、文档、缺陷报告、邮件归档、论坛讨论、技术博客等。这些数据中蕴含了大量软件知识，是帮助软件开发者理解与复用软件项目的重要学习资源。如何从这些多源、异构、海量的软件项目学习资源中，高效、精准地找到与复用问题相关的信息，是当前软件开发人员面临的关键问题。

当前，有大量研究工作基于信息检索技术来帮助软件开发人员在软件项目的学习资源中进行信息定位。然而，对于一个用户查询，信息检索技术常常会返回几十乃至上百个相关的学习资源。用户需要耗费大量的时间与精力对其进行逐一阅读与理解，从中筛选出符合自己的关注点的信息，以解决在复用该软件项目时遇到的问题。为了帮助软件开发人员在软件复用的过程中更为高效、精准地在这些学习资源中进行信息定位，需要解决如下两方面的问题：

其一，是软件知识的提炼与关联问题。软件项目的学习资源中蕴含有丰富的软件知识，且这些知识之间具有广泛的语义关联。但在原始数据中，很多软件知识及其间的语义关联是以自然语言文本的形式描述的，这不利于学习者的浏览与查询，也不利于机器的分析处理。因此，对于一个待复用的软件项目，需要对其相关的学习资源进行语义理解，从中提炼出软件知识，建立起其间的语义关联，并加以组织管理。例如，从软件项目的源代码中可以提炼出代码实体（如类、方法等）以及其间的关联关系（如调用、继承等）；从文档中可以提炼出这些代码实体的功能描述、使用场景、约束条件、负责人员等知识；从论坛讨论中可以提炼出这些代码实体的使用示例、常见错误等知识；等等。解决这一方面的问题，是进行更为高效、精准的信息定位的前提条件。

其二，是用户查询的语义理解问题。在软件复用的过程中，软件开发人员遇到的问题是多种多样的，各种不同类型的问题具有不同的关注点。信息检索技术将用户输入的查询视为关键词的集合，虽然能够找到相关的学习资源，但由于缺乏对用户查询的语义理解，导致难以在找到的学习资源中进行进一步的精准定位。例如，

对于软件中的同一个功能，有些用户希望定位到与之相关的源代码、有些用户希望了解其使用场景、有些用户希望了解其内部实现原理、还有些用户希望了解其演化历史，等等。因此，为了更好地进行信息定位，需要对用户查询进行语义理解，包括其问题类型、关注点、关键实体、约束条件等。之后，以从软件资源中提炼出的软件知识为背景，借助从用户查询中提炼的语义信息，为软件开发人员提供更为高效、精准的信息定位服务。

4.2. 研究计划

因此，我们着眼于研究**面向软件复用的自动知识问答技术**。对于一个待复用的软件项目，如互联网上的开源软件项目或软件企业中积累的可复用软件项目，我们希望研究如何从其全生命周期数据（包括但不限于：源代码、文档、缺陷报告、邮件归档、论坛讨论、技术博客等）中自动提炼出软件知识，形成针对该软件项目的领域知识库；在此基础上，我们希望能够允许软件开发者以自然语言问句的形式输入其在复用该软件项目时遇到的问题，自动地理解问句的语义，并在知识库中进行针对性的语义搜索与推理，从而更为高效、精准地定位到问题的答案。

