基于大数据的软件项目知识图谱构造及问答方法

邹艳珍[1,2]，王敏[1,2]，谢冰[1,2]，林泽琦[3]
1. 北京大学信息科学技术学院，北京 100871；
2. 高可信软件技术教育部重点实验室（北京大学），北京 100871；3. 微软亚洲研究院，北京 100080

摘要
随着软件规模的不断扩大、软件演化周期的不断延长，构建软件项目知识图谱对软件维护、软件开发的意义越来越重大。如何基于软件项目开发过程中产生的源代码、邮件列表、缺陷报告等多源异构大数据，快速构建语义关联丰富的软件知识图谱，是软件工程领域亟待解决的关键问题。提出了以代码结构为核心的软件知识图谱模型，建立了“知识抽取-知识融合”两层软件知识图谱构造框架，该框架支持软件项目知识图谱的自动构造以及基于知识图谱的软件项目智能问答，有效提高了软件项目理解和软件复用的效率。目前，软件项目知识图谱已经在Apache开源社区以及国内著名软件企业成功展开应用实践。

关键词
软件复用；软件知识图谱；软件知识抽取；知识问答

中图分类号：TP311 文献标识码：A doi: 10.11959/j.issn.2096-0271.2021002

Software knowledge graph construction and Q&A technology based on big data

ZOU Yanzhen[1,2], WANG Min[1,2], XIE Bing[1,2], LIN Zeqi[3]
1. School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China
2. Key Lab of High Confidence Software Technologies (Peking University), Ministry of Education, Beijing 100871, China
3. Microsoft Research Asia, Beijing 100080, China

Abstract
With the increasing of software scale and software evolution, it is more and more important to construct software project knowledge graph for software maintenance and software development. Automatically constructing software knowledge graph with complex structure and rich semantic relations based on the multi-source heterogeneous mass data such as source code, mailing list, issue report and Q&A document generated in the process of software project development is a key challenge to be solved urgently in the field of software engineering. A code-centric software knowledge model was

proposed, a two-layer plugin framework for knowledge graph construction and software Q&A was provided, which improves the efficiency of software understanding and software reuse. At present, software project knowledge graph has successfully deployed in the Apache open source community and in the domestic famous enterprises.

Key words
software reuse, software knowledge graph, software knowledge extraction, knowledge Q&A

1 引言

软件复用可以提高软件开发的效率和质量。但随着现代软件规模的不断扩大和复杂度的日益提高，复用一个软件项目越来越难[1-2]。软件项目在其整个生命周期中形成并积累了大量的数据，如源代码、邮件列表、缺陷报告和问答文档等。这些数据中蕴含了规模庞大、结构复杂、语义关联丰富的软件知识，能够帮助软件开发人员理解软件功能，进行软件复用[3]。然而，组织、利用这些知识面临着以下挑战。

- 软件规模扩大引发的软件知识爆炸问题。随着软件规模的扩大和软件复杂度的提高，在软件开发与复用中需要理解与掌握的知识越来越多，开发者的学习成本越来越高。
- 软件数据中蕴含的信息在多源异构数据中呈碎片化分散的形态。在一个软件项目中，有助于复用者学习与理解这个软件项目的信息通常以相对独立的形态碎片化地分散在多源异构的数据之中，缺乏全局、统一的组织整理，彼此之间也缺乏关联。
- 大量信息是以无结构文本的形式表示的，如代码标识符、代码注释、邮件、用户手册、缺陷描述。文本信息具有很高的随意性和模糊性，对于同一件事情，不同的人可能会使用完全不同的单词对其进行描述。因此，即使复用者可以通过关键词来对无结构文本进行检索，其效果也不尽如人意。

为了有效地组织和利用多源异构软件大数据，更好地进行软件理解和复用，本文提出了基于大数据的软件项目知识图谱构造及问答方法。这里，软件项目知识图谱是指由不同类型软件数据的软件知识图有机融合构成的用于描述某一软件的知识体系。软件知识图是指由同一类型软件数据的软件知识实体及其之间的关联关系构成的图。具体地，本文提出了以代码结构为核心的软件项目知识图谱模型，以及“知识抽取–知识融合”两层插件框架，从而实现基于多源异构软件大数据的软件项目知识图谱自动构造；提出了基于知识图谱的软件智能问答方法和系统，从而有效地支持面向软件项目知识图谱的自然语言查询以及面向多源异构软件大数据的智能问答。在此基础上，本文设计并实现了软件项目知识图谱构造及智能问答平台SnowGraph，并为Apache开源社区中的192个软件项目和5家软件企业（浪潮通用软件有限公司、神州数码信息系统有限公司、东软集团股份有限公司、中创软件工程股份有限公司、金蝶软件（中国）有限公司）中的10个技术领域自动构造出软件项目知识图谱，并提供问答服务。

2 软件项目知识图谱模型设计

针对一个软件项目，本文的目标是基于多源异构软件数据自动构造出一个软

件知识图谱。软件知识图谱是一个有向图$G=(V,E)$，G中的点集$V=\{v_(1),v_(2),\cdots,v_(|V|)\}$代表软件项目中的概念和实体，$G$中的边集$E=\{e_(1),e_(2),\cdots,e_(|E|)\}$代表这些实体之间的关系。本文将软件复用中的知识模型简单地记为$T=(A,R)$，其中，A是所有实体类型的集合，存在一个从实体到实体类型的映射函数$\tau:V\rightarrow A$，使得知识图谱中的每个实体$v\in V$都具有一个特定的实体类型$\tau(v)\in A$；R是所有关系类型的集合，存在一个从关系到关系类型的映射函数$\emptyset:E\rightarrow R$，使得知识图谱中的每条关系$e\in E$都具有一个特定的关系类型$\emptyset(e)\in R$。因此，在构造知识图谱之前，首先需要分析软件大数据，并进行软件知识图谱模型的设计。

2.1 多源异构软件大数据

对于一个软件项目来说，常见的数据来源和类型包括源代码及其版本控制记录、需求与设计文档、用户手册、开发者邮件列表、用户论坛等。不同来源的数据有不同的存储格式，且其中包含的信息类型也各不相同。本节总结软件项目中有助于软件复用的几类代表性数据，分析并讨论其中包含的信息类型，这是实现知识建模以及知识图谱自动构造的基础。

（1）代码库

源代码是软件项目中最核心的资产。代码库中包含丰富的有助于复用者学习与理解该软件项目的信息，代码库信息大致可以分为以下3个部分。

- 结构信息：包括抽象语法树、抽象语法树中的结点绑定和结点引用关系、从抽象语法树中可以抽取到的控制流和数据流等。
- 描述信息：包括各个代码实体的标识符以及代码中的注释。这些描述信息是机器的底层表示（即结构信息）与人在高层认知和理解之间的沟通桥梁。
- 演化信息：指在版本控制系统中记录的源代码变更历史。演化信息有助于帮助复用者了解源代码中的每一个子部分的来龙去脉。

