doi:10.11959/j.issn.1000-436x.2015312

面向 ScholarSpace 知识库的关键词查询方法

李和瀚[1]，孟小峰[1]，邹磊[2]
(1. 中国人民大学 信息学院，北京 100872；2. 北京大学 计算机科学技术研究所，北京 100871)

摘 要：知识库中存储着大量关于真实世界中的实体信息及实体之间的关系，随着规模的不断增长，其应用也愈发广泛。另一方面，由于大量互联网用户通过关键词描述问题和查询意图，因此如何让知识库具备更好的关键词查询应答能力，成为了研究的热点。从中文知识库的构建入手，提出了一套完整的面向中文限定领域知识库的关键词检索框架，实现并改进了基于模板的关键词查询转换方法，提出了基于语义的知识库释义和实体索引方法，提高了关键词映射能力。同时在 SPARQL 转换过程中采用了缺失关系索引，提高了转换成功率，提升了能够处理的查询数量。同时在学术空间 ScholarSpace 上对该框架进行了系统实现，取得了良好的应用效果。

关键词：知识库；关键词检索；查询转换；语义相似度
中图分类号：TP311.1 **文献标识码**：A

Keyword search approach for knowledge base in ScholarSpace

LI He-han[1], MENG Xiao-feng[1], ZOU Lei[2]
(1. School of Information, Renmin University of China, Beijing 100872, China;
2. Institute of Computer Science and Technology, Beijing University, Beijing 100871, China)

Abstract: Knowledge bases (KB) store large amount of structured information about the entities and their relationships. As the scale of KBs increased, their application also varied. On the other side, large amount of users describe their question or query intention by submitting keyword queries. Thus enabling KB to answer these keyword queries becomes of crucial importance. A framework from building a Chinese KB to answering keyword search over it was established. A novel approach based on query template to translate the keyword queries into structured queries was proposed. A semantic based paraphrase and index approach to improve the result of query term mapping and an absent predicate index to deal with the predicate absence during the query translation step was proposed. Significant improvement of the ability of translating keyword query to structured query was achieved. Finally the framework and approach was implemented in the ScholarSpace system and get a good performance.

Key words: knowledge base; keyword search; query translation; semantic similarity

1 引言

随着互联网数据的不断增加，可用的知识呈现爆炸性增长，人们从海量非结构化的网络数据中集成实体信息，将实体抽象为概念、挖掘实体和概念之间的关系，进而构建知识库。中文学术信息集成系统学术空间 ScholarSpace 是这方面成果的代表之一。截至目前 ScholarSpace 已经集成超过 100 万作者、200 万篇论文的学术信息，这些学术信息由真实存在的作者、论文、期刊、机构等实体及实体之间的关系构成。同时，规模的增长使知识库的检索成为一个重要课题，ScholarSpace 中传统的通过作者姓名、作者单位、论文题目等离散条件对特定种类的实体进行检索（如图 1 中“作者搜索”、“高级

收稿日期：2015-11-23；**修回日期**：2015-12-10
基金项目：国家自然科学基金资助项目（61379050, 91224008）；国家高技术研究发展计划（“863”计划）基金资助项目（2013AA013204）；中国人民大学科学研究基金资助项目（11XNL010）
Foundation Items: The National Natural Science Foundation of China(61379050, 91224008)；The National High Technology Research and Development Program of China (863 Program) (2013AA013204); Science Research Foundation of Renmin University of China (11XNL010)

搜索”）的方式已经无法满足用户的需求。越来越多的用户希望 ScholarSpace 能够支持通过关键词查询方便快捷地检索学术信息，如通过输入查询“软件学报 2013 云计算”来查找 2013 年发表在《软件学报》上且与云计算相关的论文（图 1 中“语义检索”）。

RDF 知识库采用三元组[1]的形式对实体及关系信息进行存储。W3C 组织推荐的 RDF 查询语言 SPARQL[2]是一种具有严格语法、强大表达能力的查询语言。因此对知识库上的关键词查询[3~5]或自然语言查询[6,7]进行理解并将其转换为结构化的 SPARQL 查询语句，进而在知识库上检索答案是满足上述用户检索需求的合理方式。这两方面均有大量的工作，本文主要关注前者。

本文主要贡献在于构建了结构完整、数据量较大的中文 RDF 学术知识库、使用 RDF 查询引擎对其进行存储和查询。提出了基于词向量的同义词表建立方法，利用同义词表对知识库的释义字典和命名实体索引进行了优化。在关键词查询向 SPARQL 查询的转换过程中，提出缺失关系索引，提高了查询转换成功率。通过实验验证本文方法对知识库的关键词查询应答能力有较大幅度的提升。同时，在 ScholarSpace 中对本文方法进行了系统实现，取得了良好的应用效果。

