doi:10.3772/j.issn.2095-915x.2022.03.003

产业政策知识图谱的自动化构建

揣子昂[1] 耿骞[1,3] 潘慧瑶[1] 靳健[1,2]

1. 北京师范大学政府管理学院信息管理系 北京 100875；
2. 富媒体数字出版内容组织与知识服务重点实验室 北京 100038；
3. 北京师范大学珠海校区政府治理研究中心 珠海 519085

开放科学（资源服务）标识码（OSID）

摘要：[目的 / 意义] 为方便民众和政府工作人员快速了解产业政策中的重要内容，本研究提出了一套自动化的产业政策知识图谱构建框架，用于梳理有关政策文本。[方法 / 过程] 具体地，考虑到中文语料的缺乏，本研究利用基于句法分析的三元组抽取模型 DSNF 从产业政策中抽取政策实体和关系，并结合政策分析场景的特点对结果进行调整。由于原始三元组的表达较为分散，本研究利用三元组表示模型 TransP 和层次聚类模型 BIRCH 对原始三元组进行表示和归并。[局限] 当前模型的性能尚需要在更大规模的数据集上进行检验，并与已有的方法进行比较。[结果 / 结论] 本研究调用并调整了一系列模型用于解决产业政策知识图谱构建过程中的问题，并探索了基于知识图谱的政策分析模式，具有重要的理论和实践意义。

关键词：产业政策；知识图谱；开放域三元组抽取；三元组表示；图数据库

中图分类号：G35；TP391

Automatic Construction of Knowledge Graph for Industrial Policies

CHUAI Ziang[1] GENG Qian[1,3] PAN Huiyao[1] JIN Jian[1,2]

1. Department of Information Management, School of Government, Beijing Normal University, Beijing 100875, China;
2. The Key Laboratory of Rich-Media Knowledge Organization and Service of Digital Publishing Content, Beijing 100036, China;
3. Center for Governance Studies, Beijing Normal University at Zhuhai, Zhuhai 519085, China

Abstract: [Objective/Significance] To facilitate public and government servants to understand the key points of industrial policies efficiently, an automatic framework for knowledge graph construction is proposed, which aims to improve the readability of the policy. [Methods/Process] Specifically, considering the lack of Chinese corpus, an unsupervised open triple extraction model,

基金项目 富媒体数字出版内容组织与知识服务重点实验室开放基金“产业政策图谱构建研究”(ZD2021-11/04)；国家社会科学基金重点项目“面向集成管理的政府数据组织与传递机制研究”（19ATQ005）。

作者简介 揣子昂（1997-），硕士研究生，研究方向为文本挖掘与语义链接；耿骞（1965-），博士，教授，研究方向为政府数据治理；潘慧瑶（1999-），硕士研究生，研究方向为知识组织与文本挖掘；靳健（1982-），博士，副教授，研究方向为用户需求分析及产品评论挖掘，E-mail: jinjian.jay@bnu.edu.cn。

引用格式 揣子昂 , 耿骞 , 潘慧瑶 , 等 . 产业政策知识图谱的自动化构建 [J]. 情报工程 , 2022, 8(3): 28-51.

named as DSNF which is based on syntactic analysis, is applied to extract entities and relations from industrial policy texts, and the results are further enriched according to the policy analysis scenario. Since the expression of original triples is diversified, the triple representation model TransP and the hierarchical clustering model BIRCH are used to summarize the original triples. Finally, the extracted triples are provided to a graph database, Neo4j, based on which a series of functions are provided for users, including visualization, retrieval, etc. [Limitations] The performance of the proposed model needs to be further evaluated on larger datasets and compared with baselines. [Results/Conclusions] State-of-the-art models are applied and modified to build knowledge graph for industrial policies automatically, and a novel policy analysis method based on knowledge graph is explored, which is of important theoretical and practical implications.

Keywords: Industrial policy; knowledge graph; open triple extraction; triple representation; graph

引言

我国产业经济的高速发展离不开政策环境的滋养。在新一轮信息技术浪潮下，各级政府积极推动基于互联网 +、区块链等信息技术的产业政策公开工作。公开的政策文本能够直接体现政府部门的决策过程，是民众可获取、可追溯、可信赖的官方文字记录，能够帮助民众了解政府部门对各类产业经济发展所持态度，从而实现更加紧密的政民联系。然而，政策文本通常具有数量多、篇幅长、不易读等特点。这些无疑不利于公众了解政策动向，明确自己的办事诉求；抑或是政府工作人员解读、调整当前政策。基于此，有必要对公开的海量政策文本进行自动化梳理，帮助民众和政府工作人员快速了解产业政策中的重要内容，为产业经济发展赋能。

知识图谱的概念是由谷歌公司于 2012 年提出，且一系列研究表明知识图谱有助于重新组织梳理政策内容[1-5]。与一般的分析方法不同，政策知识图谱强调利用自然语言处理技术，在文本挖掘的基础上，明确政策实体之间的关系，在提炼、萃取、关联和整合的基础上构建知识图谱，并基于此进行智能分析和决策[6,7]。一般地，构建政策图谱的步骤主要包括从海量政策文本中抽取政策实体并明确实体之间的关系，构成关系三元组。在图谱构建完成后，可搭建前端界面，并将可视化、知识图谱检索等技术嵌入其中，从而呈现政策文本中包含的语义知识，完成与用户的交互。

在知识图谱构建的相关工作中，研究人员利用分类、序列标注等技术从非结构化文本中抽取关系三元组。随后，考虑到从原文中直接抽取出的实体和关系比较分散，仍需调用三元组的表示模型为其生成嵌入向量，并基于此对关系三元组进行聚类和归并。在得到关系三元组后，已有的研究通常利用 Protégé 软件从模式层构建本体，并将抽取到的实体和关系插入到相应的类别下，从而完成知识图谱的构建。然而，Protégé 软件过于专业，且不具备知识图谱展示、响应用户查询等功能，对于一般用户而言不具有易用性。在大数据背景下，研究人员常利用可视化技术，帮助用户快速获取所需要的信息[9,10]，有效降低 Protégé 软件的使用门槛，有助于用户进行决策。此外，也有部分学者对于知识图谱的检索进行了探讨。例如，许德山等总结了知

doi:10.3772/j.issn.2095-915x.2022.03.003

识图谱可视化检索的主要方法，并就可视化平台的功能设计进行了详细阐述[11]。由丽萍等利用查询工具 Jena 的 RDF 和 OWL API 接口对知识图谱进行解析和存储，从而实现查询的智能化和个性化[12]。而目前关于领域知识图谱构建的研究大多基于 Protégé 等软件，其呈现方式和检索效果仍有较大改善空间。

相比之下，在政策分析领域，基于知识图谱的分析方法发展得较为缓慢。一方面，很大一部分研究以研究政策的文献作为研究对象，从文献计量学的角度基于文献之间的共现关系构建知识图谱，并利用 CiteSpace 等软件对图谱进行可视化，同时完成检索、推理、图谱演化等功能[14-17]。然而，研究文献作为二次信息具有一定的时滞性，基于文献的政策分析不利于民众了解实时的政策动向；此外，此类研究所构建的图谱中通常只包含引证、共现等计量学方面的关系，而在政策文本中，实体之间的关系种类是多样且不固定的。基于此，部分学者从政策文本自身出发，从中抽取政策实体、关系以及属性。然而，已有的研究通常依赖于文本字面的模式匹配以及预设的抽取类别[18,19]。这样的做法意味着较高的人力和时间成本，且对于不同产业的政策文本不具备通用性，忽视了政策文本中实体、关系类别的多样性。

针对已有研究的局限性，本研究以养老产业政策为例，利用开放域的产业政策三元组抽取模型，从政策文本中抽取实体和关系。该模型不依赖人工制定的模板或预设的关系类型进行抽取。此外，本研究利用表示模型为抽取到的原始三元组生成表示向量，并利用聚类模型对其进行归并。在完成政策图谱构建后，本研究利用图数据库为构建的知识图谱实现可视化展示以及检索功能，注重提升易用性和交互性。在调用模型的过程中，本研究重点关注产业政策文本的特殊性，并基于此对模型进行调整，使之更适用于当前场景。

1 国内外研究现状

1.1 关系三元组抽取

首先，已有的关系三元组抽取模型按照抽取步骤可大致分为联合抽取模型、基于 Pipeline 的抽取模型。基于 Pipeline 的方法指先抽取实体，再抽取关系；而联合抽取模型则是同时从文本中获取实体和关系。与联合抽取相比，基于 Pipeline 的方法易于实现，前后两个模型的灵活性高，且不需要同时标注有实体和关系的数据集。然而，在实际应用中，实体抽取中的错误会影响下一步关系抽取的性能，且无法建模两个任务之间的内在联系。

此外，关系三元组的抽取可分为限定域和开放域的抽取模型。其中限定域关系抽取是指，根据固定的模版，从非结构化文本中识别出实体对以及其间关系，从而构成关系三元组。其优势在于可以抽取到原文字面上未出现的关系，其缺陷在于无法抽取出预定类别之外的实体或关系。此外，限定域的抽取模型通常需要人工标注一定数量特定领域的少量关系和对应实体作为训练集，费时费力，且对于新的领域不具有通用性。基于此，Yates 等[8]率先提出了开放域关系抽取的概念，即不需要限定关系类型，从非结构化文本中抽取关系三元组，其中关系的指代词是在文本中存在的。开放域关系抽取

模型通常具有以下特点：（1）模型能够自动进行三元组抽取，而不依赖预定的关系类型；（2）对于非同源语料，模型具有较强的通用性；（3）模型不需要针对个别领域进行训练，从而节省了数据标注和训练计算的成本，在大数据背景下具有较高的效率。综上，本小节主要总结开放域的三元组联合抽取模型。