（2）正式文档

为了在项目中建立良好的沟通与协作，或为项目的用户和复用者提供指导和帮助，软件项目的开发者或管理者需要编写各种说明文档。常见的文档类型包括需求分析文档、系统架构与设计文档、接口文档、用户手册等。一般而言，这些文档具有规范的编写模板、准确规范的用词和叙述方式，以及良好的组织结构和索引目录。本文将符合以上特点的软件文档统称为正式文档。

软件项目中的正式文档可能以不同的数据格式进行存储与管理。对于软件企业内部的可复用软件项目而言，正式文档大多为DOC、PDF、PPT等数据格式。正式文档中的信息大致可以分为文本描述信息、篇章结构信息和关联引用信息。

（3）交流渠道

在一个软件项目中，开发者和复用者经常使用Web2.0形式的交流渠道进行沟通与讨论。这些交流过程被归档记录下来，对于后续的软件项目理解和软件复用都具有重要的参考价值。目前常见的交流渠道包括以下几种。

- 邮件列表：是最常见的交流渠道，很多老牌的开源社区以及软件企业将邮件列表作为沟通的主要工具。
- 事务跟踪系统：近年来，越来越多的软件项目开始使用事务跟踪系统（如JIRA、Bugzilla、Trac等）对大量的变更需求进行系统化的管理。在事务跟踪系统中，参与者可以提交一个事务，并通过自然语言、异常信息、代码片段等形式对事务进

行描述。

● 讨论板：除了邮件列表和事务跟踪系统，还有很多类型的交流渠道允许参与者自由地发布与该项目有关的内容，并支持跟帖式的讨论，本文将这些交流渠道统称为讨论板。讨论板既包括较为传统的软件项目自建的开发者论坛和用户论坛，也包括近年来兴起的以Stack Overflow为代表的在线问答社区等。

这些交流渠道可以被看作更广泛意义上的软件文档。与正式文档类似，交流渠道中也包含文本描述信息、篇章结构信息和关联引用信息，但在具体形态和分布上，其与正式文档有所不同：交流渠道中的信息总量更大，但碎片化特点明显，篇章结构信息弱化，缺乏有效的组织与索引机制；交流渠道中的参与者一般较多，因此对描述格式的要求并不严格。

2.2 以代码结构为核心的软件知识图谱模型

源代码是软件项目中最核心的资产，也是软件复用过程中最重要的知识来源。笔者提出了一种代码结构知识模型，如**图1**所示。模型中的实体细分为3类：Class（类或接口）、Method（成员方法）和Field（成员变量）。这些不同类型的实体之间具有各种不同类型的结构依赖关系，例如：Class实体之间的继承关系（Extend）、Method实体之间的调用关系（Invoke）。

代码结构知识能够反映出软件项目中高层的领域知识，从而能够对无结构文本的语义分析提供支持。在软件复用的语境下，基于代码结构知识对无结构文本的语义进行结构化表示是可行的，这也是本文以代码结构为核心来构造软件项目知识图谱的最重要原因。基于代码结构的文本语义表示方法在后续的知识利用过程中扮演着重要的角色。

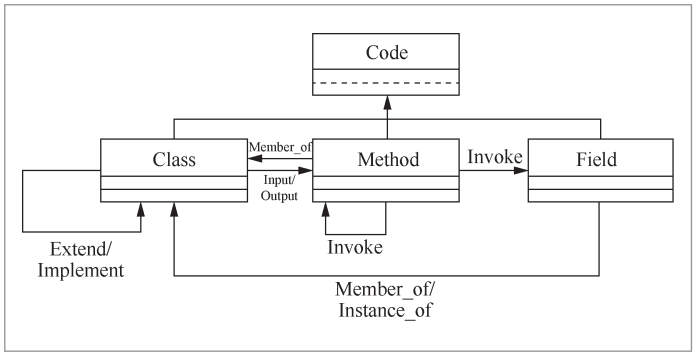


图1 代码结构知识模型

围绕代码结构知识，笔者提出软件复用中的知识模型，从而支持对源自版本控制系统、正式文档、交流渠道等源代码之外的其他数据中的知识进行组织与关联。**图2**展示了该知识模型的主体，以加粗的边框来标注模型中最基本的实体类型，包括：Code（代码实体）、Commit（代码提交实体）、Document（文档实体）和Participant（参与者实体）。该知识模型中共有22种实体和17种关系。

本文将软件项目的正式文档和交流渠道中的各种形态的数据统一抽象为知识图谱中的Document类型的实体。其中，文本描述信息被抽象为Document实体的内部属性，篇章结构信息被抽象为Document实体之间的Parent关系，关联引用信息则被抽象为Document实体之间以及Document实体到Commit实体或代码实体的Ref关系。文档实体可以被进一步细化为更多的实体类型。例如，对于软件企业中的DOC/PDF格式的需求文档和设计文档，根据文档的模板将其中合适的章节细化为需求实体（Requirement）或设计实体（Design），并通过章节的名称自动建立这两类实体之间的追踪关系（Trace）；使用Mail实体表示邮件列表中