2　相关工作

随着知识库数量和规模的不断发展，知识库上的关键词检索研究受到越来越多的学者关注。部分研究方向集中在利用关键词直接在 RDF 图上进行搜索，即图查询方法，这类方法通常利用关键词匹配和图算法生成一系列包含查询关键词的子图，最后对生成的子图按一定规则排序，求得 top-k 个结果。如 DING 等[8]提出基于动态规划的斯坦纳树求法，TRAN 等[9]提出基于搜索的最小匹配子图算法，BHALOTIA 等[10]采用对每个关键词节点进行扩展的方法求得包含所有关键词的斯坦纳树。另一个研究方向是将关键词查询转换为结构化查询语句，再通过 RDF 查询引擎进行检索。LEI 等[3]利用人工定义的模板将查询词映射到结构化查询中。SHEKARPOUR 等[4]将查询词映射到实体、类别、属性等语义成分，通过语义成分与结构化查询之间的对应关系生成结构化查询。POUND 等[5]利用词性标注和概率模型计算查询词的语义成分，再同与之对应的模板进行匹配和转换。

无论直接在图上进行检索还是转换为结构化查询语句的方法都会面临关键词歧义的问题，仅通过文字的相似度无法准确地将查询词映射到 RDF 图的节点或边上。POUND 等[11]对查询词的所有可能映射构建图，并通过基于嵌套循环的消歧模型进行评分，得到最佳的映射结果。ZOU 等[7]提出一种数据驱动的方法，利用 ReVerb[12]、Patty 等[13]针对关系短语的数据集构建释义字典，帮助进行消歧。

图 1　ScholarSpace 计算机领域首页

3　系统架构

如图 2 所示，本文从中文学术知识库的构建入手，利用学术空间 ScholarSpace 所集成的学术数据，通过数据转化构建了具有一定规模的 RDF 知识库。进而对其构建实体索引、释义字典以及同义词表。同时选取了一些质量较好的关键词查询，进行人工标注，作为查询转换模板。系统前端由网页前端和微信公众号前端 2 部分组成，负责接收用户输入的查询并连接到后端查询转换核心程序，查询转换程序负责对关键词查询进行序列标注、模板匹配并转换为结构化查询，最后向 RDF 引擎发送结构化查询，接收查询结果并向前端返回。

4　线下部分

线下部分主要解决知识库的构建、知识库索引以及查询模板标注统计 3 个方面的问题。知识库的构建不是本文研究的重点，只做简要介绍。知识库索引是在同义词典的辅助下通过释义字典和实体索引完成。查询模板由人工进行标注。

4.1　学术知识库构建

学术空间 ScholarSpace 是一个面向中文学术领域的学术集成系统，迄今为止已集成了 20 个领域、

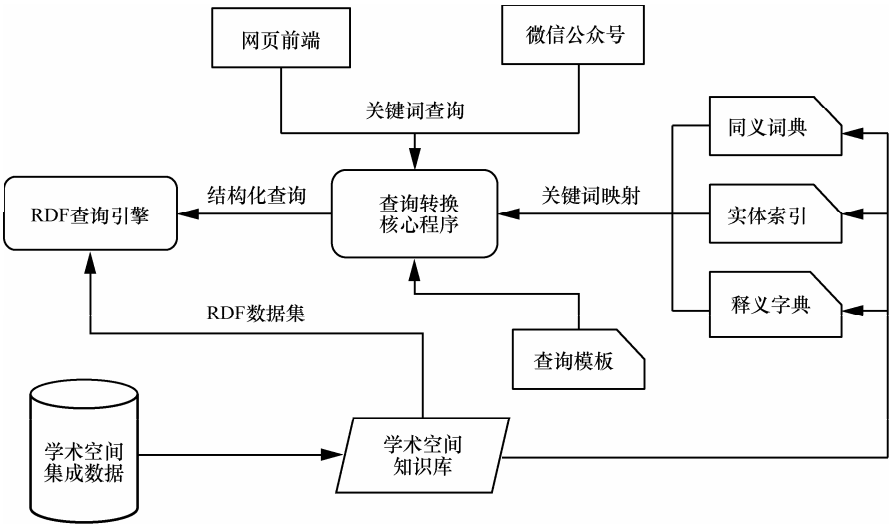


图 2　系统架构

近 200 万篇论文、100 万名作者的数据，并在此基础上对作者进行了重名区分，对各期刊、研究领域按作者、单位进行了统计排名，是目前规模最大中文学术集成系统。本文选取了 ScholarSpace 计算机领域数据中的论文、期刊、单位、作者、论文关键词等重要实体构建学术知识库。

