计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications [doi: 10.15888/j.cnki.csa.008592]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

发动机故障领域知识图谱构建与应用①

许驹雄[1], 李敏波[1,2], 刘孟珂[1], 曹志月[3], 唐　波[3], 葛　浩[3]

[1](复旦大学 软件学院, 上海 200438)
[2](复旦大学 上海市数据科学重点实验室, 上海 200438)
[3](潍柴动力股份有限公司, 潍坊 261061)
通信作者: 李敏波, E-mail: limb@fudan.edu.cn

摘　要: 发动机生产故障和售后维修报告中有大量动力总成和零部件故障信息. 本文将知识图谱引入柴油发动机故障领域, 设计发动机故障领域知识图谱构建的系统流程, 针对多源故障数据进行本体建模. 使用 BERT 和 BiLSTM-CRF 结合的实体识别框架, 挖掘故障数据中的专家知识. 提出实体相关性评价指标 FF-IEF, 并基于知识图谱和贝叶斯网络进行故障诊断. 设计并开发 EFKG 原型系统, 共包含 12 534 个实体和 408 972 条三元组, 该系统提供知识抽取、可视化检索、辅助决策等功能, 有效提高信息检索和维修效率, 对知识图谱在发动机故障领域的应用具有一定指导意义.

关键词: 柴油发动机; 知识图谱; 智能制造; FF-IEF; 故障诊断

引用格式: 许驹雄,李敏波,刘孟珂,曹志月,唐波,葛浩.发动机故障领域知识图谱构建与应用.计算机系统应用. http://www.c-s-a.org.cn/1003-3254/8592.html

Construction and Application of Knowledge Graph in Diesel Engine Fault Field

XU Ju-Xiong[1], LI Min-Bo[1,2], LIU Meng-Ke[1], CAO Zhi-Yue[3], TANG Bo[3], GE Hao[3]

[1](Software School, Fudan University, Shanghai 200438, China)
[2](Shanghai Key Laboratory of Data Science, Fudan University, Shanghai 200438, China)
[3](Weichai Power Co. Ltd., Weifang 261061, China)

Abstract: There is a large amount of failure information from the engine after-sales maintenance and failure reports. This study introduces knowledge graphs and designs a systematic building procedure for the field of engine fault. It carries out ontology modeling for the multi-source fault data. The entity recognition framework that combines BERT with BiLSTM-CRF is used to mine expert knowledge in fault data. The index FF-IEF (fault frequency-inverse event frequency) is proposed, and fault diagnosis is performed based on the knowledge graph and Bayesian network. We design and develop the prototype system EFKG that contains 12 534 entities and 408 972 triplets. The system provides knowledge extraction, visual retrieval, and auxiliary decision-making. It can effectively improve the efficiency of information retrieval and maintenance and is of guiding significance for the application of knowledge graphs in the field of engine fault.

Key words: diesel engine; knowledge graph; intelligent manufacturing; FF-IEF (fault frequency-inverse event frequency); fault diagnosis

随着智能制造时代的到来, 越来越多的制造企业和服务商都开始搭建基于产品全生命周期的物理信息系统用于采集产品的设计、采购、加工、装配、测试和售后返修等过程数据及结果数据, 例如发动机装配档案, 出厂测试阶段的试车数据、售后返修的故障维修报告等. 这些数据蕴含了丰富的价值, 但厂商们缺乏有效的技术手段, 无法从数据和知识层面指导发动机的故障诊断和维修工作[1].

① 基金项目: 国家重点研发计划 (2018YFB1703104); 国家自然科学基金 (61671157)
收稿时间: 2021-10-20; 修改时间: 2021-11-18; 采用时间: 2021-11-30; csa 在线出版时间: 2022-03-18

目前的工业领域故障诊断方法大多基于生产过程中的状态数据, 通过构建传感器获取的特征数据和机器状态之间的关系, 将故障诊断问题转化为模式识别问题, 在制造业[2]、电力[3] 等领域都有诸多应用. 但由于维修信息中具有大量的文本信息, 如何提取其中蕴涵的领域知识是需要深入研究的课题[4].