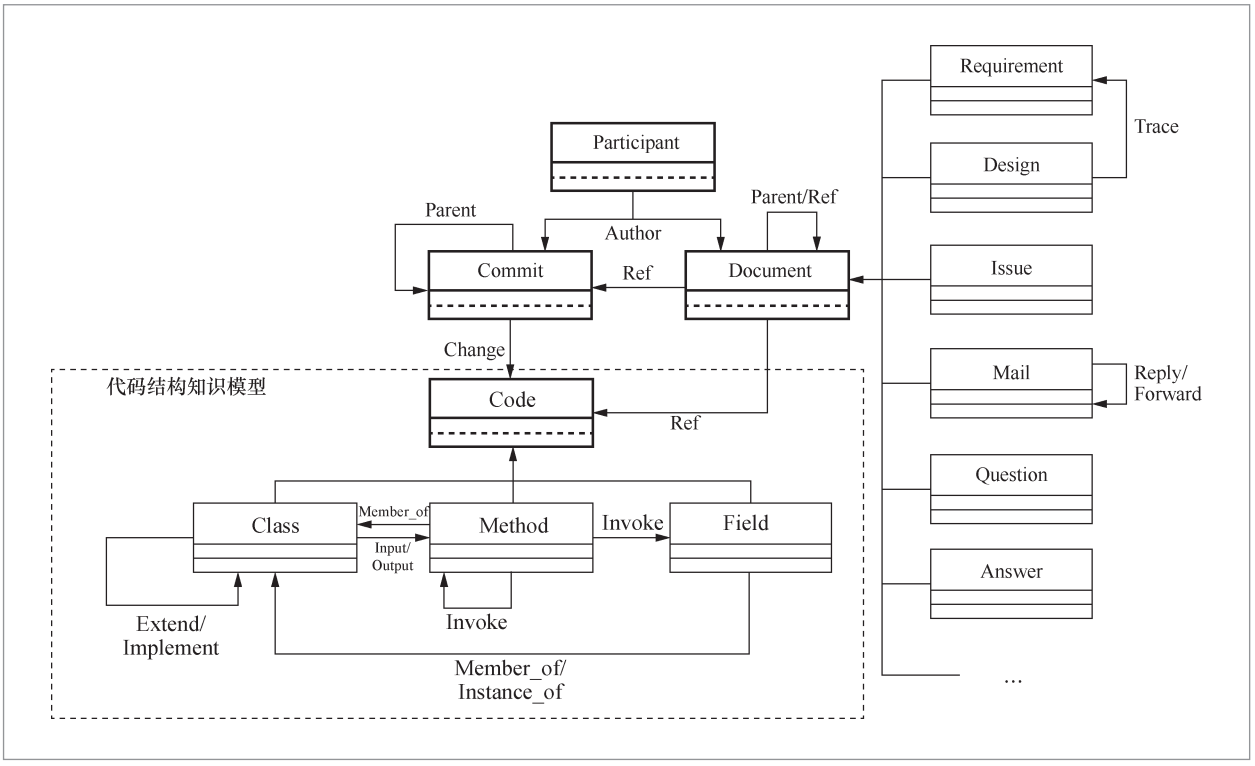


图 2 软件项目知识图谱的元模型

的每一条消息，并抽取出它们之间的回复关系（Reply）和转发关系（Forward）；对于来自像Stack Overflow这样的在线问答社区中的数据，将从中抽取出的Document实体，并将其细化为问题实体（Question）和答案实体（Answer）。

3 软件项目知识图谱的自动构造

软件项目知识图谱的自动构造是关键技术。这部分要解决的主要问题是：面对种类繁多，而且可能层出不穷的各种扩展知识和新来源、新格式的数据，如何设计一个框架以满足知识图谱构造的自动性和可扩展性。

针对上述问题，本文将知识图谱的自动构造过程划分为两个阶段：首先，对于不同的数据源，分别对其中的信息进行解析与整理，抽取出实体与关系，形成特定于该数据源的子知识图谱；其次，在这些相互独立的子知识图谱之间建立广泛关联、跨数据源的关系，并从中进一步提炼出更多的知识，从而形成最终的软件项目知识图谱。笔者将这两个阶段分别称为知识抽取和知识融合，如**图3**所示。

在知识抽取阶段，需要各种不同的知识自动抽取算法提供支持。每个知识自动抽取算法对应于一种特定的数据类型，将这种类型的数据作为输入，对其中的信息进行自动解析与整理，抽取出实体与关系，从而构造出相应的子知识图谱。例如，对于Java源代码数据，使用Eclipse JDT对其进行解析，获得源代码的抽象语法树，再从抽象语法树中抽取出代码实体以及这些代码实体之间的结构依赖关系，从而形成

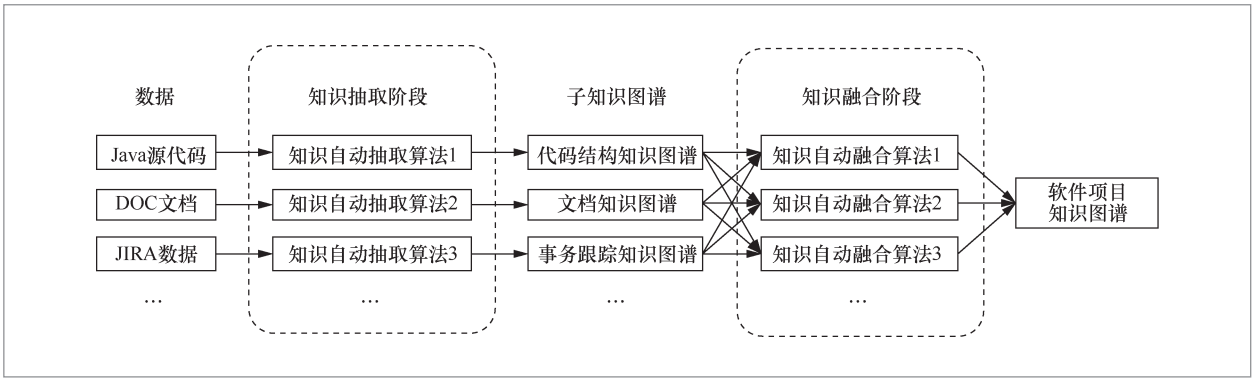


图 3 两阶段的知识图谱自动构造框架

一个由代码结构知识构成的子知识图谱；对于从JIRA事务追踪系统中导出的JSON格式数据，可以从中解析出Issue、Patch、Issue Comment、Issue User等类型的实体，并建立这些实体之间的关系，从而形成一个以事务跟踪知识构成的子知识图谱。这些知识自动抽取算法只负责处理原始数据中已有的结构化信息，并不涉及更深层次的分析与挖掘（如对无结构文本的语义分析）。

在知识融合阶段，需要各种不同的知识自动融合算法提供支持。在这些知识自动融合算法中，有些用于建立知识抽取阶段构造的各个子知识图谱之间广泛、跨数据源的关系，从而将这些原本相互独立的子知识图谱整合为一个完整、统一的软件项目知识图谱；有些用于对现有的简单知识进行进一步的分析与挖掘，提炼出更复杂的知识（如API使用示例实体、软件功能特征实体），并将其加入知识图谱中，从而更好地支持复用者的学习理解及计算机的分析利用[2,4-5]。后文将这些知识自动抽取算法与知识自动融合算法统称为知识自动获取算法[6-10]。

此外，为了实现动态的软件项目知识图谱构造和演化，笔者提出了一种可扩展的插件框架：系统为知识自动获取算法提供统一的接口，并将各种不同的知识自动获取算法实现为各个不同的插件；对于不同的软件项目，根据具体的数据情况与知识需求配置合适的插件，框架将自动按照特定的契约依次执行这些插件，渐次地向知识图谱中加入各个插件所负责获取的知识，从而形成最终的软件项目知识图谱。譬如在知识抽取阶段，对于代码结构知识，当前可以先实现针对Java、C#等常见的编程语言的知识自动抽取算法，但在将来还可能面对由Python、JavaScript等其他类型的编程语言所实现的源代码，需要在后续实现相应的自动抽取算法，并将它们扩展到现有的知识图谱自动构造过程中。