定义 1　定义知识库 $\mathcal{K}=\{\mathbb{C},\mathbb{E},\mathbb{P}\}$，其中，$\mathbb{C}$ 表示知识库中所有实体类别的集合，$\mathbb{E}$ 表示所有实体的集合，$\mathbb{P}$ 表示所有谓语的集合。知识库上的关键词查询 $Q=<q_1,q_2,\cdots,q_n>$，其中 $<q_1,q_2,\cdots,q_n>$ 为组成该查询的关键词序列。

构建的知识库部分示例如图 3 所示，是由 RDF 三元组构成的一张 RDF 数据图。每个三元组(triple)包含主语、谓语和宾语 3 个元组(item)，如<论文 A，作者，李刚>。三元组中的谓语对应 RDF 图上的边，主语和宾语对应 RDF 图上的节点。通过对 ScholarSpace 计算机领域的数据进行转换，本文构建了包含 7 种实体类别、66 种谓语、超过 50 万实体、近 900 万条三元组的学术知识库。

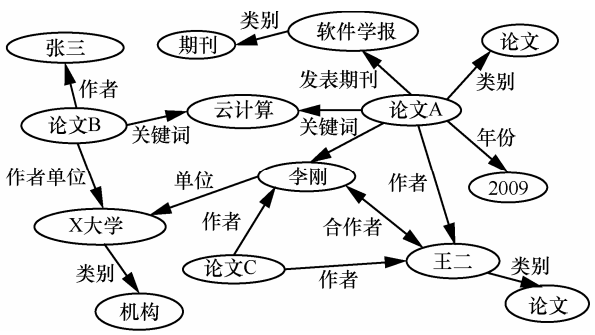


图 3　RDF 数据图示例

4.2　释义字典

释义字典中存储了对知识库中关系(谓语)、类别的自然语言表述，用于对查询中表述关系、类别的关键词进行映射。

定义 2　对于序列 $Q=<q_1,q_2,\cdots,q_n>$，定义映射函数 $M_p:Q\rightarrow 2^{\mathbb{C}\times\mathbb{P}}$，表示将 Q 中的关键词映射到知识库类别和谓语集合的某个子集。

由于对同一种属性或类别可以有多种不同的自然语言表述，如对<作者>这一类别，可以表述为“作者”“学者”、“专家”等不同形式，单纯的人工构建释义字典开销较大。传统的检索方法大多使用已有的、人工构建的同义词库或本体帮助进行消歧，如 WordNet[14]等，中文方面有《同义词林》等。NAKASHOLE 等[13]提出了利用语料库建立释义字典的方法，本文加以改进，首先人工构建小规模释义字典，为每个关系和类别人工标注 1 到 2 个自然语言表述，在此基础上利用 4.5 节中同义词典对其进行规模扩充，使每个属性和类别的自然语言表述形式数量大大增加。

4.3　实体索引

实体索引是知识库中必不可少的部分，借助索引可以快速准确地进行检索词到实体的匹配映射，从而得到正确的查询转换结果。

定义 3　对于关键词序列 $Q=<q_1,q_2,\cdots,q_n>$，定义映射函数 $M_p:Q\rightarrow 2^{\mathbb{E}}$，表示将 Q 中的关键词映射到知识库实体集合的某个子集。如图 4(a)所示，在学术知识库上建立包含所有实体的实体索引，完成查询词到 RDF 图上节点的映射，用于通过实体名称查找实体标识及其类别。

对于释义字典和实体索引，本文采用 Jaccard 系数对结果进行打分。

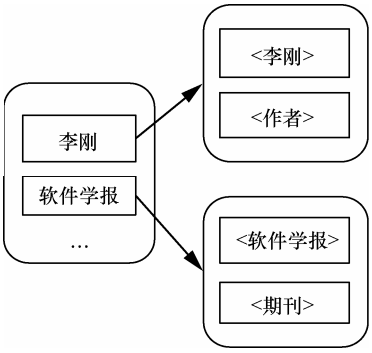
$$Score(q,e)=Jaccard(Name(q),Name(e))=\frac{Name(q)\cap Name(e)}{Name(q)\cup Name(e)}$$

4.4 缺失关系索引

在实体索引的基础上，还对 RDF 图建立了缺失关系索引，用于在 5.3 节介绍的查询转化过程中，帮助处理结构化查询语句谓语缺失的情况，即查询词中不包含谓语信息，需要通过实体标识信息和谓语另一端的实体类别信息反推谓语信息。

定义 4　对于知识库 $\mathcal{K}=\{\mathbb{C},\mathbb{E},\mathbb{P}\}$，定义映射函数 $M_R:\mathbb{E}\times\mathbb{C}\to\mathbb{P}$，表示将实体、类别的组合映射到某个谓语。