起初研究人员会通过手工定义规则的方式抽取三元组。Fader 等基于词性标注和句法分析对句子中的动词短语进行限制，提出了 ReVerb 模型抽取关系短语，并为其匹配头尾实体 [22]。类似地，PredPatt 模型利用通用依存句法分析 [23] 找到关系谓词和实体集合中第一个词项之间的依存路径，在此基础上构建有向图来抽取三元组 [24]。Exemplar 在利用人工定义的句法规则抽取到关系和实体后，进一步基于语义角色标注将实体分为主、副实体 [25]。考虑到一句话中可能同时包括多个关系三元组，Kraken 模型基于句法分析搜集句子中每个关系短语对应的所有实体来构建三元组，从而避免抽取不完全的情况 [26]。

虽然自监督抽取模型通常也使用句法特征，但是与基于规则的模型不同，自监督模型会首先通过启发式的方式构建训练数据集，并依此训练出有监督的抽取模型。Yates 等提出 TextRunner 模型，利用深层次语法解析器从少量语料中自动抽取三元组，并将其中置信度高的作为正例、置信度低的作为负例作为训练数据集，并基于此训练了朴素贝叶斯分类器作为最终的三元组抽取模型 [8]。类似地，Mausam 等利用基于规则的抽取模型 REVERB 先从语料中抽出关系三元组，并利用其构建训练数据 [21]。在此基础上，Wu 等放弃使用模型构造训练集，改为利用维基百科信息框中的键值对作为训练数据 [20]。相比 TextRunner 的标注方式，维基百科提供的训练语料质量更高，数据量更大，因此训练所得模型的性能也优于 TextRunner。

然而以上模型过分依赖人工定义的特征，在不同任务、领域和场景下的通用性较差。为了解决这一问题，研究人员使用神经网络自动获取特征，完成三元组抽取。例如，RNNOie 模型将开放域的关系抽取视为序列标注问题，利用标准的 BIO 模式对句子中的词项进行标注 [27]。其模型结构整洁，包括嵌入层、基于 bi-LSTM 的特征提取层和 softmax 标注层。在此基础上，HNN4ORT 在神经网络中加入了局部注意力机制和抽取全局信息的卷积层，同时使用有序的 LSTM 结构 [29]，使之更适合开放域关系抽取 [28]。和 RNNOie 一样，HNN4ORT 也借助神经网络将开放域的关系抽取问题转化为序列标注问题，但此类方法无法抽取隐式关系。基于此，微软公司提出 Neural OIE 模型，将编码器 - 解码器架构引入到开放域关系抽取任务中，以期能够生成原文中未出现的隐式关系 [30]。

虽然自监督模型和基于神经网络的抽取模型是当前学界在相关领域的研究热点，但是考虑到产业政策分析场景下缺乏中文语料，难以训练出高性能的三元组抽取模型，且应用上述模型通常对硬件有较高要求。因此，本研究基于 Jia 等针对中文句法特征提出的无监督开放域抽取模型从产业政策文本中获取三元组，并针对政策文本中实体、关系的特点对模型进行调整 [31]。

doi:10.3772/j.issn.2095-915x.2022.03.003

1.2 关系三元组表示

由于从文本中直接抽取到的三元组中实体和关系较为分散，因此需要在构建图谱前对其进行进一步的聚类和归并。当前已有很多学者探索过如何将关系型数据表示为低维向量[32]。例如，RESCAL 是一种基于张量分解的表示方法，其损失函数定义为 $f(A, R_k)+g(A, R_k)$。其中 f 为损失；g 为正则化项，以防止过拟合；A 为由实体的表示向量组成的矩阵；R_k 为表示实体之间关系的矩阵[33]。为了减少 RESCAL 中的参数，Bordes 等提出了 TransE 方法，其得分函数定义为 $f_r(h, t)=\|h+r-t\|_2^2$。其中，h, r, t 分别表示头实体、关系、尾实体对应的向量[34]。直观上，如果该三元组确实存在，$h+r$ 应该接近 t。虽然 TransE 的假设很简单，但是其在实际应用中的表现非常好，且衍生出了很多变体。为了使 TransE 能够适应 1 对 N，N 对 1，N 对 N 的使用情景，TransH 将头实体和尾实体投影到超平面上进行学习，其得分函数为 $f_r(h, t)=\|h_\perp+r-t_\perp\|_2^2$。其中 $h_\perp$，$t_\perp$ 表示头实体和尾实体的投影[35]。基于此，TransR/CTransR 对投影的超平面做了进一步假设：投影的超平面应与关系 r 相关[36]，其得分函数可写作 $f_r(h, t)=\|M_rh+r-M_rt\|_2^2$。TransD 进一步假设头尾实体应该被投影到不同的超平面上，即 $f_r(h, t)=\|M_{rh}h+r-M_{rt}t\|_2^2$，其中 $M_{rh}=r_ph_p^T+1$，$M_{rt}=r_pt_p^T+1$。相比于 TransR/CTransR，TransD 所使用的投影矩阵同时基于关系 r 以及头尾实体 h 和 t。此外，TransD 中待训练参数较少，且不需要进行矩阵乘法，使得其更适合用于大规模数据集[37]。

然而，以上表示方法都没有考虑到实体和关系在政策文本中的语义信息。具体地，以上方法将随机向量作为初始向量进行学习，这样会丢失原始语境中的语义信息。此外，以上模型在学习过程中对向量的变化没有任何约束，这样会导致实体和关系的嵌入向量在学习过程中远离其原始含义，学习得到的表示向量不利于后续对相关的关系和实体进行归并。

1.3 政策知识图谱的构建与分析

当前学界中基于知识图谱对政策进行分析的研究还相对较少，且多数从文献计量的角度展开。此类研究以政策分析文献为研究对象，利用文献计量工具考察文献之间引证关系以及关键词、作者之间的共现关系等，并最终基于知识图谱探究政策研究的趋势。例如，罗哲等从 CSSCI 数据库中收集了 579 篇人才政策的研究文献，并利用 CiteSpace 构建知识图谱，其中包括对文献集合中的作者与机构共现关系、作者共被引关系等[14]。此外，马续补等基于 VOSviewer 关注了文献集合中作者的合作关系以及关键词的共现关系等[15]。类似的研究还包括吴宾等[16]以及赵绘存等[17]对于养老以及科技政策的分析。在基于知识图谱的政策分析上，胡春阳等针对不同时期的中外区域政策进行了对比研究，并结合历史重大事件给出解释[13]。

然而，基于文献计量的方法所构建的知识图谱中只包含固定的几类实体和关系，如作者、关键词、共现、共被引等。一方面，此类实体和关系与民众对政策的关切不符；另一方面，政策中包含的实体和关系类型远不止上述几种，而且不同领域的政策也不尽相同。此外，政府部门时常会根据实际情况对政策作出调整，而政策研究文献作为二次信息具有一定的时滞性，

以其作为研究对象构造的知识图谱很难为民众展示实时的政策动向。

基于此，部分研究人员开始直接从政策文本中抽取政策实体、关系以及属性，以构建政策知识图谱并基于其进行政策分析。例如，霍朝光等利用关键词抽取技术从开放公文集合中抽取政策实体，并基于人工设置的模板抽取实体之间的关系，构建了新冠肺炎防疫政策知识图谱。随后，在时间上，该工作探究了不同防疫阶段政策的差异；在空间上，该工作比较了我国不同地区防疫政策的区别[19]。然而，在抽取实体和关系时，霍朝光等使用了基于文本字面模式匹配的抽取模型，且涉及的特征多针对防疫政策，这使得其模型通用性较差，难以迁移至其他政务场景。为解决这一问题，张雨等不再依赖基于模板的抽取模型，改为使用 bi-LSTM 模型，自动提取文本特征，从政策文本中抽取三元组[18]。然而，该模型为限定域的关系抽取模型，无法抽取到训练数据集以外的实体、关系类型，不利于应对数量持续增长、模式时常变化的政策文本。

1.4 小结

一方面，关系三元组的抽取模型可按照抽取步骤可大致分为联合抽取模型、基于 Pipeline 的抽取模型。其中，基于 Pipeline 的模型需要先从文本中抽取实体，并基于此抽取关系；而联合抽取模型能够同时抽取实体和关系。由于其能够捕捉到实体、关系抽取任务之间潜在的联系，且关系抽取的效果不会受到实体抽取效果的影响，联合抽取模型在学界和业界受到了广泛关注。另一方面，关系三元组的抽取可分为限定域和开放域的抽取模型。相比限定域抽取模型，开放域抽取模型能够实现自动抽取，无需依赖指定类型的实体和关系，且无需针对个别领域进行训练，对于非同源语料具有较强的通用性。开放域抽取模型可大致分为基于规则的抽取模型、自监督抽取模型和基于神经网络的抽取模型。其中，基于规则的模型借助词性标注、句法分析结果手工定义规则，直接从原文中抽取关系三元组；自监督模型则根据手工定义的规则构造训练数据集，训练有监督的模型进行抽取；基于神经网络的抽取模型将抽取问题转化为序列标注问题，此类模型通常需要大量高质量的训练数据。考虑到产业政策分析场景下缺乏中文语料，难以训练出高性能的有监督三元组抽取模型，本研究重点关注基于规则的开放域三元组抽取模型。

由于从原始文本中抽取到的三元组表达较为分散，需要对其进行归并。根据已有的研究，恰当的表示学习模型能够将语义相近的词语表示为相似的向量，使其更容易被分类、聚类模型所区分。基于此，本研究考虑调用表示学习模型将实体及其间关系表示为低维向量，便于后续归并。然而，已有的表示模型在产业政策三元组归并任务中具有以下三点局限。首先，已有的表示模型通常以随机向量作为实体和关系的初始化向量，导致最终所得的表示向量中不包含实体及其所在语境的语义信息。第二，已有的表示模型对于表示向量没有限制，即向量可以在整个空间中自由变动，导致经过数轮训练后的向量会远离其自身的语义，与表示学习的初衷相悖。第三，当前表示模型的演化趋势主要关注“实体和关系不应被投影到同一个

doi:10.3772/j.issn.2095-915x.2022.03.003

向量空间”，而语义信息以及对向量的相关限制还鲜有提及。

当前学界利用知识图谱进行政策分析的研究还较为少见。其中大多数研究从文献计量学的角度出发，以政策分析的文献为研究对象，利用文献分析工具考察其间的关键词分布、引证关系、作者和机构的共现情况等。然而，此类知识图谱中包含实体、关系类型与民众对于产业政策的关注不符；且政策分析文献作为二次信息具有一定的时滞性，使得知识图谱无法向民众呈现实时的政策动向。此外，也有部分研究人员直接从政策文本中抽取三元组构建知识图谱，并基于此进行政策分析。然而，已有的研究所使用的模型通用性较差，依赖于特定的字面匹配模板或指定类型的训练数据，难以迁移至其它政策分析场景，且不利于应对数量持续增长、内容实时更新的政策文本。