4 基于软件项目知识图谱的智能问答方法

基于上述软件项目知识图谱自动构造框架和方法，笔者设计并实现了相应的软件项目知识图谱构造及智能问答平台——SnowGraph（software knowledge graph），其系统框架如**图4**所示。SnowGraph遵循数据-信息-知识-智慧（data-information-knowledge-wisdom，DIKW）的层次体系：对于一个待复用的软件项目，以自动化

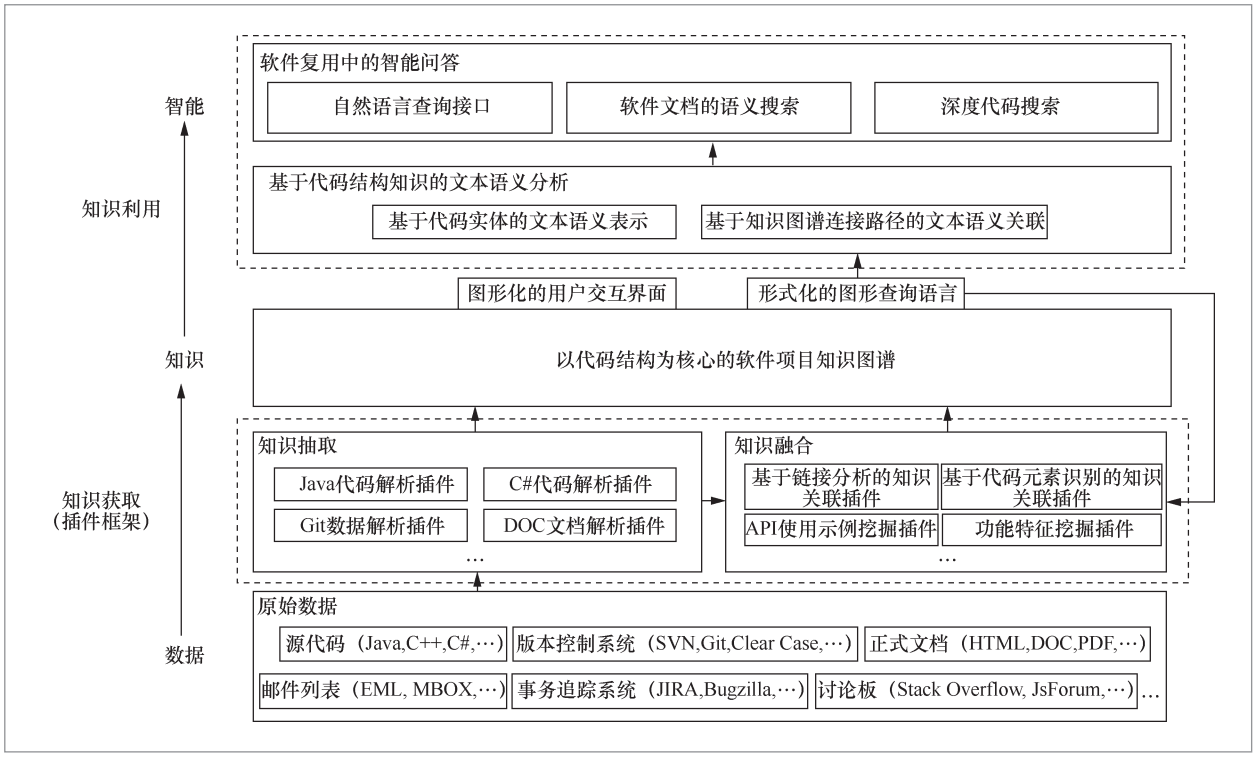


图 4 SnowGraph 平台的系统框架

方法对其中的多源异构数据进行处理，将分散、非结构化的软件数据提炼为广泛关联、语义性强的知识，并以知识图谱的形式进行表示；在此基础上，将知识图谱融入机器对无结构文本的处理过程之中，进而为复用者提供准确、有效的智能问答服务，从而提高软件复用过程的效率与质量。

4.1 面向软件项目知识图谱的自然语言问答

软件项目知识图谱往往采用Neo4j等图数据库存储结构，支持基于Cypher的形式化查询；需要开发人员熟悉并掌握Cypher语法，人工将用户意图转化为Cypher查询语句。许多软件知识图谱的数据量较大、数据类型较多，开发人员在查询过程中存在较高的学习成本，因此，笔者研究并提出了一种基于自然语言的知识库/知识图谱查询方法[11]。该方法能够将用户的自然语言问句自动转化为Cypher形式化查询语句，有效支持了面向软件项目知识图谱的自然语言问答。

该方法首次提出了推理子图的概念，并通过构造和度量推理子图，建立自然语言到形式化查询语句之间的转化桥梁。具体地，首先抽取软件项目知识图谱元模型（即针对软件项目知识图谱的实体类型以及关系类型进行提炼得到的元模型）。其次，对不同类型的自然语言查询语句进行分析，对自然语言语句进行切词、去停用词等基本预处理，同时解析得到自然语言词语的词性、主被动态等信息，并将自然语言词语与知识图谱元素进行匹配，得到词语匹配图。再次，生成与度量推理子图，将自然语言转换成知识图谱元模型上的一系列子图，这些子图被称为推理子图。笔者提

出并实现了一个基于隐藏结点扩展的推理子图生成方法，同时基于推理子图的结构特征和自然语言的文本信息，研究了如何对不同的子图进行合理度量，并给出了一个统一的度量方法。最后，构造形式化查询，目标是将推理子图转化为Cypher查询语句，并在Neo4j数据库上执行，最后返回用户查询结果。查询中同时提供推理子图的中间结果，使用户可以理解形式化查询构造方法的前因后果，从而选择合适的候选结果，解决了许多查询及检索系统需要用户自行验证答案正确性的问题。

图5展示了一个面向软件项目知识图谱的自然语言问答示例。软件开发者在复用Apache Lucene的开源项目时，提出了这样的一个问题：Which issue written by Alex that modifies a method called by IndexWriter？关于这个问题，可以在知识图谱上进行自然语言单词与图谱元素（如实体、边等）匹配，得到词语匹配图，然后进行逻辑推导，理解问题含义，推导出符合语义的推理子图，构造约束条件，最终构造形式化查询语句（Cypher查询语句），并执行得到返回结果。

4.2 面向多源异构软件大数据的智能问答

针对开发者给出的面向多源异构软件大数据的复杂问题，需要从软件项目的各类自然语言文档中抽取出一段最合适的文本作为答案。然而传统的基于关键词匹配的方法并不准确，在实际过程中开发者还必须对大量的搜索结果进行浏览与筛选。