如图 4(b)所示，缺失关系索引对实体按标识进行索引，每个索引项下包含 2 个子节点“入边”和“出边”，“入边”表示所有指向该实体的谓语，“出边”表示所有由该实体指向其他实体的谓语，再下一层子节点为谓语另一端的实体类别，最后索引值为谓语标识。



(a) 实体索引

(b) 缺失谓语索引

图 4　对 RDF 图构建的索引示例

例如图 4(b)中实体<李刚>的出边中，另一端实体类别为<机构>的边是<单位>；而其所有入边中，另一端实体类别为<论文>的边是<作者>。建立这样的索引后，就可以根据实体标识和另外一端的类别快速检索出两者之间的谓语。如若实体与某类别之间有多个可能的谓语，则按在 RDF 图上出现的概率排序，取最大者。

4.5 同义词典

在释义字典和实体索引的使用中，如果只利用简单的文字信息来进行匹配效果不佳，很多时候甚至找不到合适的结果。另外由于学术知识库的特殊性，大量的查询会涉及论文关键词，如检索“SVM 论文”时，用户希望得到的不仅仅是关键词中包含 SVM 的论文，而是希望得到所有关键词中包含与 SVM 相近或相关术语的论文，如关键词是“支持向量机”、“支撑向量机”甚至“线性分类”的论文。对于这类关键词模糊匹配的问题，传统的基于《同义词林》等词库的扩展方法显然对于特定领域的知识库无法提供有力支撑，经实验统计学术知识库中超过 97%的论文关键词都不包含在《词林》中。

为此，本文采用基于语料的同义词典构建方法，众所周知语料的获取相对于 WordNet 或《词林》等外部词典要容易很多，且大部分具有一定规模的知识库，如 DBpedia、百度百科等自身就包含大量可用的语料，可以直接用于构建同义词典。

采用 ScholarSpace 计算机领域约 10 万篇论文的摘要和正文快照作为语料，对其进行去噪、分词、去除停用词处理后利用 Word2Vec[15]进行训练。

Word2Vec 为语料中的每个词生成一个低维向量，语义越相近的词之间向量的相似度越高。本文采用 skip-gram 模型，取每个词的前后 10 个词作为词窗，生成的词向量维数 n 取 200，最低词频设为 5，即只为在语料中出现至少 5 次的词生成向量，其他词忽略不计。采用欧氏距离对词向量进行相似度计算

$$Sim(W_1,W_2)=Dis(V_1,V_2)=\sqrt{\sum_{i=1}^{n}(v_{1,i}-v_{2,i})^2}$$

表 1 为生成的同义词典中的部分结果。

表 1　同义词典部分结果

检索词	同义词 1	同义词 2	同义词 3	同义词 4
SVM	支持向量机	支撑向量机	核函数	分类器
数据库	数据库系统	数据库管理系统	DBMS	关系型数据库
知识库	领域知识	知识表示	知识库系统	知识获取

同义词典的生成为释义字典的扩充和实体索

引的优化提供了支持。本文对人工构建的释义字典的每个自然语言表述取同义词典中排名前 5 的同义词，作为新的自然语言表述。在对实体进行索引时，同时对欧氏距离小于一定阈值的同义词进行索引，将所有结果的并集作为最终的索引结果返回。

$$M_e^{'}(q) = M_e(q) \cup M_e(\tau(q))$$

其中，$\tau(q)$ 为 q 在同义词典中的映射集合。

4.6 查询模板标注

本文的查询转换方法是借助查询模板对关键词查询进行转化，模板的标注是线下关键一环，从 ScholarSpace 知识库的查询日志中选取了 150 条质量较好的记录，对其中 50 条进行标注和统计，其余 100 条用于测试。对查询记录的标注分为词性标注、语义成分标注和结构化查询标注。

定义 5 定义查询模板集合 $T = \{< Q^T, PQ^T, CQ^T, SQ^T >\}$，其中，$Q^T$ 为组成查询的关键词序列，PQ^T 为词性标注序列，CQ^T 为语义成分标注序列，SQ^T 为 Q^T 对应的结构化查询语句。

如 Q^T 为<软件学报，X 大学，论文>，词性标注后的结果 PQ^T 为<软件学报:名词，X 大学:机构名，论文:名词>，语义成分标注序列 CQ^T 为<软件学报:实体，X 大学:实体，论文:类别>，对应的结构化查询语句 SQ^T 为

```
Select ?x where
{
    ?x:0   <类别>:0   <论文>:3.          (1)
    ?x:0   <发表期刊>:0   <软件学报>:1.   (2)
    ?x:0   <作者单位>:0   <X 大学>:2.     (3)
}
```

元组后的数字代表该元组对应的关键词在关键词序列 Q^T 中的位置，如第 2 个元组中的谓语<软件学报>对应的 Q^T 中的第 1 个关键词“软件学报”；0 代表该元组在 Q^T 中没有对应关键词。