随着人工智能的飞速发展, 知识图谱逐渐成为工业界和学术界研究的重点, 广泛应用于医疗[5]、教育[6] 等领域. 在制造业领域, 西门子提出了领域知识图谱计划[7], 博世构建了底盘系统控制相关数据的大型知识图谱[8]. 知识图谱通过三元组描述数据之间的关系, 这种结构化的表示降低了从中提取信息的难度. 与此同时, 利用知识抽取相关技术将非结构化数据构建成知识图谱, 可以将文本信息用接近人类认知的格式保存, 从而挖掘数据蕴含的价值.

在知识图谱的自动化构建方面, 关键技术包括命名实体识别、关系抽取和实体对齐等. 目前具有代表性的工作有 Huang 等人提出的双向长短时记忆网络 (bidirectional long short term memory, BiLSTM) 配合条件随机场 (conditional random field, CRF) 的模型[9]. Qiu 等人使用空洞卷积加强模型的上下文信息编码能力和运行速度[10]. Yan 等人将相对距离驱动的注意力机制引入 Transformer 模型, 以提高其在命名实体识别中的表现[11]. Li 等人提出一种多粒度点阵框架, 实现了提取中文文本关系的任务[12]. Sun 等人提出了一种基于嵌入实体对齐的引导方法, 迭代地将可能的实体对齐标记为训练数据, 以学习面向对齐的图嵌入[13]. Cao 等人将图卷积神经网络和注意力机制引入实体对齐任务, 以获得表示知识图中实体分布的连接实体的重要性权重[14].

在知识图谱应用方面, 目前基于知识图谱的个性化推荐技术主要分为基于路径和基于图嵌入两种. Zhao 等人引入元图概念获取知识图中更丰富的语义信息[15]. Zhu 等人使用知识图中实体间的关系链接来传播用户偏好并了解其潜在偏好[16]. Wang 等人提出一种顺序学习框架, 通过特征学习得到实体向量和关系向量, 利用 CNN 融合得到用户向量和物品向量[17]. Zhang 等人将知识学习和协同过滤的目标函数结合进行联合学习[18]. Wang 等人使用联合学习框架来计算多跳响应[19], 并在后续工作中提出多任务学习框架交替学习图嵌入和推荐算法, 同时利用了两个任务的互补信息[20].

然而, 通过文献调研和与制造业厂商交流发现, 在制造业领域应用知识图谱还存在诸多不确定性, 缺乏系统的研究. 例如, 厂商们对制造业知识图谱的应用前景有所怀疑, 并且不确定如何将其应用到产品设计、装配、售后等流程. 此外, 目前还缺乏一个有效的、系统的从发动机故障数据端到端构建与应用知识图谱的流程. 因此, 本文的主要工作如下:

1) 将知识图谱引入柴油发动机故障领域, 提出发动机故障知识图谱 (engine fault knowledge graph, EFKG). 分析发动机故障诊断领域的业务规则和数据特点, 设计领域知识图谱的构建流程和本体, 基于真实数据集构建 EFKG.

2) 针对维修数据中知识抽取准确率较低的问题, 构建领域词典, 标注语料集, 从多维度对比现有的基于深度学习的实体抽取方法, 得出最好的发动机维修数据命名实体识别方案.

3) 设计实体相关性评价指标 FF-IEF (fault frequency-inverse event frequency) 和基于知识图谱的辅助决策模型, 并开发原型系统, 提供知识抽取、检索、辅助决策等功能.

本文对柴油发动机故障领域知识图谱的构建和应用进行研究. 实验结果表明, 本文的方法能有效地从发动机故障数据集中抽取知识, 有助于提高信息检索和售后维修效率.