2 产业政策知识图谱的构建

为方便民众和政府工作人员快速了解产业政策内容，本研究提出了一套自动化产业政策知识图谱构建框架，如图 1 所示。对于海量政策文本，该框架首先利用基于中文句法特征的 DSNF 模型从中抽取关系三元组，并根据政策文本的特征为实体补齐修饰成分。考虑到原始文本中三元组较为分散，本研究利用三元组表示模型 TransP 和层次聚类模型 BIRCH 对其进行归并。随后，本研究将三元组导入到图数据库 Neo4j 中，并基于其中嵌入的可视化和检索等功能完成与用户的交互。

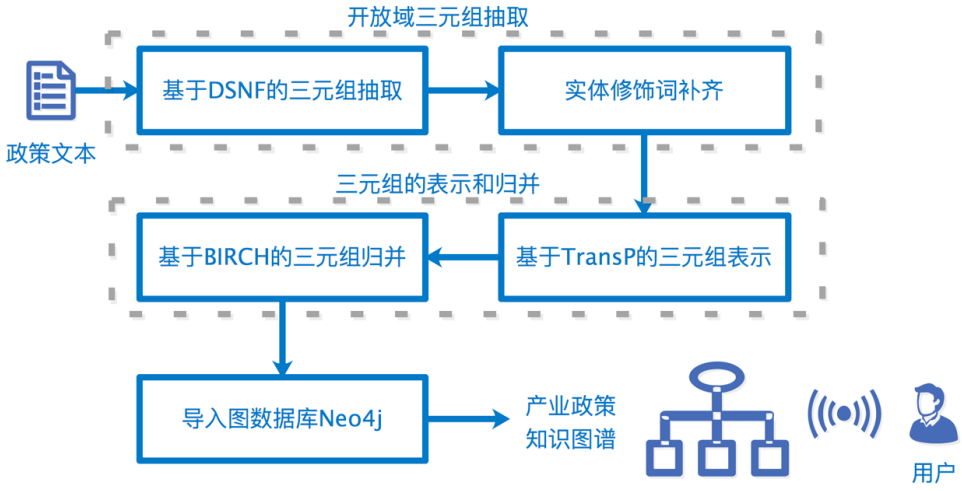


图 1 产业政策知识图谱的自动化构建

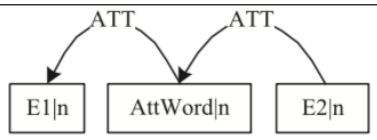
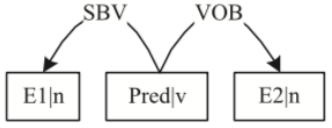
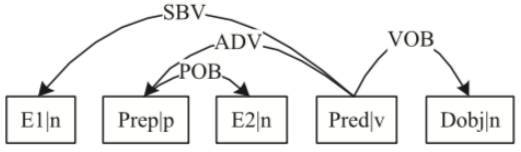
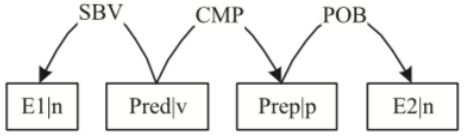
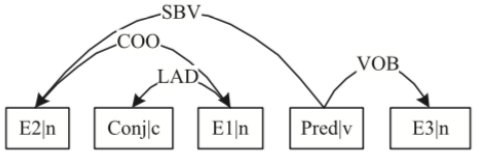
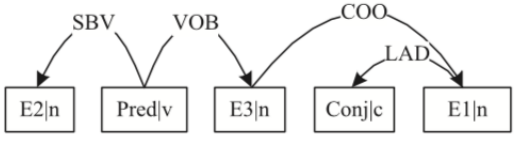
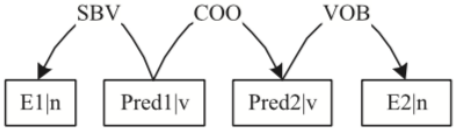
2.1 开放域三元组抽取

2.1.1 基于句法分析的抽取模型——DSNF

由于中文开放三元组抽取语料的缺乏以及中文语言学上的特点，Jia 等基于中文句法分析提出了无监督开放域三元组抽取模型 DSNF（Dependency Semantic Normal Forms）[31]。与字面特征相比，依存关系能捕获文本的语义和句法层次的特征，因此更适用于关系抽取任务。在句法依存树中，实体对通常以名词短语的形式出现，其间的依存路径恰好包含了两者之间的关系。

Jia 等总结了中文三元组中常见的句法结构，包括主语 - 谓语、谓语 - 宾语、介词 - 宾语、并列、修饰等，并将以上结构映射到句法依存树中，得到一系列三元组抽取模板。从句法分析的角度讲，抽取模板可视作词语、词性标签、依存路径的组合，且可大致被分为三类：修饰、动词以及并列关系模板，如表 1 所示。每类模板可根据具体场景扩充。

表 1 **DSNF** 开放域三元组抽取模板

序号	模板表示	关系三元组
1	 [E1-ATT-AttWord,AttWord-ATT-E2]	(E1,AttWord{1,2}+,E2)
2	 [E1-SBV-Pred,E2-VOB-Pred]	(E1,Pred,E2)
3	 [E1-SBV-Pred,Dobj-VOB-Pred,E2-POB-Prep,Prep-ADV-Pred]	(E1,Pred-[Dobj]?+,E2)
4	 [E1-SBV-Pred,E2-POB-Prep,Prep-CMP-Pred]	(E1,Pred-Prep,E2)
5	 [Conj-LAD-E1,E1-COO-E2,E2-SBV-Pred,E3-VOB-Pred]	(E2,Pred,E3) (E1,Pred,E3)
6	 [E2-SBV-Pred,Conj-LAD-E1,E1-COO-E3,E3-VOB-Pred]	(E2,Pred,E3) (E2,Pred,E1)
7	 [E1-SBV-Pred1,E2-VOB-Pred2,Pred2-COO-Pred1]	(E1,Pred2,E2)

doi:10.3772/j.issn.2095-915x.2022.03.003

修饰类模板主要关注语境中对实体对的修饰元素，并将句法依存关系中的支配词（head word）作为实体，从修饰词（attributive word）中抽取关系。在实际应用中，构成修饰词的范围很广，包括专有名词、形容词、数词等。修饰词和支配词之间的句法关系标签通常为 ATT 或 RAD。例如，在句子“工会主席张三对退休老干部进行了慰问”中，“工会”和“主席”都是支配词“张三”的修饰词。同时，“主席”同样表达了实体“工会”和“张三”之间的语义关系。基于此，DSNF 可从中抽取到关系三元组（工会，主席，张三）。值得注意的是，支配词可能拥有多层次的修饰词，但从实际考虑，DSNF 仅考虑 2-3 个修饰词构成的关系，如表 1 中模板 1 所示。

动词类模板重点关注实体对之间的谓词短语，并从中抽取关系。一般地，实体对中的两个实体分别作为谓词的主语和宾语，其与谓词之间的句法关系标签分别为 SBV 和 VOB，如表 1 中模板 2 所示。以句子“养老保险公司开展养老保障管理业务”为例，其中“养老保险公司”作为句子主语依存于谓语“开展”，“业务”则为宾语。因此，DSNF 将从中抽取到三元组（养老保险公司，开展，业务）。而在实际应用中，实体与谓词之间的关系通常会有多种变体。例如，宾语可能不出现在谓词后面，而是以介宾短语的形式在谓词之前修饰谓词。此时，介词与谓词之间的句法关系标签为 ADV，而作为宾语的实体则以 POB 标签依赖于介词，如表 1 中模板 3 所示。例如，在句子“中国保监会对养老保险公司经营行为进行规范”中，“中国保监会”作为句子主语直接依存于谓语“进行”，其句法标签为 SBV。同时，“经营行为”则以句法标签 POB 依赖于谓语，作为前置宾语。因此，模型应抽取三元组（中国保监会，进行规范，经营行为）。同理，介宾关系同样可以位于谓词短语后面，如表 1 中模板 4 所示。对于句子“养老保险的滞纳金列支于自有资金中”而言，其主语为“滞纳金”，而宾语“资金”则通过介词“于”以标签 POB 依赖于谓语“列支”。因此，DSNF 将抽取三元组（滞纳金，列支于，资金）。

并列类模板的目的是找到在句法树中地位平等的实体活动词短语，称之为并列关系，而其中一个所拥有的关系三元组，理应同样适用于另一个，即二者可互换。实体的并列关系在中文中通常会通过逗号或连词来表示。在句法分析中，连词会以标签 LAD 依赖于其中一个实体，同时该实体以标签 COO 依赖于另一个实体，如表 1 中模板 5 和 6 所示。例如，DSNF 可以基于模板 2 从句子“劳动和社会保障部、财政部和司法部印发了《通知》”中抽取到三元组（司法部，印发，通知）；而“劳动和社会保障部”、“财政部”均与“司法部”呈并列关系，即在句法树中以标签 COO 依赖于“司法部”，因此 DSNF 同样可以抽取到三元组（劳动和社会保障部，印发，通知）和（财政部，印发，通知）。类似地，动词的并列关系主要用于描述同一个实体实施的不同动作，在句子中通常并列分布，如表 1 中模板 7 所示。例如句子“由养老保险公司建立个人账户，并核算应缴费用”，根据模板 2 可从中抽取三元组（养老保险公司，建立，个人账户）。同时，“核算”与“建立”在句法分析中呈并列关系，因此同样可以抽取到三元组（养老保险公司，核算，费用）。