5 线上部分

线上部分主要由前端模块、查询转换核心程序、RDF 查询引擎 3 个部分组成。前端模块负责查询的接收和结果展示，目前开发了网页前端和微信公众号前端 2 部分。RDF 查询引擎负责从查询转换程序接收结构化查询，在 RDF 图上查找结果并返回。本节主要关注查询转换核心程序的流程和算法，主要包括词性标注、语义成分标注、模板匹配与转换，本节后续部分将一一详细介绍。

5.1 词性标注

词性标注是将关键词查询视为一个关键词序列，利用自然语言处理技术，对序列进行词性标注。

定义 6 对于关键词序列 $Q =< q_1, q_2, \cdots, q_n >$，定义其词性标注序列

$$Q =< q_1 : p_1, q_2 : p_2, \cdots, q_n : p_n >$$

其中，$p_i = \pi(q_i)$，p_i 为 q_i 的词性标签。π 是关键词到词性的映射函数。

利用开源自然语言处理工具 fudanNLP[19]为关键词序列进行词性标注，得到词性标注序列，用于后续的语义成分标注。以序列 Q=<李刚，云计算，文章>为例，标注结果 PQ 为<李刚:人名，云计算:名词，文章:名词>。本文在词性标注的同时对查询中的实体进行识别，为此需要通过 p_i 的词性来推断 q_i 是否是命名实体，当 p_i 是名词、人名、地名、机构名、专有名词等时，q_i 是命名实体的可能性较大，利用 4.3 节中的实体索引及 4.5 节中的同义词典将其映射到知识库中的命名实体上。对上例中的 Q 进行实体识别后的结果 PQ' 为<李刚:命名实体，云计算:命名实体，论文:名词>。

BARR 等[17]研究表明，查询词的词性标注可以为语义成分的推导起到帮助作用，下一节将介绍如何通过词性标注对语义成分进行推导。

5.2 语义成分标注

语义成分指查询词在对应的结构化查询语句中的所属角色。定义语义成分集合 $CS = \{entity, type, relation, attribute, none\}$。其中，*entity* 为实体，*type* 为类别，*relation* 为关系，*attribute* 为属性，*none* 表示不属于任何类别。

从 4.6 节中示例模板的结构化查询语句可知，查询词“软件学报”、“X 大学”对应的语义成分为 *entity*，而“文章”对应 *type*。通过 5.1 节得到的词性标注特征可以推测每个查询词对应的语义成分，这是一个典型的序列标注问题。使用条件随机场 (CRF, conditional random field)[18]对查询进行标注。

条件随机场是一种结合了最大熵模型和隐马尔科夫模型特点的无向图模型，克服了隐马尔科夫模型的独立性假设带来的限制，利用可观测状态对隐含变量的概率进行计算，在序列标注方面有广泛的应用。利用 4.6 节中进行词性标注和语义成分标注的查询作为 CRF 训练数据。

选取每个查询词的特征为：1) 该查询词的文字特征，即查询词本身；2) 该查询词的词性特征；3) 该查询词与前后 2 个查询词词性的组合特征。由此对用户提交的查询进行语义成分标注，得到语义成分标注序列 CQ。

5.3 模板匹配与查询转换

推导出了关键词序列 Q 的语义成分序列 CQ 之后，即可将语义成分序列同与之相符的模板进行匹配，生成候选模板集合。

定义 7 对于一个关键词序列 $Q=<q_1,q_2,\cdots,q_n>$，定义与之匹配的查询模板候选集合

$$S_T=\{<Q^T,AQ^T,CQ^T,SQ^T>\in T$$
$$\text{s.t. } len(Q^T)=len(Q),\sigma(q_i^T)=\sigma(q_i)\}$$

其中，$len(Q^T)$ 和 $len(Q)$ 分别代表模板和查询的关键词序列长度。

如对 CQ =<李刚:*entity*，云计算:*entity*，论文:*entity*>与模板中的记录 CQ^T =<软件学报:*entity*，X 大学:*entity*，论文:*type*>语义成分序列一致，均为<*entity*，*entity*，*type*>，进而将其作为一个匹配模板。对每个 CQ 而言与之匹配的模板可能有多个，本文利用后续查询转换过程中的评分机制对转换结果进行排序。

对于结构化查询语句 SPARQL，查询条件由若干个<主语，谓语，宾语>形式的三元组构成，三元组中每个元组可以分为实体(*entity*)、变量(*variable*)、关系(*relation*)、类型(*type*)和文字(*literal*)。转换过程逐句按主语、宾语、谓语的顺序进行，并对每个元组按照匹配程度进行打分。转换算法如下。