1 发动机故障知识图谱构建

1.1 EFKG 构建流程

在发动机故障诊断领域, 存在许多案例形式的故障维修数据, 与故障诊断相关的知识需从案例中挖掘, 例如故障现象、故障原因、故障状态、故障部位等. 由于该领域作为传统制造业, 专业知识存在一定的封闭性, 数据质量和应用也存在一定问题, 目前与知识图谱相关的研究较少. 本文针对发动机故障领域的数据特点和业务逻辑, 自顶向下构建知识图谱, 整体流程如图 1 所示.

具体构建流程如下:

1) 根据领域专家提供的发动机故障诊断业务规则和数据特点设计知识图谱模式层.

2) 制定映射规则对结构化数据进行转换; 从非结构化文本中通过实体识别技术抽取实体, 并与其他实体进行关联.

3) 通过实体对齐对实体进行规范化处理, 根据模式层关联关系生成三元组.

4) 计算实体相关性指标, 与三元组存储于图数据库中.

5) 利用贝叶斯推理模型进行故障诊断.

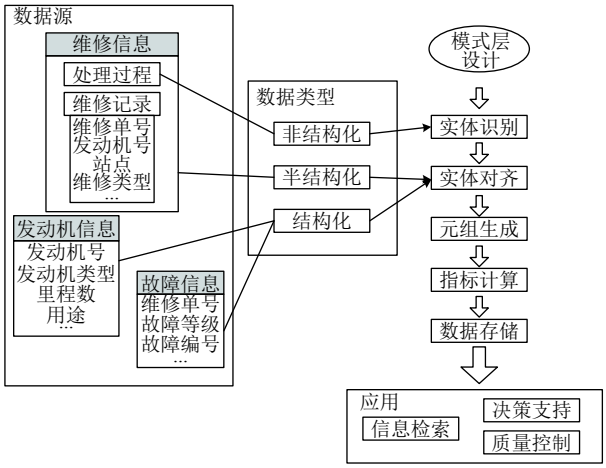

图 1 EFKG 构建流程

1.2 EFKG 模式层设计

本文使用数据来源于潍柴动力股份有限公司近年来的生产故障 (加工、试车、装配等) 和售后维修报告, 包括维修信息、发动机信息、故障信息等, 其格式涵盖结构化和非结构化数据. 每条维修记录对应一个柴油发动机故障案例, 并通过外键与发动机信息和故障信息等外表关联. 其中维修处理过程为非结构化文本, 故障信息和发动机参数为结构化数据.

EFKG 的重要用途之一是提高维修效率, 即辅助工程师定位故障位点和故障类型, 因此故障部位和故障状态是核心实体, 整体模式层设计如图 2 所示.

1.3 故障实体标注

发动机故障维修报告为工作人员手工填写的自然文本, 通常包括“客户反映-问题定位-解决方法”流程, 如表 1 所示, 下划线部分为需要抽取的信息, 包括维修信息、发动机信息、故障信息等.

在发动机故障领域, 目前并无公开的训练语料库, 需自行标注和构建数据集. 为解决训练集规模小、部分领域词汇一词多义的问题, 本文基于目前主流使用的 BiLSTM-CRF[9] 方法, 将 BERT[21] 预训练模型作为词向量输入, 可以较好地缓解上述问题, 学习到更准确的语义向量. 模型的整体结构如图 3 所示.

本文对 5 488 条发动机维修数据进行人工标注, 构建了发动机维修数据集, 如表 2 所示.

本文采取 BIO 和 BIOES 两种标注方法. BIO 的标注方案将词语分成两类, 一类是目标实体, 由 B 和 I 组成, 分别代表目标实体的第一个词语和其他词语, O 表示该词语不属于目标实体. BIOES 的 B、I、E 分别表示实体的开始、中间和结束部分, S 表示实体为单个字词, O 表示该部分不是实体.