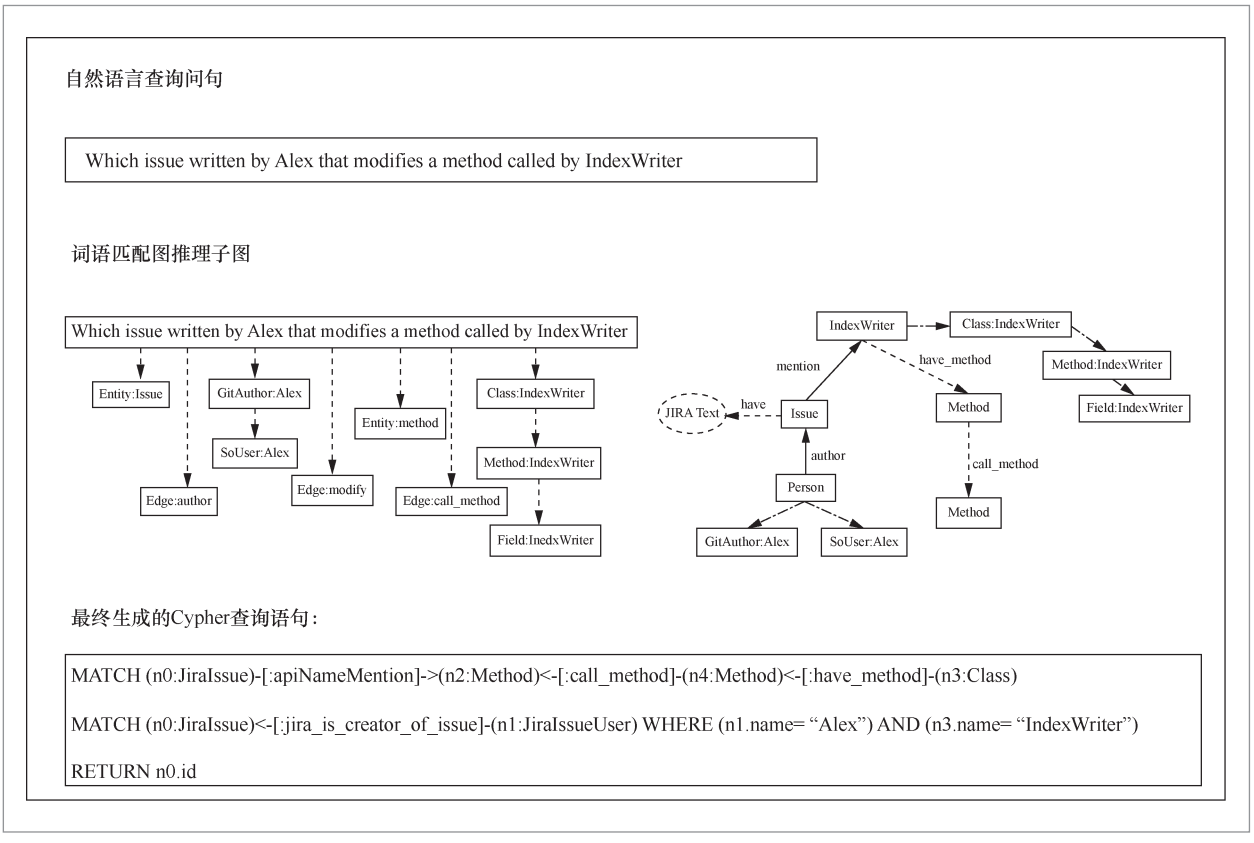


图 5 自然语言查询示例

为了解决这一问题，笔者提出了融合代码知识的智能问答方法[12-13]。该方法借助软件项目的知识图谱来计算不同单词之间的潜在语义相关度，从而对候选文本集合进行筛选与评估，返回更准确的答案。

具体解决方案是：以单词匹配为主，结合多种歧义消除技术，识别出与一段软件文本相关的代码实体集合；参考信息检索领域的指标，度量软件文本与代码实体之间的关联关系的强弱程度，从而将软件文本的语义结构化地表示为一个带权重的代码实体的集合；基于知识图谱表示学习技术，以代码实体之间的关联关系为桥梁，度量软件文本间的语义相似度；以语义相似度为核心特征，综合考虑多方面特征，建立对候选答案段落的评估模型，从而抽取出最合适的文本片段，并将其作为答案返回给复用者。具体方法流程如**图6**所示。

不同于现有的相关工作中常用的基于LDA、Word2Vec等统计学习方法的文档搜索改进策略[5]，该方法借助软件项目源代码中的代码实体对自然语言文本的语义进行结构化表示，并利用代码实体之间的结构依赖关系实现了对文本之间的潜在语义关联的更直接、更有效的挖掘与利用，从而显著地改进文档搜索的效果。

5 软件项目知识图谱的应用

目前，SnowGraph已在开源社区与软件企业分别开展应用实践。在开源社区方面，针对Apache开源软件基金会中的192个开源软件项目，笔者共收集了包含软件源代码、软件版本记录、软件变更提交、软件缺陷追踪数据以及在线问答社区数据等在内的约80 GB的原始数据，并为这些软件项目分别构建了相应的知识图谱。由此形成了面向Apache开源软件整体的软件项目知识图谱集，验证了本文方法的有效性。在企业应用方面，SnowGraph目前已经在神州数码信息系统有限公司、中创软件工程股份有限公司、金蝶软件（中国）有限公司、东软集团股份有限公司、浪潮通用软件有限公司等软件企业进行了安装运行，为这些公司的优势领域构建了10个技术领域的软件知识图谱（见**表1**），并在此基础上提供了智能问答服务。