算法 1 关键词序列转换为 SPARQL 查询算法

输入：查询关键词序列 $Q=<q_1,q_2,\cdots,q_n>$；模板结构化查询语句 SQ^T

输出：转换后的结构化查询 SQ

1) FOREACH $tuple^T$ IN SQ^T DO
2) FOREACH $item_i^T$ IN $tuple^T$ DO
3) IF $item_i^T$ is *subject* or *object* DO
4) SWITCH $item_i^T$:
5) CASE *variable*:
6) $Item_i = item_i^T$
7) CASE *entity*:
8) $item_i = M_e(q_{\text{position}(item_i^T)})$
9) CASE *literal*:
10) $item_i = M_p(q_{\text{position}(item_i^T)})$
11) END IF
12) IF $item^T$ is *predicate* DO
13) IF $item^T$ == <类别> DO
14) $item_i = item_i^T$
15) ELIF position($item_i^T$) != 0 DO
16) $item_i = M_p(q_{\text{position}(item_i^T)})$
17) ELIF position($item_i^T$) == 0 DO
18) IF type($item_{i-1}$) == variable DO
19) $item_i = M_R(item_{i+1}, type(item_{i-1}))$
20) ELIF type($item_{i+1}$) == variable DO
21) $item_i = M_R(item_{i-1}, type(item_{i+1}))$
22) END IF
23) END FOR
24) END FOR

首先，对 $tuple^T$ 中的主语和宾语进行转换。若元组为变量，认为 SQ 中的变量元组与 SQ^T 中完全一致，不做任何变化；若元组为实体，根据该元组的位置信息，取 Q 中相应的检索词在 4.3 节实体索引中进行查找，得到该检索词在知识库中对应的实体，其中，position($item_i^T$) 为 $item_i^T$ 在模板中的位置信息。取其索引得分为转化过程中该元组的得分；若元组为文字，则根据该元组的位置信息，直接取 Q 中相应检索词作为生成新元组；对于类别元组，取对应查询词到 4.2 节中介绍的释义字典进行匹配，取字典的匹配得分为转换过程中该元组的得分。

其次，对 $tuple^T$ 中的谓语进行转换。若谓语是<类别>，这是一种较为特殊的关系，表示三元组指明的是某个实体在知识库中所属的类别，对于这种关系不需做任何处理，直接照写；对非<类别>且位置信息不为 0 的关系元组，表示关键词序列 Q 中有直接指明该关系的查询词，根据模板中的位置信息取得该查询词，利用 4.2 节中的释义字典进行查找；对非<类别>且位置信息为 0 的关系元组，该元组在关键词序列 Q 中没有与之对应的关键词，需要利用 4.4 节中的缺失谓语索引进行推导，由谓语一端的实体及另一端的实体类型得到缺失的谓语。

如通过 4.6 节中示例模板对查询 Q=<李刚，云计算，文章>进行转换时，首先考虑示例模板中的第一个三元组(1)的每个元组，对于变量?x 和关系<类别>不需做任何处理，得到部分转换的三元组

?x　　<类别>　　?o

对于宾语，从三元组(1)中可知其对应 Q 中的第 3 个检索词，即“文章”。通过释义字典可知“文章”是类别<论文>的自然语言表述，于是得到转换后的第一个三元组

?x <类别> <论文>

对于模板中的第 2 个三元组(2)，对主语和宾语的处理方法与三元组(1)类似，得到以下缺失三元组

?x <类别> <论文>

?x ?p <李刚>

而谓语既不是<类别>，也不含有位置信息。通过图 4(b)所示的缺失关系索引可知

$$?p = M_R(<\text{李刚}>,<\text{论文}>)=<\text{作者}>$$

同理，可利用式(3)对 Q 进行转换，得到 SQ

?x <类别> <论文>

?x <作者> <李刚>

?x <关键词> <云计算>

对于每个三元组，其得分为主语、谓语、宾语的乘积

$$score(triple)=\sum_{i=1}^{3} score(item_i)$$

转换过后的结构化查询的得分设为所有三元组得分的归一化结果

$$score(SQ)=\frac{1}{n}\sum_{i}^{n} score(triple_i), n=|\{triple^T \in SQ^T\}|$$

不同模板生成的 SPARQL 语句需要按照最终得分进行重排，之后依次向查询引擎提交，直至有效结果数量满足要求或前 N 个 SPARQL 提交完毕。

6 实验结果与分析

本文选取了 ScholarSpace 学术知识库过往查询日志中质量较好的 200 条记录，将其中 100 条作为查询模板进行人工标注，其余 100 条用于进行测试。知识库统计信息如表 2 所示。