2.1.2 补齐实体修饰词

在实际应用中，DSNF 能够较为准确地抽

取到文本中的关系三元组。然而，产业政策的语料中通常包含大量专有名词，包括机构名称、政策标题等。此类专有名词一般由多个修饰词和一个支配词组成，因此很难在预处理中的分词过程中完整保留下来。加之，专有名词的种类繁多且不确定，因而无法通过给分词模型添加用户词典的方式避免其被分割。这就会导致抽取到的政策实体丢失重要的定语，即实体的修饰词，从而造成民众理解上的偏差和歧义。例如，对于句子“为了规范保险公司养老保险业务”，其分词结果可表示为“为了 / 规范 / 保险公司 / 养老 / 保险 / 业务”，因而“业务”会被作为实体抽取出来，而无法抽取到“养老保险业务”。

为解决以上问题，本研究基于句法分析的结果对 DSNF 抽取到的实体的定语进行补充。具体地，本研究会同时返回句子中所有以 ATT 标签依赖于实体的词语，词语的顺序与原文中保持一致。仍以上文中的句子为例，句中词语“养老”、“保险”均依赖于 DSNF 抽取到的实体“业务”，且句法分析标签为 ATT，遂将“养老保险业务”视为整体返回。

2.2 三元组的表示和聚类

从原始文本中抽取到的实体和关系过于分散，不适合直接用来构建知识图谱。即语义上相似或相同的实体和关系在不同的政策文本中可能具有不同的表达形式，直接将其加入到政策知识图谱中会造成冗余，与知识图谱精炼、易读的特点相违背。因此，本研究利用三元组的表示和聚类模型对原始文本中的三元组进行归并。

2.2.1 政策三元组的模式化设定

在对原始政策三元组进行表示和归并前，本研究针对产业政策的特点，为政策实体和关系从语义层面设定模板。对于模板以外的三元组，本研究认为其与产业政策的联系不紧，不予加入到产业政策知识图谱中。具体如图 2 所示。

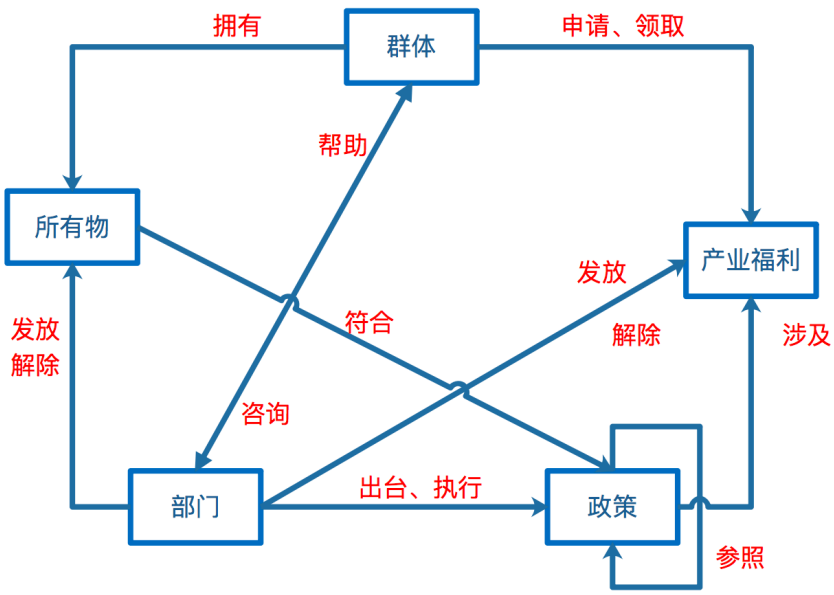


图 2 产业政策三元组模式化表示

doi:10.3772/j.issn.2095-915x.2022.03.003

具体地，本研究定义了三种通用政策分析的实体类型，包括部门、政策、群体。同时，考虑到民众需要通过了解产业政策内容来明确自身的办事需求，本研究针对产业政策特点以及民众潜在的办事需求另外定义了两个实体类型，包括产业福利、个人所有物。具体如下：

部门：指与产业相关的各级政府部门，如中共中央宣传部、司法部、社保经办机构等

政策：指各级政府部门颁布的产业政策名称，如通知、办法、规定等；

群体：指产业政策文本中涉及到的人员，如离休人员、城镇职工、参保人员等；

产业福利：指当前产业能够为民众带来的福利、权益，如养老金、补贴、保险等；

个人所有物：指民众自身拥有的权利或资本，如工资、余额、个人账户等。

在明确了实体类型后，本研究进一步定义了各类实体之间的关系类型，具体如下：

出台：指一类以部门为头实体、政策为尾实体的关系，用于表示政策来源，如出台、颁布、印发等；

执行：指一类以部门为头实体、政策为尾实体的关系，用于表示政策去向，如执行、遵照、利用等；

发放：指一类以部门为头实体、个人所有物或产业福利为尾实体的关系，用于表示个人所有物和产业福利的来源，如发放、下发、提供等；

解除：指一类以部门为头实体、个人所有物或产业福利为尾实体的关系，用于表示个人所有物和产业福利不再存在，如解除、取消、减免等；

帮助：指一类以部门为头实体、群体为尾实体的关系，用于表示政府部门为民众完成的工作，如帮助、安置、安排等；

咨询：指一类以群体为头实体、部门为尾实体的关系，用于表示民众向政府部门表达办事需求的过程，如咨询、查询、访问等；

拥有：指一类以群体为头实体、个人所有物为尾实体的关系，用于表示个人所有物的所属关系，如拥有、享受、享有等；

申请：指一类以群体为头实体、产业福利为尾实体的关系，用于表示民众获取产业福利的过程，如申请、缴纳、参加等；

领取：指一类以群体为头实体、产业福利为尾实体的关系，用于表示民众已经获取产业福利的状态，如领取、获取、保有等；

参照：指一类以政策为头尾实体的关系，用于表示政策之间的引证关系，如参照、引用、包含等；

涉及：指一类以政策为头实体、产业福利为尾实体的关系，方便民众了解产业福利的变动，如涉及、提及、提到等；

符合：指一类以个人所有物为头实体、政策为尾实体的关系，用于表示民众的特定属性与政策中的要求一致，如符合、满足、达到等。

2.2.2 基于TransP的三元组表示

考虑到从产业政策文本中抽取到的实体和关系通常包含丰富的实际语义，且需要根据语义对三元组进行归并，本研究使用能够编码语义信息的表示模型 TransP 为原始三元组生成嵌入向量[39]。

具体符号记法如表 2 所示。h, t, v 分别表示三元组中的头实体、尾实体、动词短语。相应

的加粗字母 $\boldsymbol{h}$, $\boldsymbol{t}$, $\boldsymbol{v}$ 代表对应的向量。G 表示正例三元组集合，G' 表示负例三元组集合。

表 2 TransP 符号记法

符号	说明	符号	说明
h	头实体	$\boldsymbol{h}$	头实体向量
t	尾实体	$\boldsymbol{t}$	尾实体向量
v	关系	$\boldsymbol{v}$	关系向量
G	正例三元组集合	G'	负例三元组集合

在以往的研究中，模型使用随机向量作为实体和关系向量的初始值，这会导致语义信息的丢失。为缓解这一问题，本研究使用预训练的中文词向量作为实体和关系的初始向量。为了降低定语和副词对表示向量的影响，对于实体，本研究仅选择其中的名词性成分的词向量作为其初始向量；对于关系，本研究仅选择其中的动词性成分的词向量。例如，对于实体“养老保险 /n 公司 /nt”而言，其分词结果中均为名词性结构，因此本研究使用“养老保险”、“公司”的预训练词向量的平均值作为该实体的初始表示向量；对于关系“列支 /v 于 /p”，“列支”的词性为动词，“于”为介词，因此本研究使用“列支”的预训练词向量作为此关系的初始表示向量。

在进一步区分实体类别之前为它们生成恰当的表示向量的好处在于语义相近的实体、关系可以获得相似的向量表示，这样可以提升模型识别的性能。但是已有的研究中使用的表示模型，如 TransE，TransH 等，都没有在学习过程中对向量加以限制。这会使得表示向量在几轮训练以后就远离其原始描述，而使用这样的嵌入向量对于进一步的识别表现没有显著提升。因此，TransP 在表示模型的得分函数中加入了以自身描述向量为中心的罚项。即 $||h-h_c||_2^2$ 和 $||t-t_c||_2^2$，其中 $h_c=d_h$，$t_c=d_t$。基于以上考虑，TransP 的得分函数定义如下：

$$f_v(h,t)=||h+v-t||_2^2+\lambda_1||h-h_c||_2^2+\lambda_2||t-t_c||_2^2 \quad (1)$$

其中 $\lambda_1, \lambda_2 \in [0,1]$ 是用于控制 h_c 和 t_c 监督程度的超参数。

直观上，得分函数 $f_v(h, t)$ 可以理解为 $h+v$ 和 t 之间的一种距离。对于真实存在的三元组（正例），$f_v(h, t)$ 的值应该较小，反之较大。

为加大正负例三元组之间的得分差距，TransP 的损失函数定义如下：

$$L=\sum_{(h,v,t)\in G}\sum_{(h',v',t')\in G'}[\gamma+f_v(h,t)-f_{v'}(h',t')]_+ \quad (2)$$

其中 G 为正例三元组集合，G' 为负例集合。γ 为正负例边界，通常 γ=1。

通过最小化损失函数 L 即可获得各实体及关系的嵌入向量。所得关系向量将用于对原始三元组做进一步归并。

2.2.3 基于BIRCH的三元组归并

由于 DSNF 为开放域三元组抽取模型，抽取到的政策实体、关系所属类别数量较多，且难以事先确定，本研究基于所得嵌入向量 $\boldsymbol{v}$，利用层次聚类模型对原始三元组进行归并分组。在层次聚类模型中，平衡迭代削减层次聚类模型（Balanced Iterative Reducing and Clustering Using Hierarchies, BIRCH）有着较优的时间复杂度，$O(N)$，其中 N 为样本数量 [38]。考虑到从产业政策文本中抽取到的原始三元组数量庞大，本研究选择 BIRCH 作为层次聚类模型。