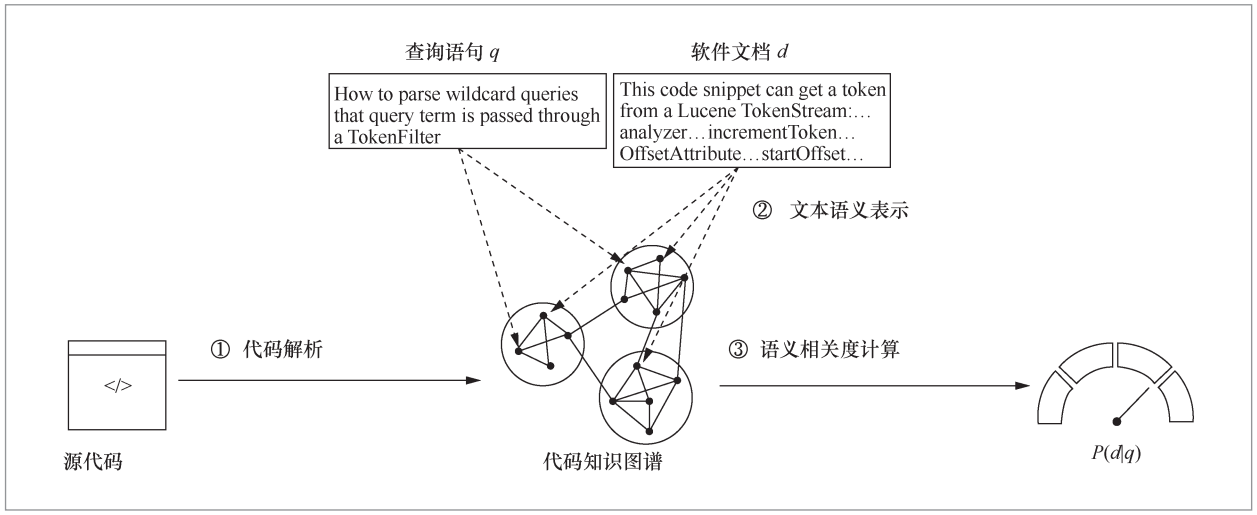


图 6 融合代码知识的答案检索模块方法流程

表1 企业知识图谱情况简介

企业名称	应用领域	原始数据类型	结点数	关系数
神州数码信息系统有限公司	市民服务	项目源代码(Java)、开发文档(DOCX)	8 482	6 604
	小额支付	项目源代码(Java)、开发文档(DOCX)	27 439	29 715
中创软件工程股份有限公司	金融	项目源代码(Java)、开发文档(DOCX)	275 904	671 831
	教育	项目源代码(Java)、开发文档(DOCX)	33 364	56 451
金蝶软件(中国)有限公司	财务会计	项目源代码(Java)、开发文档(DOCX)、版本控制数据(SVN)	72 059	138 669
	生产制造	项目源代码(Java)、开发文档(DOCX)、版本控制数据(SVN)、演示文稿(PPTX)	56 138	107 638
东软集团股份有限公司	互联网金融	项目源代码(Java)、开发文档(DOCX)、演示文稿(PPTX)	4 738	3 859
	企业应用	项目源代码(Java)、开发文档(DOCX)、版本控制数据(Git)、演示文稿(PPTX)	1 248	1 352
浪潮通用软件有限公司	智能仓储	项目源代码(C#)、开发文档(DOCX)、版本控制数据(Git)	102 298	270 483
	装备制造	项目源代码(C#)、开发文档(DOCX)、版本控制数据(Git)	131 524	918 130

5.1 面向Apache开源社区的软件项目知识图谱构造

这里以Apache Lucene开源软件项目为例，介绍软件项目知识图谱构造的基本过程和情况。与该项目相关的软件数据具体如下。

- 源代码：总代码行数超过80万行。
- Git版本控制数据：共有67次大的版本变更，合计超过26 000次代码提交记录。
- 邮件列表数据：积累了超过16万封邮件。
- JIRA事务跟踪数据：积累了7 500多个事务报告。
- Stack Overflow上的相关问答对：包括9 300多个问题及其答案。

基于这些数据，SnowGraph进行了Apache Lucene开源软件项目知识图谱的自动构造。在一个具有32 GB内存和2.9 GHz处理器的服务器上，自动构造出这个软件项目知识图谱所耗费的时间为39 min。**表2**、**表3分别**给出了这一知识图谱中的实体和关系的基本统计信息。

图7展示了最终生成的Apache Lucene

表2 Apache Lucene 的软件知识图谱中实体的基本统计信息

实体类型	说明	实体数量
Class	类	2 346
Method	成员方法	16 963
Field	成员变量	7 434
Commit	代码变更提交	26 421
Mail	邮件	160 682
Issue	事务	7 591
Patch	事务中附加的补丁说明	12 715
Issue Comment	事务中的评论	74 027
Question	Stack Overflow中的问题	9 340
Answer	Stack Overflow中的答案	11 729
QA Comment	Stack Overflow中的评论	20 887
Participant	参与者	24 762
API Example	API使用示例	10 967
Functional Feature	功能特征	1 680

表3 Apache Lucene 的软件知识图谱中关系的基本统计信息

关系类别	关系数量
Code-Code	81 908
Commit-Commit	26 824
Document-Document	187 590
Code-Commit	65 598
Code-Document	939 112
Commit-Document	12 715
Participant-Commit/Document	601 651
Participant-Participant	861
API Example/Functional Feature-Code/Document	152 122

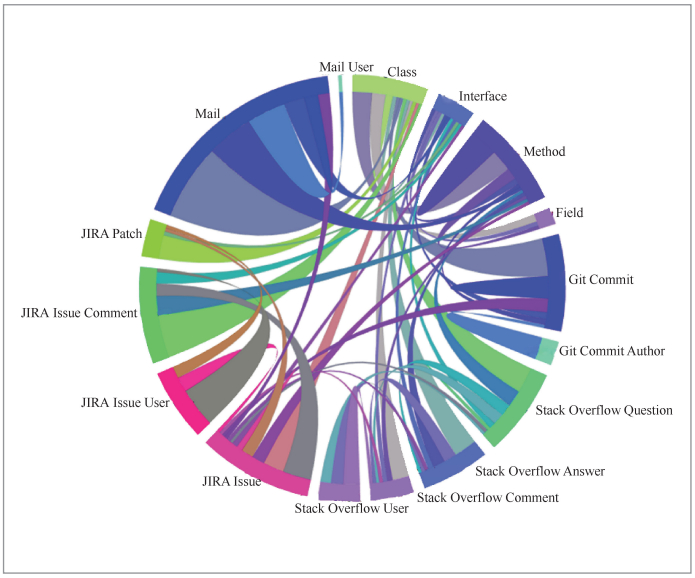


图 7 Apache Lucene 知识图谱弦图

的知识图谱弦图。Apache Lucene软件项目知识图谱共抽取了16种类型的知识实体，共计323 578个实体，17种主要类型的关联关系，共计2 200 683条边。

5.2 面向软件企业的应用示范

本节以神州数码技术有限公司为例，介绍SnowGraph在企业中的应用状况。在与神州数码技术有限公司的合作研发和应用示范过程中，SnowGraph为该公司智慧城市的市民服务与小额支付两个领域的软件智能开发提供了有效的支持。

神州数码技术有限公司智慧城市公共信息服务开发平台规模大、功能繁多复杂、平台文档庞杂，这使得软件开发人员熟练使用该平台进行软件项目开发的学习成本较高。为此，神州数码技术有限公司将笔者团队研发的软件项目知识图谱自动构造方法与智能问答工具作为解决方案，对平台以及智慧城市领域已开发完成的软件项目所包含的各种资源（包括源代码、需求文档、设计文档、测试文档、用户手册、参考文献等）进行了深度的解析与挖掘，实现了大规模软件项目数据的知识化，有效提高了智慧城市领域与税务领域的软件构造效率与质量。例如，针对市民服务领域，从109 GB的软件项目源数据中提取了2 715 134条软件实体知识、31 104 408条软件属性知识和11 270 694条软件语义关联知识，形成了一个面向市民服务领域的详尽且语义丰富的软件知识图谱。