表 2 学术知识库统计信息

信息量	值
实体数量	57 万
三元组数量	900 万
谓词数量	66
RDF 图规模	1.2 GB

实验服务器配置为 4 核 Intel Xeon E5506 2.0 GHz CPU，32 GB RAM，操作系统 Ubuntu 10.04 LTS。本文分别对 SPARQL 转换质量和查询引擎返回的最终结果质量进行了评测。

6.1 SPARQL 转换质量评测

把系统生成的 SPARQL 按照与原始问题的符合情况分为 4 个等级：1-完全符合，2-较符合，3-基本符合，4-不符合或无法处理。对于每个问题，评估系统返回前 5 个 SPARQL，从中选取评判等级最高的 SPARQL 作为该问题的评估等级。同时计算得到系统的 *MRR* 值

$$MRR(Q)=\frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{\text{rank}(i)}$$

Q 表示测试问题集合，函数 rank 表示对于第 i 个问题，系统返回的最优 SPARQL 在前 5 个候选 SPARQL 的排序位置。MRR 值越接近 1，表示正确的 SPARQL 排序越靠前。

对于 100 个测试查询，本文方法与文献[5]中方法生成的 SPARQL 评估情况如表 3 所示。

表 3 生成 SPARQL 结果评估情况

方法	评估等级				总计	*MRR*
	1	2	3	4		
本文方法	71	18	7	4	100	0.90
文献[5]方法	43	13	23	21	100	0.74

对所有查询的评估等级进行加权平均，得到平均评估等级，即

$$grade_{\text{avg}}=\frac{1}{|Q|}\sum_{i=1}^{4} i\,|\{q,\text{s.t.}grade(q)=i\}|$$

平均评估等级越接近 1，说明生成 SPARQL 的质量越高。2 个系统的平均评估等级如表 4 所示。

表 4 生成 SPARQL 平均评估等级

方法	平均评估等级
本文方法	1.44
文献[5]方法	2.22

由表 3 和表 4 可以看出，在限定领域下（中文学术知识库），本文方法对于关键词查询的处理能力较文献[5]中方法有了较大提升，无法处理的查询数量明显减少，所生成查询的评估等级也明显优于文献[5]，这主要是由于在 SPARQL 转换过程中采用了缺失谓语索引，避免了因为谓语缺失导致的转换

失败。

6.2　查询结果评测

对于一个查询可能会产生多个 SPARQL，对此本文采取的方法是依次向查询引擎发送这些查询，查询引擎会将满足 SPARQL 查询条件的实体标识返回，直至返回结果超过一定数量或前 N 个 SPARQL 全部提交完毕。

图 5 为向查询引擎发送的最大 SPARQL 数量 N 取不同值时，平均的返回结果数量。

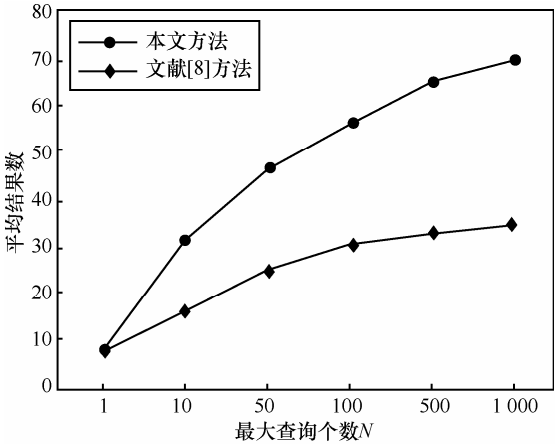


图 5　不同 N 下平均返回结果数

可以看出，本文方法的平均结果数多于文献[5]方法，这是由于本文采用了同义词典对查询词进行了扩充，尤其对于部分包含论文关键词的查询，效果尤其明显。总体上讲，2 种方法的查询结果数量都随着 N 的增大而增大，且随着 N 的增大，查询结果数量的增加越来越不明显，这是由于只有少部分实体索引结果较多的查询会生成数量较多的 SPARQL，如包含论文关键词、论文单位名称的查询，而大部分查询不会。在实际系统中，取 N=150。

在此条件下，对查询结果质量进行评估，验证返回的结果是否满足查询意图，由于对所有查询结果进行验证开销较大，因此，在 100 个测试查询中随机选择 20 个，对 top-K 个查询结果的准确率进行验证。如果所选取的查询是无法处理的查询，则用其他所有可处理查询的平均结果数作为其结果数，并将所有结果视为错误结果。结果如表 5 所示。

表 5　　查询结果正确率

方法	P@1	P@5	P@10
本文方法	0.65	0.60	0.56
文献[5]方法	0.40	0.36	0.34

可以看出，本文方法相较于文献[5]，由于具备更强的查询处理能力，因而在正确率方面也较大提升。出现错误的主要原因是同义词表中可能出现错误的同义词，导致查询结果不符合条件。如检索“SVM 论文”时，由表 1 可知“SVM”的同义词有“分类器”，这就导致检索结果可能是关于其他分类器的论文，而与 SVM 不相关，导致不符合查询意图，但整体上讲利大于弊，基于模板的查询转换方法主要瓶颈出现在寻找匹配模板以及关键词映射 2 个环节，而出现错误匹配、错误转换情况的可能性较小。