BIRCH 通过构建聚类特征树（Clustering Feature Tree, CF Tree）实现只需要单次扫描数据集即可完成聚类，每棵 CF Tree 则由若干聚

doi:10.3772/j.issn.2095-915x.2022.03.003

类特征（Clustering Feature, CF）组成。

在 CF Tree 中，一个 CF 是以三元组的形式定义的，记为（N，LS，SS）。其中 N 代表了这个 CF 中拥有的样本点的数量；LS 代表了这个 CF 中拥有的样本点各特征维度的和向量；SS 代表了这个 CF 中拥有的样本点各特征维度的平方和。

在此定义下，CF 满足线性关系，即

$$CF_1+CF_2=(N_1+N_2, LS_1+LS_2, SS_1+SS_2) \quad (3)$$

在此基础上，BIRCH 将该性质扩展到了 CF Tree 中，即对于每个父节点中的 CF 节点，它的三元组的值等于其所指向的所有子节点的三元组之和，如图 3 所示。由此，CF Tree 节点的更新效率将大幅提升。

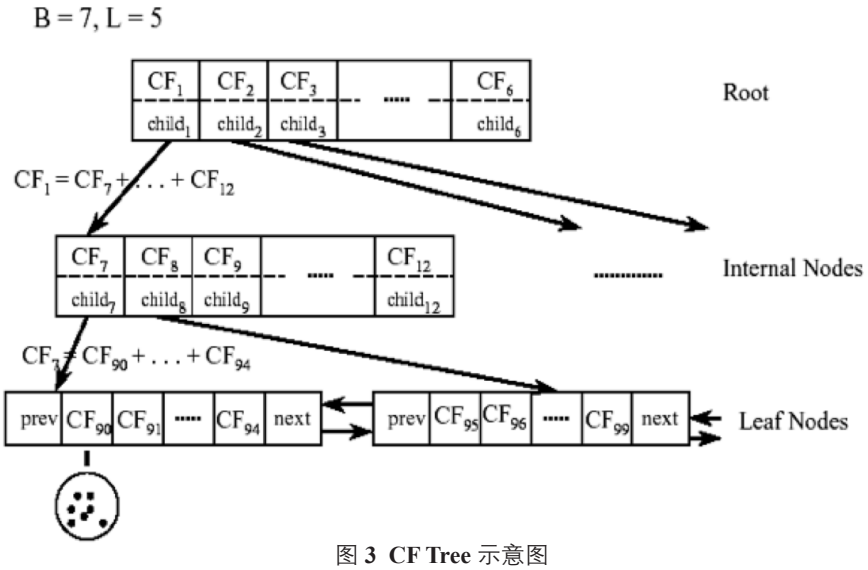


图 3 CF Tree 示意图

BIRCH 构造 CF Tree 的过程可大致分为以下四个步骤：

①从根节点向下寻找和新样本距离最近的叶子节点和叶子节点里最近的 CF 节点；

②如果新样本加入后，这个 CF 节点对应的超球体半径仍然满足小于阈值 T，则更新路径上所有的 CF 三元组，插入结束。否则转入③；

③如果当前叶子节点的 CF 节点个数小于阈值 L，则创建一个新的 CF 节点，放入新样本，将新的 CF 节点放入这个叶子节点，更新路径上所有的 CF 三元组，插入结束。否则转入④；

④将当前叶子节点划分为两个新叶子节点，选择旧叶子节点中所有 CF 元组里超球体距离最远的两个 CF 元组，分布作为两个新叶子节点的第一个 CF 节点。将其他元组和新样本元组按照距离远近原则放入对应的叶子节点。依次向上检查父节点是否也要分裂，如果需要按和叶子节点分裂方式相同。

在将所有样本建立成为 CF Tree 后，BIRCH 对应的输出就是若干个 CF 节点，并将每个节点里的样本点视作一个聚类的簇。

3 产业政策知识图谱的展示

3.1 数据收集

当前，本研究从北大法宝收集了养老产业政策文本共计 12854 条，其中包括中央政策

1422 条，各地方政策共计 11432 条。本数据集包含的政策文本数量较大，质量较高，具有一定的权威性；另一方面，数据集中的政策涉及的区域范围广（涵盖我国大部分省市、各大城市的地方性政策），部门层次结构深（包括党中央国务院至地方区级政府政策），时间跨度大（1980-2021 年），有利于在在后续的研究工作中对政策知识图谱进行对齐和演化分析。此外，本研究调用了 Jieba 自然语言处理工具对数据集进行了统计分析。收集到的政策文本平均包含 718.13 个词语，37.89 个句子，每个句子的平均长度为 17.32 个词。

表 3 养老产业政策数据集统计量

政策数量	中央政策	地方政策	平均词数	平均句子数	句子长度
12854	1422	11432	718.13	37.89	17.32

3.2 三元组抽取评测

为验证 DSNF 配合实体修饰词补齐方法的抽取效果，本研究从养老产业政策文本数据集中随机抽取了 400 余条政策文本，请专家从中手动标注政策三元组，并以其作为标准抽取结果与本研究所使用模型的抽取结果进行比对。所使用的评价指标包括查准率 (precision, P)、查全率 (recall, R)、F1 值，具体定义如下。

$$P=\frac{TP}{TP+FP}$$
$$R=\frac{TP}{TP+FN} \quad (4)$$
$$F1=\frac{2\times P\times R}{P+R}$$

其中 TP 为真正例数，FP 为假正例数，FN 为假负例数。直观上，查准率反映了抽取到的三元组中正确的三元组所占比例，查全率反映了政策文本中被正确抽取出的三元组所占的比例，F1 值兼顾了二者的考虑。

基于此，本研究对抽取结果进行评估，结果如表 4 所示。从中可以看出，本研究所使用的模型能够较为准确、全面地从政策文本中抽取到三元组，从而为产业政策知识图谱输送有价值的信息。

表 4 三元组抽取评测结果

查准率	查全率	F1
0.7449	0.7830	0.7586

3.3 基于Neo4j的产业政策知识图谱展示

对于收集到的养老产业政策文本，本研究基于提出的政策知识图谱构建框架，从中抽取关系三元组，并对其进行表示和归并，从而得到产业政策知识图谱。考虑到图数据库 Neo4j 具有较强的可扩展性和查询性能，能够在存储数十亿个实体和数万亿个关系的同时保证毫秒级查询响应时间，本研究将构建的知识图谱导入到 Neo4j 中，利用其中的可视化和检索功能响应用户的需求。

Neo4j 需要 Java 环境下运行，在使用前需要根据不同版本 Neo4j 的说明文档，安装对应版本的 Java 软件开发工具包 JDK（Java Development Kit）。同时，Neo4j 提供了 Python 接口，本研究在 Python 程序中调用 py2neo 封装包，从而实现 Python 针对 Neo4j 连接和操作。软件具体的环境如表 5 所示。

Neo4j 的数据库操作语言为 CQL（Cypher

doi:10.3772/j.issn.2095-915x.2022.03.003

Query Language）。CQL 基本命令分为增、删、改、查四类，对应的关键词是 CREATE、DELETE、SET、MATCH。CQL 是一种用户友好的图数据库查询语言，仅需少量代码，能够实现针对关系和实体及其属性的高效查询。

表 5 知识图谱创建阶段所使用软件及其版本信息

名称	版本号
Python	3.8
Neo4j	chinese-community-4.2.1-windows
JDK	11
py2neo	2020.1.1

Neo4j 中的数据类型包括实体和关系。在可视化界面中，实体由节点表示，通过颜色区分类型，同时也可以为实体赋予不同的属性值；关系由连接节点的有向线段表示，不同关系的名称不同。

3.3.1 产业政策知识图谱可视化

根据三元组数据的特点，在 Neo4j 中建立 2 种节点，分别表示头实体和尾实体，头实体有 11128 个，尾实体有 16821 个；将 2 种实体用 6147 种关系进行连接，关系总数是 32501 个。

产业政策知识图谱的最终效果如图 4 所示。由于实体和关系的数量大，使用“MATCH p=()-->() RETURN p LIMIT 500”命令，可视化 500 个关系和被它们联系的实体。图中绿色节点表示三元组中的头实体，红色节点表示尾实体，它们之间存在从头实体出发，指向尾实体的有向线段，表示关系。基于此，产业政策中的关键信息得以用一种结构化的形式来表示，民众或政府工作人员等用户不必阅读长篇的政策文本即可快速了解。

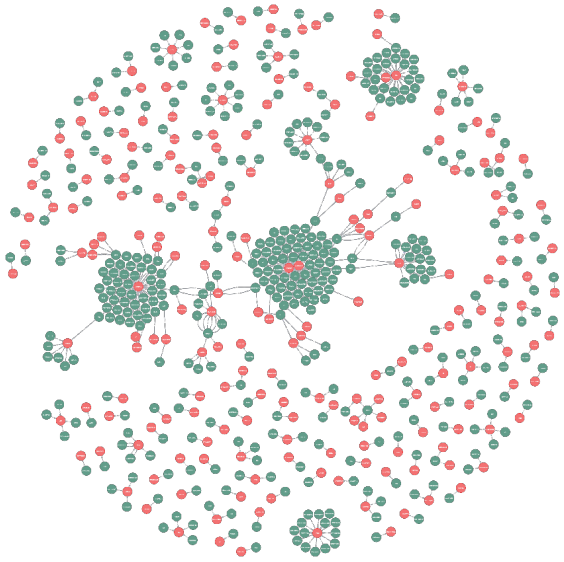


图 4 产业政策知识图谱可视化效果

3.3.2 产业政策知识图谱检索

如上文所说，产业政策知识图谱中包含大量的政策实体以及其间关系。且在实际应用中，知识图谱会根据各部门实时颁布的产业政策进行完善和更新，图谱体量会逐渐增大。因此，单独利用可视化功能从图谱中手动查找目标节点或关系意味着较高的人工成本和时间复杂度，而 Neo4j 中的检索功能恰可以响应用户的查询请求。当前知识图谱由三元组作为基本单元构成，因此可将用户的查询请求大致分为三类，分别是头实体查询、尾实体查询和关系查询，下面将结合现实生活中针对产业政策的信息需求举例说明。