6 相关工作

近年来，关于知识图谱的相关研究工作逐渐增多。相关工作可以大致划分为知识抽取、知识融合和知识利用3个方面。

知识抽取是知识图谱构建的第一步，指的是从原始数据中自动抽取出实体、关系和属性等知识单元。对于结构性较强的数据，一般可以采用人工编写规则与启发式算法相结合的方法来抽取知识。目前，学术界主要关注如何从互联网上海量的开放领域（open domain）的无结构文本数据中自动抽取知识。代表性的工作有：Finkel J R等人[14]提出了一种基于条件随机场（conditional random field，CRF）序列模型的实体抽取方法，并将其集成到Stanford自然语言处理工具集中；Angeli G等人[5]提出了一种基于句法结构分析的三元组抽取方法，研发了相应的工具OpenIE，并将其集成到Stanford自然语言处理工具集中。

从开放领域的无结构文本数据中抽取出的知识可能存在缺失，同时也可能包含大量冗余甚至错误的内容，因此必须对其进行清理、整合和补全。这一过程被称为知识融合。这方面的代表性工作有：Dong X等人[6]提出了一种基于有监督机器学习的三元组可信度评估方法，并在该方

法的指导下为谷歌知识图谱扩充了大量来自互联网的无结构文本数据的三元组知识；Gardner M等人[7]提出了一种基于随机路径游走的知识推理方法PRA（path ranking algorithm），该方法能够根据知识图谱中已有的三元组推测出新的三元组；Socher R等人[8]基于神经网络对知识图谱中的三元组进行了表示学习，从而能够推测出新的三元组。

在知识利用方面，现有的研究工作主要关注如何为知识图谱提供自然语言查询支持，以及如何基于知识图谱实现语义化的搜索引擎。

面向知识图谱的自然语言查询方法指的是对用户以自然语言问句的形式提出的问题进行语义解析，将其转换为某种结构化的语义表示形式，从而在知识图谱上进行符号化的推理，进而找到符合要求的实体作为答案返回给用户。这一问题的关键在于从自然语言问句到结构化的语义表示形式的转换，即语义解析（semantic parsing）。早期人们大多采用规则驱动的方法，多见于特定领域的语义解析任务。这类方法通过人工制定的模板或语法规则来解析自然语言问句的语义，例如，Woods W A[15]研发了一个规则驱动的自然语言查询系统LUNAR，用来支持科研人员对月球地质数据的查询。此外，以本文所处的软件工程领域为例，Begel A等人[16]研发了一个面向软件项目数据的自然语言查询系统Hoozizat，可以支持形如“FileModifiedByFileRevision ModifiedByChangesetCommittedByNeeseBarkema”的自然语言查询语句；Lin J等人[17]研发了一个面向软件项目数据的自然语言查询系统TiQi，该系统中制定了更加复杂的语法规则，从而对更多的自然语言问句句式进行支持。近年来，随着深度学习技术的飞速发展，研究者们提出了各种基于深度神经网络模型的语义解析方法。代表性的工作有：Dong L等人[18]使用带注意力机制的长短期记忆（long short-term memory，LSTM）网络将自然语言问句转换为相应的逻辑表达式；Jia R等人[19]使用指针网络（pointer network）解决了语料库中的单词稀疏问题；Yin P C等人[20]提出了一种能够对树形结构进行编解码的深度语义解析器；等等。

基于知识图谱的语义搜索方法指的是在信息检索的过程中对知识图谱中的实体和关系进行利用，从而实现语义化的文本搜索引擎。典型地，Xu Y等人[21]提出了一种基于Wikipedia描述信息的查询扩展方法。对于一个查询语句，首先识别出其中提到了Wikipedia中的哪些实体。在Wikipedia中，这些实体各自有具体的文本描述信息，包括标题、概述、详细内容、分类标签、附注、相关链接等不同的部分。这一方法将各个实体的这些不同部分的文本描述信息作为附加的查询内容，分别计算它们与待查询文本的相似度。之后，将这些相似度与原始的文本相似度线性地组合在一起，作为新的相似度计算方法，并通过有监督机器学习的方式来获得这些相似度在线性组合时的权重。Bendersky M等人[22]利用Wikipedia中的实体描述信息来度量实体的重要性，据此来计算查询语句中单词的重要性，从而改善文本检索。Dalton J等人[23]提出了一种基于多维度实体特征的查询扩展方法EQFE（entity query feature expansion）。这一方法不仅考虑了查询语句中提及的实体，还考虑了原始查询结果中排在前列的文本中提及的实体。同时，在查询扩展时，该方法不仅考虑了实体的名称，还考虑了它们在知识图谱中的别名、类别、具体文本描述等。这些特征通过一个机器学习模型综合在一起，从而形成了新的相似度计算方法。

Pan D Z等人[24]提出了一种使用Freebase知识图谱中的实体名称来实现查询扩展的方法。有两类实体的名称被用于对原始的查询语句进行扩展：第一类是其名称在原始的查询语句中完整或部分出现了的实体；第二类是这些实体在知识图谱中的邻接实体。在此基础上，使用Dempster-Shafer证据理论进行不确定推理，从这些候选实体中选出具有足够证据支持的实体，并以它们的名称来扩充原始查询语句。Guisado-Gámez J等人[25]提出了一种基于知识图谱子图模式特征的查询扩展方法SQE（structural query expansion）。这一方法指出，在Wikipedia中，若某个实体和在查询语句中出现的实体之间符合某些特定的子图模式，则这个实体与查询语句具有较高的语义相关性，可以加入扩展后的查询语句中。

7 结束语

本文研究并提出了基于大数据的软件项目知识图谱的自动构造及问答技术。本文首先对软件项目中多源异构的数据进行了分析和总结，给出了一种以代码结构为核心的软件复用中的知识模型。该知识模型具有很好的通用性与可扩展性，能够对未来可能出现的新的知识需求、知识来源，以及知识抽取、关联、提炼方法进行适应与支持。在此基础上，本文提出了软件项目知识图谱的自动构造方法和基于知识图谱的智能问答方法，并介绍了基于上述研究工作所实现的支持平台SnowGraph及其目前的应用状况。未来，基于软件开发过程中的更多数据类型，进一步的工作是进行软件知识图谱知识实体的扩充，以及建立更多的语义关联，并提供更精准的交互式智能问答服务。

参考文献：

[1] 杨芙清, 梅宏. 软件复用与软件构件技术[J]. 电子学报, 1999, 27(2): 68-75.
YANG F Q, MEI H. Software reuse and software component technology[J]. Acta Electronica Sinica, 1999, 27(2): 68-75.