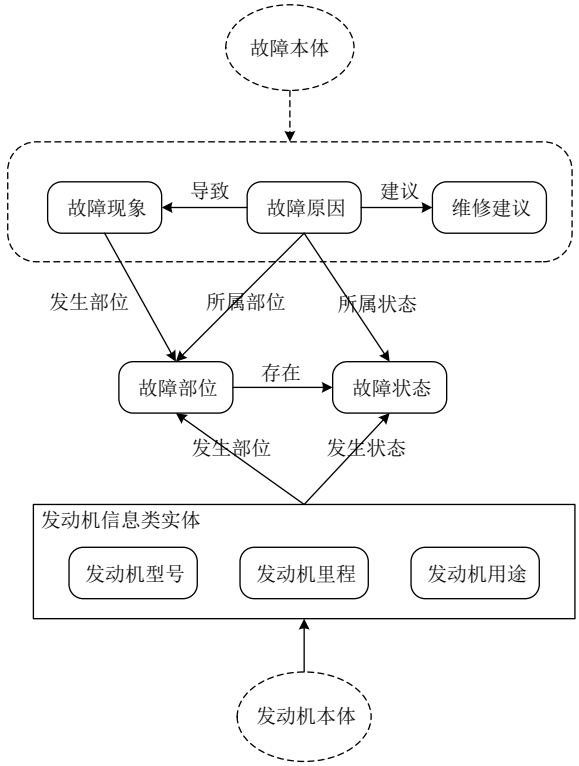

图 2 EFKG 模式层设计

表 1 非结构化数据实例

故障描述	数据来源
试车过程中发现油气分离器冒白烟, 拆检后发现增压器拉壳、烧轴, 作报废处理.	试车故障
用户反映发动机漏机油, 给予拆卸变速箱检查发现发动机曲轴后油封漏油, 给予更换处理.	售后故障

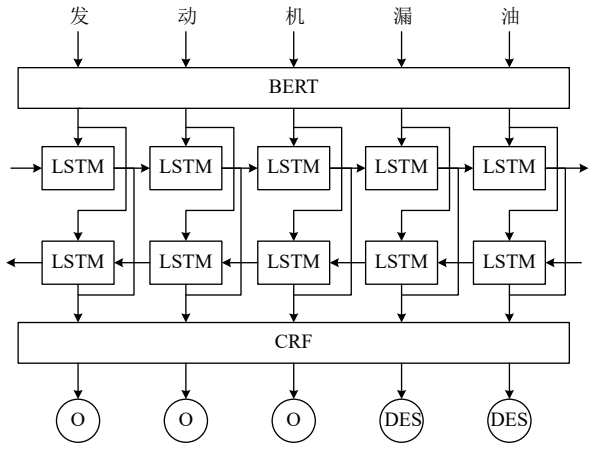

图 3 BERT-BiLSTM-CRF 模型

表 2 实验数据集大小及划分

总句数	总字数	训练集	验证集	测试集
5 488	214 064	3 842	1 098	548

在标注数据集中, 实体类型共分为 4 种: 故障现象 (description, DES)、故障部位 (location, LOC)、故障状态 (status, STA) 和维修建议 (suggestion, SUG). 故障现象是指客户向维修站点反映的发动机故障表现, 如“发动机启动困难”; 故障部位是指经检查后确定的问题起因件, 如“向心球轴承”“增压器”等; 故障状态是指起因件出现的具体问题, 例如“(增压器) 拉壳”“(油封) 漏油”等; 维修建议是指维修人员解决故障的操作, 如“更换”气缸盖垫片等. 各类实体的标注情况如表 3 所示.

表 3 实体标注情况

标签类型	标签含义	实体数量
x_DES	故障现象	3 546
x_LOC	故障部位	3 102
x_STA	故障状态	3246
x_SUG	维修建议	2 806

标注示例如表 4 所示.

表 4 标注示例

原始文本	实体类别	对应实体
用户反映发动机震动大, 经我站拆检后发现该车向心球轴承卡滞, 我站给予更换后故障排除.	故障现象	震动大
	故障部位	向心球轴承
	故障状态	卡滞
	维修建议	更换

实验结果见本文第 2 节.

1.4 实体对齐

维修报告为工作人员手工填写, 无法保证数据的规范和实用性, 常出现共指现象, 如“发动机无力”和“功率不足”指代同一问题. 同时由于数据经过 OCR 处理, 存在中英文字符识别出错的情况, 如电子控制单元 ECU 识别成 EC0, 类似的异常数据需要进行消除和修复.