（1）针对头实体的查询

在现实生活中，用户通常需要查找和了解某一类政策实体。例如，“符合什么条件的人可以领取养老金？”。在这个问句中，“符合什么条件的人”是头实体，“养老金”是尾实体，“领取”是关系。在Neo4j中，利用“MATCH (m:Head)-[`

领取`]->(n:Tail) where n.name = ‘养老金’ return m”语句进行查询，可视化结果如图 5 所示。从查询的结果可以看出，“退休职工”、“年满 60 周岁未享受城镇职工基本养老保险待遇的农村有户籍的老年人”、“参加工作连续工龄包括缴费年限满 10 年的人员”等可以领取养老金。

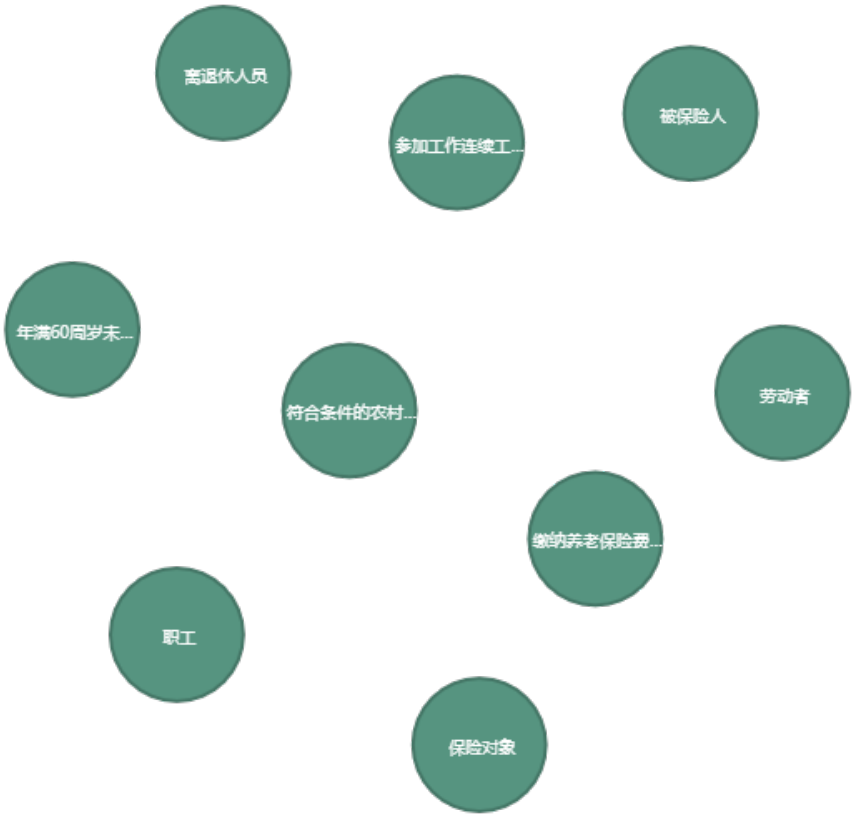


图 **5** 针对头实体查询结果可视化图

（2）针对尾实体的查询

类似地，用户感兴趣的政策实体同样可以作为三元组当中的尾实体，例如“投保人应填写什么文件？”。其中，“投保人”是头实体，“什么文件”是尾实体，“应填写”是关系。在 Neo4j 中，利用“MATCH (m:Head)-[r:` 应填写 `]->(n:Tail) where m.name = ‘投保人’ return n”语句进行查询，可视化结果如图 6 所示。从查询的结果可以看出，投保人需要填写“合同”、“申请书”或者“变更合同申请书”。

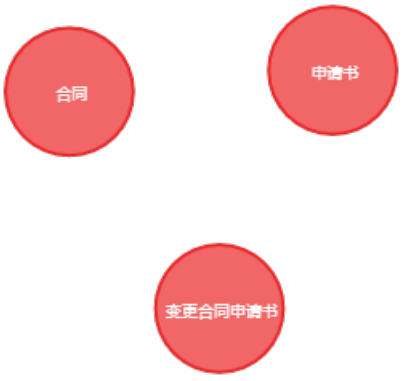


图 **6** 针对尾实体查询结果可视化图

（3）针对关系的查询

此外，用户有时也需要对实体之间的关系

doi:10.3772/j.issn.2095-915x.2022.03.003

进行查询，以了解关键的政策措施。比如“养老机构对老年人有哪些举措？”。在这个问句中，“养老机构”是头实体，“老年人”是尾实体，“有哪些举措”是描述两个实体的关系。在 Neo4j 中，利用“MATCH r=(m:Head)-->(n:Tail) where m.name = ‘养老机构’ and n.name = ‘老年人’ return r”语句进行查询，可视化结果如图所示。查询的结果如图 7 所示，从中可以看出，养老机构与老年人之间有“提供生活照料服务”、“建立健康档案”和“密切接触照护”等关系。

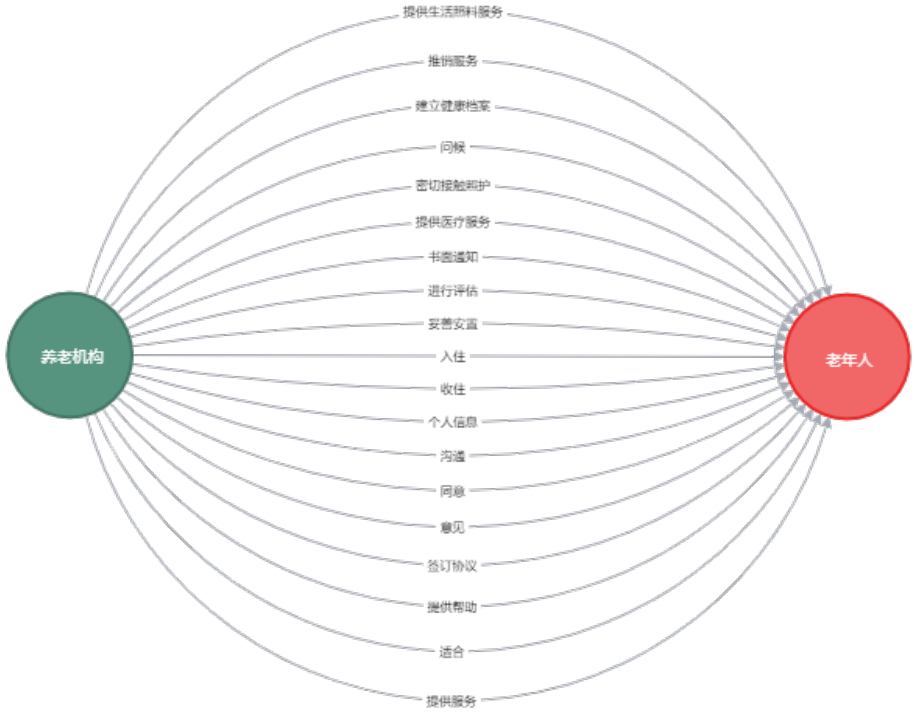


图 7 针对关系查询结果可视化图

除了上述三个例子外，用户可以根据自己的信息需求，编写对应的 CQL 语句，实现针对知识图谱的查询，并得到可视化的结果。

3.4 产业政策问答系统

为了完成与用户的交互，本研究为产业政策知识图谱构建了前端界面。同时，本研究从用户需求侧出发，开发了基于知识图谱的产业政策问答系统。

3.4.1 需求分析

开发产业政策问答系统的出发点是更好地解决用户的信息需求，包括政府公务人员、社会大众和其他潜在群体，要解决的关键性问题包括：1）实现关于产业政策信息的问答功能；2）系统与用户进行友好和人性化的交互。

考虑到微信的便利性和高用户粘性，本研究将最终的系统发布在公众号平台。设计的用户界面如图 8 所示，本智能问答系统最终将基于公众号平台的消息对话接口，可以理解用户以自然语言提出的问句，捕捉文字背后的搜索意图，然后自动生成查询语句来访问 Neo4j 图谱库，并最终将答案反馈给用户。

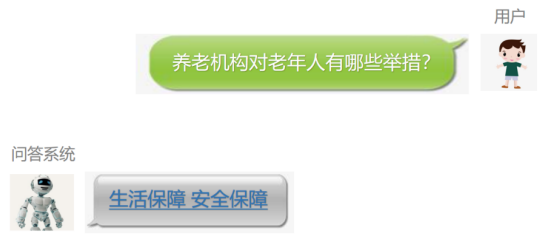


图 8 系统原型设计

本系统的核心是问答功能模块，能够理解简单的带有询问意味的自然语言中的信息需求，并在此基础上给出相应的答案。从实现的角度来看，图谱中保存的知识能够保障问答功能的实现，使得系统能够与用户进行一问一答。

3.4.2 系统整体框架设计

本研究借助微信公众号的消息对话接口实现智能问答系统，具体设计情况如图 9 所示。

为了连接公众号的消息对话接口，本研究编写 Web.py 框架，用于解析公众号的输入数据和产生输出内容。其中 GET 函数，用于实现云服务器对公众号对话接口的连接请求；POST 函数可以调用问答和检索模块。在解析到用户的输入问句后，根据问句的特点，调用对应的模块，生成正确的输出答案，将输出上传到公众号对话框页面。