[2] TOMER A, GOLDIN L, KUFLIK T, et al. Evaluating software reuse alternatives: a model and its application to an industrial case study[J]. IEEE Transactions on Software Engineering, 2004, 30(9): 601-612.

[3] 李文鹏, 王建彬, 林泽琦, 等. 面向开源软件项目的软件知识图谱构建方法[J]. 计算机科学与探索, 2017, 11(6): 851-862.
LI W P, WANG J B, LIN Z Q, et al. Software knowledge graph building method for open source project[J]. Journal of Frontiers of Computer Science & Technology, 2017, 11(6): 851-862.

[4] 刘峤, 李杨, 段宏, 等. 知识图谱构建技术综述[J]. 计算机研究与发展, 2016, 53(3): 582-600.
LIU Q, LI Y, DUAN H, et al. Knowledge graph construction techniques[J]. Journal of Computer Research and Development, 2016, 53(3): 582-600.

[5] ANGELI G, PREMKUMAR M J J, MANNING C D. Leveraging linguistic structure for open domain information extraction[C]//The 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. [S.l.:s.n.], 2015: 344-354.

[6] DONG X, GABRILOVICH E, HEITZ G, et al. Knowledge vault: a web-scale approach to probabilistic knowledge fusion[C]//The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2014: 601-610.

[7] GARDNER M, MITCHELL T. Efficient and expressive knowledge base completion using subgraph feature extraction[C]//

The 2015 Conference on Empirical Methods in Natural Language Processing. [S.l.:s.n.], 2015: 1488–1498.

[8] SOCHER R, CHEN D, MANNING C D, et al. Reasoning with neural tensor networks for knowledge base completion[C]//Advances in Neural Information Processing Systems. New York: ACM Press, 2013: 926–934.

[9] RIGBY P C, ROBILLARD M P. Discovering essential code elements in informal documentation[C]//2013 35th International Conference on Software Engineering. Piscataway: IEEE Press, 2013: 832–841.

[10] BACCHELLI A, D'AMBROS M, LANZAETAL M. Benchmarking lightweight techniques to link e-Mails and source code[C]//2009 16th Working Conference on Reverse Engineering. Piscataway: IEEE Press, 2009: 205–214.

[11] WANG M, ZOU Y Z, CAO Y K, et al. Searching software knowledge graph with question[C]//The 18th International Conference on Software and Systems Reuse. Cham: Springer, 2019: 115–131.

[12] LIN Z Q, ZOU Y Z, ZHAO J F, et al. Improving software text retrieval using conceptual knowledge in source code[C]// The 32th IEEE/ACM International Conference on Automated Software Engineering. Piscataway: IEEE Press, 2017: 123–133.

[13] 凌春阳, 邹艳珍, 林泽琦, 等. 基于图嵌入的软件项目源代码检索方法[J]. 软件学报, 2019, 30(5): 1481–1497.
LING C Y, ZOU Y Z, LIN Z Q, et al. Approach to searching software source code with graph embedding[J]. Journal of Software, 2019, 30(5): 1481–1497.

[14] FINKEL J R, GRENAGER T, MANNING C. Incorporating non-local information into information extraction systems by Gibbs sampling[C]//The 43rd Annual Meeting on Association for Computational Linguistics. New York: ACM Press, 2005: 363–370.

[15] WOODS W A. Progress in natural language understanding: an application to lunar geology[C]//The National Computer Conference and Exposition. New York: ACM Press, 1973: 441–450.

[16] BEGEL A, KHOO Y P, ZIMMERMANN T. Codebook: discovering and exploiting relationships in software repositories[C]//2010 ACM/IEEE 32nd International Conference on Software Engineering. Piscataway: IEEE Press, 2010: 125–134.

[17] LIN J, LIU Y, GUO J, et al. TiQi: a natural language interface for querying software project data[C]//2017 32nd IEEE/ACM International Conference on Automated Software Engineering. New York: ACM Press, 2017: 973–977.

[18] DONG L, LAPATA M. Language to logical form with neural attention[C]//The 54th Annual Meeting of the Association for Computational Linguistics. [S.l.:s.n.], 2016: 33–43.

[19] JIA R, LIANG P. Data recombination for neural semantic parsing[C]//Association for Computational Linguistics. [S.l.:s.n.], 2016.

[20] YIN P C, NEUBIG G. A syntactic neural model for general-purpose code generation[J]. arXiv preprint, 2017, arXiv: 1704.01696.

[21] XU Y, JONES G J F, WANG B. Query dependent pseudo-relevance feedback based on Wikipedia[C]//The 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press, 2009: 59–66.

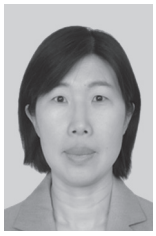
[22] BENDERSKY M, METZLER D, CROFT W B. Parameterized concept weighting in verbose queries[C]//The 34th International ACM SIGIR Conference on Research and development in Information Retrieval. New York: ACM Press, 2011: 605–614.

[23] DALTON J, DIETZ L, ALLAN J. Entity query feature expansion using knowledge base links[C]//The 37th International ACM SIGIR Conference on Research & Development in Information Retrieval. New York: ACM Press, 2014: 365–374.

[24] PAN D Z, ZHANG P, LI J F, et al. Using dempster-shafer's evidence theory for query expansion based on freebase knowledge[C]// Asia Information Retrieval Symposium. Heidelberg: Springer, 2013: 121-132.

[25] GUISADO-GÁMEZ J, PRAT-PÉREZ A, LARRIBA-PEY J L. Structural query expansion via motifs from Wikipedia[C]// The Explore DB'17. New York: ACM Press, 2017.

作者简介



邹艳珍（1976- ），女，博士，北京大学信息科学技术学院副教授，主要研究方向为软件工程、软件复用、知识图谱和智能软件开发等。

王敏（1994- ），男，北京大学信息科学技术学院博士生，主要研究方向为软件工程、软件复用、代码审查和智能化软件开发等。

谢冰（1970- ），男，博士，北京大学教授、信息科学技术学院常务副院长、软件研究所所长，国家杰出青年基金获得者，中国软件行业协会理事，中国计算机学会高级会员，*Chinese Journal of Electromics*编委，入选教育部新世纪优秀人才支持计划、北京市科技新星计划，获得"中创软件人才奖"。主要研究方向为软件工程、计算机理论科学和分布式系统等。

林泽琦（1992- ），男，博士，微软亚洲研究院研究员，主要研究方向为机器学习、智能数据分析、智能开发环境。

收稿日期： 2020-10-20

通信作者： 谢冰，xiebing@pku.edu.cn

基金项目： 国家重点研发计划基金资助项目（No.2016YFB1000800）；国家自然科学基金资助项目（No.61972006）

Foundation Items: The National Key Research and Development Program of China(No.2016YFB1000800), The National Natural Science Foundation of China(No.61972006)