本文采用计算相似度的方法进行实体对齐, 定义好相似度函数和阈值后, 将实体间相似度得分大于设定阈值的实体对只保留其中一个实体, 并更新图谱中的三元组, 用保留后的实体替换被对齐的实体. 本文采用编辑距离和 Jaccard 相关系数法进行实体相似度计算.

1) 编辑距离: 对一个单词或词语可以采取插入、删除或替换字符 3 种方式. 将一个单词 (词语) 通过这 3 种操作方式转换为另一个单词需要的最小操作次数, 即为编辑距离.

2) Jaccard 相关系数法: Jaccard 系数描述了两个有限样本集的相似性, 定义为两个集合的交集与并集之比. 该比值越大, 说明两个集合越相似; 该比值越小, 说明两个集合差异越大, 相似性越低. Jaccard 相关系数的计算方法如式 (1) 所示.

$$J(S_1,S_2)=\frac{|S_1\cap S_2|}{|S_1\cup S_2|}=\frac{|S_1\cap S_2|}{|S_1|+|S_2|-|S_1\cap S_2|} \tag{1}$$

实体对齐流程如下所示. 由于相似度计算仅考虑文本的字面相似性, 而忽略了语义, 因此该方法不能保证实体对齐的完全正确, 可能存在错误对齐或遗漏对齐的情况. 由于本文涉及的实体主要与发动机故障信息相关, 实体种类和数量相对较少, 因此对实体对齐结果进行人工校对与完善.

算法 1. 实体对齐算法

输入: 所有故障实体, 相似度阈值 s
输出: 对齐后的实体和关系

```
1. Function EntityAligned(engineFailNodes)
2.     for uniNode in engineFailNodes:
3.         for alignedNode in engineFailNodes:
4.             if uniNode == alignedNode : continue;
5.             uniAttrList ← uniNode.attrs
6.             aliAttrList ← alignedNode.attrs
7.             sim ← simComp(uniAttrList, aliAttrList)
8.             if sim <= s : continue;
9.             for hasConnect(uniNode, alignedNode):
10.                node.relation = uniNode
```

1.5 指标计算

在 EFKG 中, 一条三元组并非绝对正确或错误的. 例如“发动机震动大”这一故障现象, 可能由于“轴承卡滞”导致, 也可能由 “减震器损坏”引起. 为了描述一条三元组在 EFKG 中的重要程度, 本文参考 TF-IDF (term frequency-inverse document frequency) 的思想, 设计了发动机故障实体相关性指标 FF-IEF (fault frequency-inverse event frequency).

对于 EFKG 中的一条三元组 (h_{ij}, r_i, t_{ij}), 故障频率 (FF) 衡量尾实体在给定头实体条件下出现的概率, 如式 (2) 所示.

$$FF(h_{ij},t_{ij})=\frac{N(h_{ij},t_{ij})}{\sum_{k=1}^{|H_i|}N(h_{ij},t_{ik})} \tag{2}$$

其中, $N(h_{ij}, t_{ij})$ 表示该条三元组出现的次数, 可从维修

数据集中统计并作为三元组的属性存储. H_i 表示头实体 h_{ij} 所属的实体类别 (故障现象、故障部位等).

逆向事件频率 (*IEF*) 衡量尾实体对头实体的区分程度, 定义为头实体所属类别的元组总数与该头实体所在元组数的比值, 如式 (3) 所示.

$$IEF(H_i,t_{ij})=\log(\frac{N(H_i)}{|\{(h_{ik},r_i,t_{ij}):h_{ik}\in H_i,\forall k\}|}) \quad (3)$$

其中, $|\{(h_{ik},r_i,t_{ij}):h_{ik}\in H_i,\forall k\}|$表示尾实体为 t_{ij} 的三元组集合.