7　结束语

本文从中文学术知识库的构建入手，实现了一种基于模板的关键词查询方法，同时通过同义词表、缺失关系索引等方法对其进行了改进，使能够处理的查询数量、查询转换准确率、查询结果覆盖率均得到显著提升，明显加强了对关键词查询的应答能力，并通过实验进行了验证。在此基础上，对该方法在 SchoarSpace 中进行了系统实现，取得了良好的应用效果。

未来的工作方向可以考虑如何提高知识库上实体索引的准确率、减少查询转换过程对于模板的依赖，以及通过算法自动学习查询模板，减少人工标注的开销。

参考文献：

[1] MANOLA F, MILLER E. RDF Premier[S]. W3C Recommendation, 2004.

[2] PRUD E, SEABORNE A. Sparql query language for rdf[EB/OL]. http://www.w3.org/TR/rdf-sparql-query/,2006.

[3] LEI Y, UREN V, MOTTA E. Semsearch: a search engine for the semantic Web[A]. Managing Knowledge in a World of Networks[C]. Springer Berlin Heidelberg, 2006.238-245.

[4] SHEKARPOUR S, AUER S, NGOMO A C N, et al. Keyword-driven sparql query generation leveraging background knowledge[A]. Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference[C]. 2011.203-210.

[5] POUND J, HUDEK A K, ILYAS I F, et al. Interpreting keyword queries over Web knowledge bases[A]. Proceedings of the 21st ACM International Conference on Information and Knowledge Management[C]. ACM, 2012.305-314.

[6] YAHYA M, BERBERICH K, ELBASSUONI S, et al. Natural language questions for the Web of data[A]. Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning[C]. Association for Computational Linguistics, 2012. 379-390.

[7] ZOU L, HUANG R, WANG H, et al. Natural language question an-

swering over rdf: a graph data driven approach[A]. Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data[C]. ACM, 2014. 313-324.

[8] DING B, XU Y J, WANG S, et al. Finding top-*k* min-cost connected trees in databases[A]. Data Engineering, IEEE 23rd International Conference[C]. 2007.836-845.

[9] TRAN T, WANG H, RUDOLPH S, et al. Top-*k* exploration of query candidates for efficient keyword search on graph-shaped (rdf) data[A]. Data Engineering, IEEE 25th International Conference[C]. 2009.405-416.

[10] BHALOTIA G, HULGERI A, NAKHE C, et al. Keyword searching and browsing in databases using BANKS[A]. Data Engineering, Proceedings 18th International Conference[C]. 2002.431-440.

[11] POUND J, ILYAS I F, WEDDELL G. Expressive and flexible access to Web-extracted data: a keyword-based structured query language[A]. Proceedings of the 2010 ACM SIGMOD International Conference on Management of data[C]. 2010.423-434.

[12] FADER A, SODERLAND S, ETZIONI O. Identifying relations for open information extraction[A]. Proceedings of the Conference on Empirical Methods in Natural Language Processing Association for Computational Linguistics[C]. 2011.1535-1545.

[13] NAKASHOLE N, WEIKUM G, SUCHANEK F. PATTY: a taxonomy of relational patterns with semantic types[A]. Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning[C]. 2012.1135-1145.

[14] MILLER G A. WordNet: a lexical database for English[J]. Communications of the ACM, 1995, 38(11): 39-41.

[15] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality[A]. Advances in Neural Information Processing Systems[C]. 2013.3111-3119.

[16] QIU X, ZHANG Q, HUANG X. FudanNLP: a toolkit for chinese natural language processing[A]. ACL Conference System Demonstrations[C]. 2013.49-54.

[17] BARR C, JONES R, REGELSON M. The linguistic structure of English Web-search queries[A]. Proceedings of the Conference on Empirical Methods in Natural Language Processing Association for Computational Linguistics[C]. 2008.1021-1030.

[18] SHA F, PEREIRA F. Shallow parsing with conditional random fields[A]. Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1 Association for Computational Linguistics[C]. 2003.213-220.

作者简介：

李和瀚（1990-），男，四川成都人，中国人民大学硕士生，主要研究方向为知识图谱上的自然语言查询、本体构建。

孟小峰（1964-），男，河北邯郸人，中国人民大学教授、博士生导师，主要研究方向为 Web 数据管理、移动数据管理、大数据。

邹磊（1980-），男，安徽安庆人，北京大学副教授，主要研究方向为图数据管理、基于 RDF 的知识库管理。