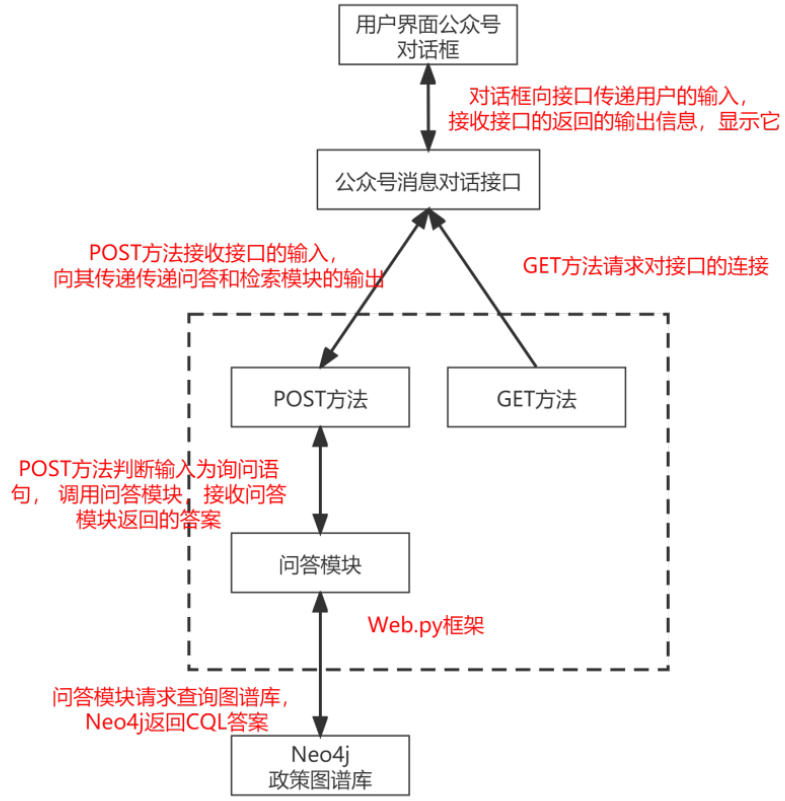


图 9 问答系统的架构图

3.4.3 问答功能模块开发

基于政策知识图谱库，实现问答功能。问答的步骤分为理解输入语句和组织输出语句两个步骤。流程见图 10。

doi:10.3772/j.issn.2095-915x.2022.03.003

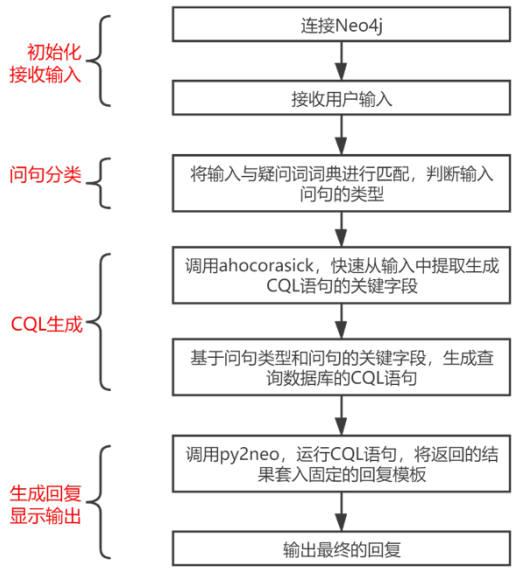


图 **10** 问答功能流程图

其中，理解输入，需要分辨提问的内容是关于实体，还是对实体的属性，或者是对实体之间的关系。而组织输出语句，是在明确了询问的主体后，编写和运行查询其的 CQL 语句，根据 Neo4j 返回的结果，生成答案。为了使得答案便于人的理解，可对 CQL 查询到的结果进行包装 [40]。

问答功能的软件开发环境如表 6 所示。理解用户的输入主要利用字符串匹配算法实现，需要利用 pyahocorasick 封装包，并通过 ++ build tools 在 C 语言编译环境下运行。实现输出主要依靠 Python 语言驱动 Neo4j 进行查询，调用 py2neo 即可。

表 **6** 问答功能开发阶段所使用软件及其版本信息

名称	版本号
Windows	10
Python	3.8.0
Neo4j	chinese-community-4.2.1-windows
JDK	11.0
py2neo	2020.1.1
pyahocorasick	1.4.1
visual c++ build tools	2015

（1）理解输入语句

为了实现理解输入语句的目的，首先针对知识图谱库的实体关系模型进行分析，明确问答范围。问题可以分为查询头实体、关系和尾实体三类。

明确问答范围后，设计调用知识图谱库的 CQL 语句模板。与问题涉及的知识图谱数据类型对应，CQL 语句模板分为三类，分别是：头实体匹配语句、关系匹配语句和尾实体匹配语句。CQL 模板如表 7 所示。

表 **7** **CQL** 语句模板示例汇总表

问句示例	检索项	CQL模板示例
什么能够领取到养老金?	h,name	MATCH (m:Head)-[r:`领取`]->(n:Tail) where n.name = ‘养老金’ return m.name, type(r), n.name
养老机构对老年人有哪些举措?	type(r)	MATCH (m:Head)-[r]->(n:Tail) where m.name = ‘{养老机构’ and n.name = ‘老年人’ return m.name, type(r), n.name
投保人应填写的东西?	t.name	MATCH (m:Head)-[r:`投保人`]->(n:Tail) where m.name = ‘填写’ return m.name, type(r), n.name

理解 CQL 查询语言后，剩下的理解输入过程主要分为三个步骤：问句分类和 CQL 数据库查询语句生成。

问句分类是一个简单的字符串匹配问题，依赖于提前建立好的不同类型问句的匹配字典。按照人们说话的习惯，本文建立了用于问句分

类的匹配字典，如表 8 所示。一旦识别到问句中有与某种类型的问句匹配字典相同的字段时，可以得到问句为该类型的结论。

完成问句分类后，开始生成 CQL，需要从问句中提纯关键字段的信息，然后按照对应问句类型的 CQL 模板，生成查询语句。生成 CQL 的过程，是指把关键字段的数据组合到 CQL 模板中，并从中提纯关键字段，下面进行详细说明。

表 **8** 用于问句分类的匹配字典汇总表

问句类型	检索项	匹配字典
什么能够领取到养老金?	h,name	什么能够、什么头实体、的头实体、什么人、什么、怎么
养老机构对老年人有哪些举措?	type(r)	什么尾实体、的尾实体、的文件、的材料、的动作、的东西
投保人应填写的东西?	t.name	的关系、针对、对

提纯关键词字段是基于 AC 自动机而实现（以下简称 AC）。关键词字段包括头实体、关系和尾实体，字符串数量庞大，利用 AC 能够缩短提取关键词字段的时间。

AC 的原理是利用字典树的形式存储字符串数据，借鉴 KMP 算法的思想，来判断待匹配字符串是否为字典树中存入字符串的子串。字典树利用树状结构，除了根节点外，每一个子节点按顺序逐个存储字符串最小单元，所以某个节点对应的字符串由从根节点到这个节点路径中所有的字符串最小单元按顺序组合而成。在向字典树中存入新的字符串数据时，要保证每一层的节点不出现重复。将 KMP 算法利用于树状结构则是指，当在某一条路径上匹配字符串失败时，首先应当从匹配失误的前一个字符串开始向前找到匹配路径中的某个子串，这个子串与字典树其它部分存在最大的相似之处，注意，这些相似之处节点比匹配路径上的节点后存入字典树，然后下一次匹配的起点，则是这个相似之处的节点的子节点。

（2）组织输出语句

CQL 语言生成后，调用 Graph.run 运行 CQL，得到针对问题的查询的结果。生成最终输出的过程与生成 CQL 的过程类似，需要把查询结果组合到回复模板之中。最后，打印输出加工后的回复模板。至此，问答功能模块得以实现。

3.4.4 问答系统界面搭建与展示

问答系统界面是沟通用户和问答功能模块的媒介。为将开发的问答系统发布在微信公众号平台上，以便用户查询使用，本研究设置了公众号的服务器地址 URL 和令牌 Token。URL 由 URL 云服务器外网 IP 地址和问答系统所在端口号组成。Token 是自己设置的一串字符，后期系统请求访问消息对话接口时，提交的 Token 要与此时在公众号平台配置的字符串一致。此外，本研究设计了 Web.py 框架，使系统能够接收用户输入的问句，判断问句的类型，并调用问答功能模块，接收到功能模块的返回信息，最终将返回语句发送到公众号对话框界面。整个框架的流程如图 11 所示，本研究主要通过编写 GET 方法和 POST 方法实现以上功能。

（1）GET 方法

为了访问公众号接口，编写了 GET 方法，

doi:10.3772/j.issn.2095-915x.2022.03.003

它的主要功能是向公众号平台提交 Token，此处的 Token 要与之前在配置公众号时输入的字符串一致。正确的 Token 可以证明系统是被该公众号平台认证了的，可以访问到公众号对话框中消息接收和回复接口的自主开发程序。

（2）POST 方法

通过 GET 方法成功连接对话接口后，系统编写了 POST 方法，来实现对话功能。首先，POST 可以接收用户在公众号平台对话的输入数据。由于接口提供的输入为 XML 形式，所以需要先解析，得到纯净的消息文本。之后，POST 方法根据问句的类型，调用问答功能模块，查询图谱库中对应的记录，生成回复文本。端口可以接收的输出形式也是 XML，所以需要进一步把回复文本包装成 XML 格式，然后 POST 方法向消息对话接口传送输出数据，回复的文本信息将出现在公众号对话框中。