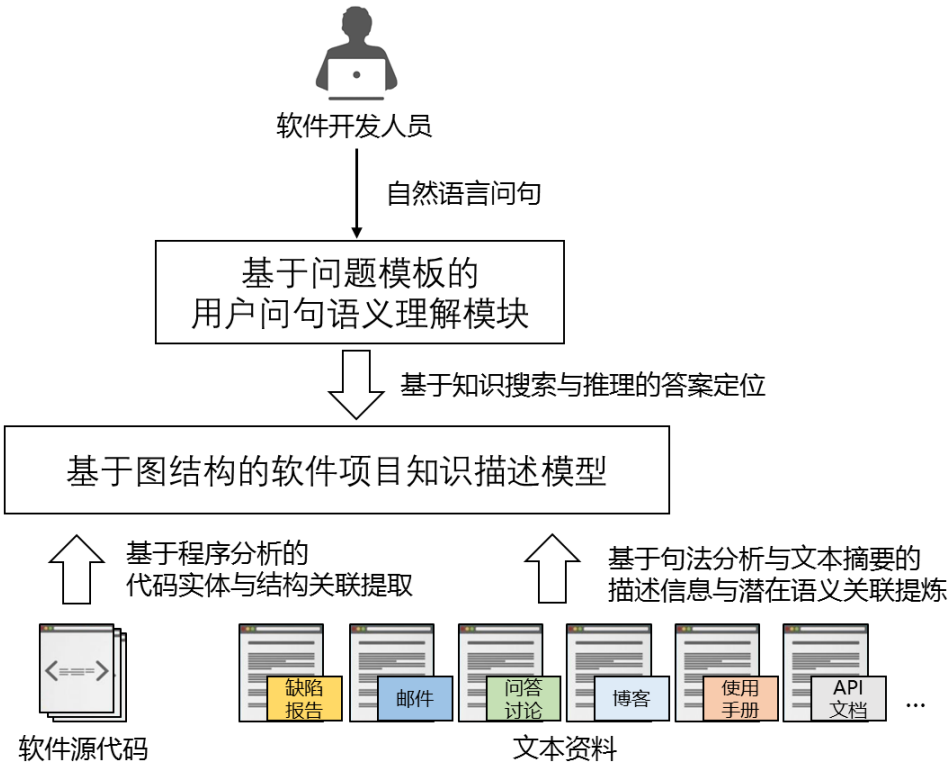

图 4-1 研究计划总体框架

图 4-1 是研究计划的总体框架。其中拟解决的关键技术问题包括：

1) 研究面向软件项目复用的知识描述模型

我们将对软件项目的相关学习资源的主要形态与特征以及软件开发人员在复用一个软件项目的过程中的知识需求进行调研。以此为基础，我们希望建立基于图结构的面向软件项目复用的知识描述模型，并基于图数据库系统提供相应的知识表示、管理与查询框架。软件项目中的知识将主要以“实体-关联”的图结构形式进行描述：主体是大量各种类型的实体（如类、方法、功能、缺陷、用例等）；实体间各种类型的关联关系（如调用、继承、包含、引用等）描述了其外部特征；实体和关联关系拥有各种类型的属性，以描述其内部特征：例如，对于一个类实体，其属性既包括类名、包路径、类注释等可以从这个类的源代码中直接解析出的基本知识，也包括功能描述、使用约束、内部原理、可靠性等需要从相关学习资源中提炼出来的进阶知识。

2) 研究基于软件项目学习资源的知识提炼方法

根据软件知识的描述模型，我们将进一步研究基于软件项目学习资源的知识提炼方法。对于软件项目源代码，我们拟通过静态分析（如：抽象语法树分析、控制流分析等）与动态分析（如：运行时环境分析、函数调用分析等）为主的程序分析技术对其进行知识提炼，形成软件项目的核心知识层。对于其它软件项目的学习资源，其主要形式为自然语言文本，其中包含了大量混杂的信息，例如：代码片段、表格结构、层次关联、引用链接、噪音等。我们将对这些结构信息进行识别与整理，并基于自然语言处理技术，从中抽取出软件知识描述模型中所需的知识加入到知识库中。其中，我们拟主要研究如何为软件中的代码实体抽取功能描述、实现原理与使用示例。

3) 研究面向软件项目复用的用户问题语义理解方法

我们将对软件开发人员在复用一个软件项目的过程中可能会遇到的问题类型进行调研。以此为基础，我们将为不同类型的问题制定问题模板，并基于自然语言处理技术实现从用户输入的自然语言问句到问题模板的语义映射。这些问题类型包括但不限于：一个功能是在源代码中的哪一部分中被实现的（特征定位）；两个代码实体之间具有怎样的关联关系；一个模块的内部实现原理是什么；一个类应该怎样被使用；一个方法的参数的取值范围是否有限定；等等。我们将在问题模板中对问题的语义进行规整，包括：问题类型、问题中涉及的关键实体以及它们在问题中所扮演的角色、问题关注的是这些实体的哪些侧面、问题中对答案的约束条件，等等。

这一工作是我们后续在知识库中进行语义搜索与推理，从而定位到问题的答案的前提条件。

4) 研究基于软件知识的答案搜索与推理方法

针对软件开发人员在复用一个软件项目的过程中遇到的各种不同类型的问题，我们将研究如何基于软件知识来从软件项目的相关学习资源中定位到问题的答案。我们主要从两个层次开展这一工作：一方面，对于不同的问题类型，我们将研究在知识库中相应的推理规则，从而精准地提取出问题的答案；另一方面，若知识库中不存在问题的答案，我们将以知识库中的软件知识作为背景知识，研究如何对软件项目的相关学习资源进行语义搜索，以较为精确地找到可能包含答案的段落。更具体地，我们希望通过对软件知识进行表示学习研究以挖掘其潜在语义，并将软件项目的学习资源中的文本信息与软件知识中的实体建立语义关联，从而能够判断用户问题与文本段落在潜在语义的层面上的相关性，过滤掉信息检索的返回结果中的大量噪音。

4.3. 作者简介

林泽琦，男，福建莆田人，2010 年 6 月本科毕业于北京大学信息科学技术学院计算机科学系，2014 年 9 月保送进入北京大学信息科学技术学院软件工程研究所攻读博士学位。导师张路教授，协助指导老师谢冰教授。研究方向包括软件工程、软件复用、软件数据挖掘、软件智能开发等。

- 参与的项目
  - 基于大数据的软件智能开发方法和环境（国家科技重大专项）
  - 大型网构化软件的可信开发、运行和演化技术体系与服务支撑环境（863 项目）
- 发表的论文及专利
  - **Zeqi Lin**, Bing Xie, Yanzhen Zou, Junfeng Zhao, Xuandong Li, Jun Wei, Hailong Sun, Gang Yin. Intelligent Development Environment and Software Knowledge Graph. Journal of Computer Science and Technology, 2(32), 242-249.
  - **林泽琦**，赵俊峰，谢冰. 一种基于图数据库的代码结构解析与搜索方法. 计算机研究与发展，2016, 53(3): 531-540.（第十三届全国软件与应用学术会议最佳论文）

- 李文鹏，王建彬，**林泽琦**，赵俊峰，谢冰. 面向开源软件项目的软件知识图谱构建方法. 计算机科学与探索，2016.（第十五届全国软件与应用学术会议最佳论文）
- 专利：朱子骁，**林泽琦**，谢冰. 一种从单元测试代码中提取 API 使用示例的方法与工具. 专利申请号：2016109356947.
- **Zeqi Lin**, Junfeng Zhao, Bing Xie. A Graph Database Based Crowdsourcing Infrastructure for Modelling and Searching Code Structure. Proceedings of the 6th Asia-Pacific Symposium on Internetware on Internetware. ACM, 2014.
- Zhu Zixiao, Zou Yanzhen, Bing Xie, Yong Jin, **Zeqi Lin**, Lu Zhang. Mining Api Usage Examples from Test Code. In Software Maintenance and Evolution (ICSME), 2014 IEEE International Conference on (pp. 301-310). IEEE.