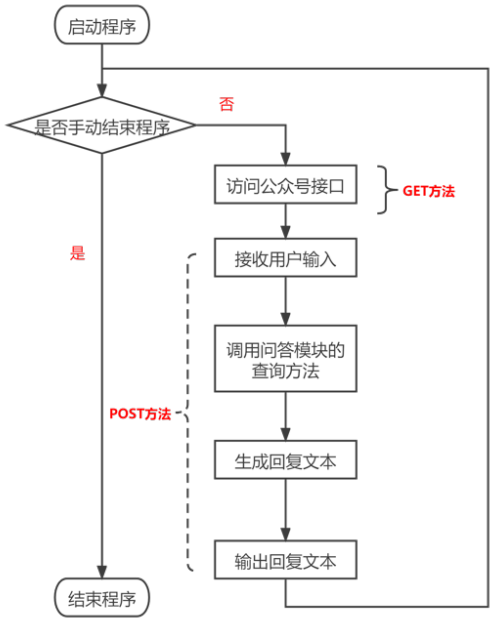


图 **11** 智能问答系统程序流程图

系统开发完成后，在云服务器上启动 Neo4j 图数据库和 Web.py 之后即可在公众号对话框使用产业政策信息的查询服务。

用户可通过扫码的方式进入公众号聊天界面，随后输入问句就可以开始以对话的形式对政策信息进行查找。公众号二维码见图 12。数据库系统能够回复文本形式的咨询。如果用户发送的消息不属于系统能识别的问句的格式，系统会提示用户输入正确形式的语句，如图 12 所示。

图 **12** 问答系统功能演示图

4 结论

当前我国各类产业经济在相应政策环境的滋养下蓬勃发展，然而产业政策通常具有数量大、篇幅长、易读性差等特点，不利于民众、政府工作人员快速了解政策文本的主要内容，

从而降低政府部门的办事效率。针对以上问题，本研究提出了一套产业政策知识图谱自动化构建框架，旨在梳理相关政策文本，方便民众和政府工作人员阅读。

对于输入的产业政策文本，本研究首先利用基于句法分析的开放域三元组抽取模型 DSNF 从中抽取政策实体和关系，并结合政务领域特点补齐了实体的修饰词。该方法的优势在于充分考虑了中文的语法特点，以及不需要大规模训练数据。随后，考虑到原始三元组较为分散，本研究利用三元组表示模型 TransP 和层次聚类模型 BIRCH 对三元组进行表示和归并。其中，TransP 能够将三元组所在语境包含的语义信息嵌入到表示向量中，而 BIRCH 能够在未知类别个数的前提下以较低的时间复杂度完成聚类。最后，本研究将三元组导入到图数据库 Neo4j 中，并基于其可视化和检索功能与用户交互。

当前研究工作存在一定的局限性。具体地，本研究重点关注从非结构化的产业政策文本中抽取政策实体以及其间关系，并基于此构建产业政策知识图谱。以后的工作将利用当前产业政策知识图谱中的关系进行推理分析，发掘隐式的关系并对现有的三元组进行纠错，从而将更加完整准确的知识图谱呈现给用户。此外，所搭建的政策知识图谱还可用于政策分析。第一，本研究计划利用知识图谱对齐技术对比不同地区、部门颁布的产业政策，帮助民众和政府工作人员了解其差异；第二，本研究将结合知识图谱的演化分析模型对不同时期的产业政策进行分析，方便民众和政府工作人员了解产业政策动向。

参 考 文 献

[1] 张璇，苏楠，杨红岗，等 . 2000-2011 年国际电子政务的知识图谱研究 —— 基于 Citespace 和 VOSviewer 的计量分析 [J]. 情报杂志，2012, 31(12):51-57.

[2] 朱晓峰，崔露方，陆敬筠 . 国内外政府信息公开研究的脉络、流派与趋势 —— 基于 WOS 与 CNKI 期刊论文的计量与可视化 [J]. 现代情报，2016, 36(10):141-148.

[3] 高天鹏，莫太林，莫太齐 . 基于知识图谱的国际开放政府数据研究概貌和热点透析 [J]. 现代情报，2019, 39(3):86-100.

[4] 陈强 . 我国政务微信研究的知识图谱与核心主题 [J]. 情报杂志，2018, 37(4):194-200.

[5] 夏立新，徐晨琛 . 基于主题图的电子政务门户知识导航系统构建研究 [J]. 图书馆论坛，2010, 30(6):184-187+146.

[6] 杨建梁，祁天娇 . 从电子文件到知识图谱 : 电子文件知识服务新途径 [J]. 档案学通讯，2020(2):10-19.

[7] 雷洁，赵瑞雪，李思经，等 . 科研档案管理知识图谱构建研究 [J]. 科技管理研究，2020, 40(11):162-169.

[8] Yates A, Cafarella M, Banko M, et al. TextRunner: Open information extraction on the web[C]. Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, NAACL-Demonstrations '07, Rochester, New York. 2007: 25-26.

[9] 畅玉洁 . 大数据背景下 web 数据的可视化研究分析 [J]. 信息系统工程，2018(1):148.

[10] 洪敏，吴红亚，杨保华 . 基于 HTML 的 ECharts 的动态数据显示前端设计 [J]. 计算机时代，2018(8):27-28+32.

[11] 许德山，张智雄，邢美凤 . 面向本体知识库的可视化检索研究 [J]. 情报理论与实践，2010, 33(8):114-117.

[12] 由丽萍，郎宇翔 . 基于商品评论语义分析的情感知识图谱构建与查询应用 [J]. 情报理论与实践，2018, 41(8):132-136+131.

[13] 胡春阳，廖信林 . 中外区域政策比较分析 : 理论综

doi:10.3772/j.issn.2095-915x.2022.03.003

述及实践启示 —— 基于 CiteSpace 的知识图谱量化研究 [J]. 中国流通经济 , 2017, 31(6):97-105.

[14] 罗哲 , 唐迩丹 . 我国人才政策的演变趋势与发展方向 —— 基于 CiteSpace 知识图谱分析 [J]. 软科学 , 2021, 35(2):102-108.

[15] 马续补 , 刘玮 , 秦春秀 . 基于知识图谱的我国政策评估研究主体、知识基础、研究热点与演进分析 [J]. 现代情报 , 2019, 39(3):166-177.

[16] 吴宾 , 唐薇 . 基于知识图谱的国内养老政策研究热点主题与演化路径 (2005-2016)[J]. 人口与发展 , 2018, 24(2):101-112.

[17] 赵绘存 , 高峰 , 闫杰 . 2007—2017 年国际科技政策研究热点与前沿 —— 基于科学知识图谱视角 [J]. 科技管理研究 , 2018, 38(3):42-49.

[18] 张雨 , 吴俊 . 科技政策知识图谱构建研究 [J]. 数字图书馆论坛 , 2021(8):31-38.

[19] 霍朝光 , 钱毅 , 祁天娇 . 基于开放公文的新冠肺炎政策知识图谱构建与分析 [J]. 档案学通讯 , 2021(2):53-62.

[20] Wu F and Weld D S. Open information extraction using Wikipedia[C]. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL '10, Uppsala, Sweden. 2010: 118-127.

[21] Mausam, Schmitz M, Bart R, et al. Open language learning for information extraction[C]. Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL '12, Jeju Island, Korea. 2012: 523-534,.

[22] Fader A, Soderland S, and Etzioni O. Identifying relations for open information extraction[C]. Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '11, Edinburgh, United Kingdom. 2011: 1535-1545.

[23] De Marneffe M-C, Dozat T, Silveira N, et al. Universal Stanford dependencies: A cross-linguistic typology[C]. Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC'14, Reykjavik, Iceland. 2014: 4585-4592.

[24] White A S, Reisinger D, Sakaguchi K, et al. Universal decompositional semantics on Universal Dependencies[C]. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP'16, Austin, Texas. 2016: 1713-1723.

[25] Mesquita F, Schmidek J, and Barbosa D. Effectiveness and efficiency of open relation extraction[C]. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP'13, Seattle, USA. 2013: 447-457.

[26] Akbik A and Löser A. KrakeN: N-ary facts in open information extraction[C]. Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-Scale Knowledge Extraction, AKBC-WEKEX '12, Montreal, Canada. 2012: 52-56.

[27] Stanovsky G, Michael J, Zettlemoyer L, et al. Supervised open information extraction[C]. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (New Orleans, Louisiana). 2018: 885-895.

[28] Jia S and Xiang Y. Hybrid neural tagging model for open relation extraction[J]. Arxiv, 2019, https://arxiv.org/abs/1908.01761v3.

[29] Shen Y, Tan S, Sordoni A, et al. Ordered neurons: Integrating tree structures into recurrent neural networks[J/OL]. Arxiv, 2018, https://arxiv.org/pdf/1810.09536

[30] Cui L, Wei F, and Zhou M. Neural open information extraction[C]. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL'18, Melbourne, Australia. 2018: 407-413.

[31] Jia S, Shijia E, Li M, et al. Chinese Open Relation Extraction and Knowledge Base Establishment[J]. Acm Transactions on Asian & Low Resource Language Information Processing, 2018, 17(3):1-22.

[32] Yang B, tau Yih W, He X, et al. Embedding entities and relations for learning and inference in knowledge bases[C]. Proceedings of 3rd International Conference on Learning Representations, ICLR'15. 2015: 1-12.

[33] Nickel M, Tresp V, Kriegel HP. A three-way model for collective learning on multi-relational data[C]. Proceedings of the 28th International Conference on

Machine Learning. ICML'11, Washington, D.C., USA. 2011: 809-816.

[34] Bordes A, Usunier N, Garcia-Duran A, et al. Translating embeddings for modeling multi-relational data[C]. Proceedings of Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems. 2013: 1-9.

[35] Feng J. Knowledge graph embedding by translating on hyperplanes[C]. Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, AAAI'14, Quebec City, Canada. 2014: 1112-1119.

[36] Lin Y, Liu Z, Sun M, et al. Learning entity and relation embeddings for knowledge graph completion[C]. Proceedings of AAAI. 2015: 2181-2187.

[37] Ji G, He S, Xu L, et al. Knowledge graph embedding via dynamic mapping matrix [C]. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, IJCNLP'15, Beijing, China. 2015: 687-696

[38] Zhang T, Ramakrishnan R, and Livny M. BIRCH: An efficient data clustering method for very large databases[C]. Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, SIGMOD '96, Montreal, Quebec, Canada. 1996: 103-114.

[39] Geng Q, Chuai Z, Jin J. Automatic construction of academic profile: a case of information science domain[J/OL]. Journal of Information Science, 2021(4):016555152199804.

[40] Liu H Y. QA System On Medical KG[DB/OL]. (2020-08-13) [2021-12-03]. https://github.com/liuhuanyong/QASystemOnMedicalKG.