FF-IEF 指标同时衡量三元组的出现频率和区分程度, 定义如式 (4) 所示.

$$FF\text{-}IEF(h_{ij},t_{ij})=FF(h_{ij},t_{ij})\times IEF(H_i,t_{ij}) \quad (4)$$

该指标可有效衡量尾实体对于头实体的重要程度, 可用于信息检索和推荐等应用. 在 EFKG 中, 一条三元组可被描述为$<(h,r,t),M>$, 其中 h 表示头实体, r 表示关系, t 表示尾实体, M 包含 3 个属性值: 出现频率 N, 故障频率 *FF* 和逆向事件频率 *IEF*.

2 实验结果与图谱应用

2.1 实体识别结果

2.1.1 评价指标

本文使用准确率 (*Precision*)、召回率 (*Recall*) 和 *F*1 值作为模型的评估指标. 计算公式如下:

$$Precision=\frac{TP}{TP+FP}\times 100\% \quad (5)$$

$$Recall=\frac{TP}{TP+FN}\times 100\% \quad (6)$$

$$F1=\frac{2\times Precision\times Recall}{Precision+Recall}\times 100\% \quad (7)$$

其中, *TP* 为正确预测的实体数, *FP* 为预测错误的实体数, *FN* 为没有识别出的实体数.

2.1.2 实验设置

实体识别实验基于 PyTorch 进行搭建, 具体的环境配置参数等如表 5 所示.

表 5 实验环境

实验环境	配置参数
操作系统	Ubuntu 18.04.3 LTS
深度学习库	PyTorch 1.8.1
编程语言	Python 3.8.8
CPU	Intel(R) Xeon(R) Silver 4210 @ 2.20 GHz
GPU	NVIDIA GeForce RTX3090

本实验采用 BERT-Base 模型, 该模型使用 Bi-Transformer 关注语义上下文, 在多项 NLP 任务中表现良好. 其他模型参数如表 6 所示.

表 6 参数设置

超参数	值
LSTM size	128
Dropout	0.5
Learning rate	1E–4
Batch size	32
Epoch	200

2.1.3 实验结果

实验采取层次抽样的方法构建训练集、验证集和测试集, 数据划分情况见上文表 2. 不同模型、不同标注粒度和标注方案的实验结果如表 7 所示.

表 7 不同维度的 3 种模型实体识别情况 (%)

模型	标注方案	准确率	召回率	*F*1值
CRF	基于字的BIO	82.33	70.50	75.95
	基于字的BIOES	86.54	78.25	82.18
	基于词的BIO	83.63	63.54	72.21
	基于词的BIOES	86.10	68.93	76.56
BiLSTM-CRF	基于字的BIO	80.92	73.87	77.23
	基于字的BIOES	82.44	78.28	80.31
	基于词的BIO	86.81	84.48	85.63
	基于词的BIOES	88.26	85.73	86.97
BERT-BiLSTM-CRF	基于字的BIO	87.31	80.05	83.52
	基于字的BIOES	90.22	81.47	85.62
	基于词的BIO	91.55	84.71	87.99
	基于词的BIOES	93.01	87.66	90.25

可以看到, 采用 BERT-BiLSTM-CRF 模型和基于词的 BIOES 标注方案得到的命名实体识别效果最好, *F*1 值为 90.25%.

基于词和基于字是两种不同的标注粒度. 由于中文的词之间没有严格的界限, 且自动化的分词工具有一定误差, 因此在通用领域中, 基于字的标注粒度更为主流. 但对于发动机故障领域而言, 领域词典能保证较高的分词准确率, 并且词向量相比字向量能包含更准确的语义信息, 因此整体而言, 基于词的标注方案优于基于字的方案.

从标注方案角度而言, 3 种模型的结果都是 BIOES 优于 BIO 方案, 即更细致化的标注能给命名实体识别带来更好的效果. 例如, “发动机”一词在 BIO 方案下会被标注成“B_LOC”, 其后可能存在“共轨管 (I_LOC)”或其他类型的标注, 对整体的识别造成困难; 而在

BIOES 方案中, “发动机”直接标注成“S_LOC”, 实体边界更清晰, 有利于识别效果提升.

本实验中, 不同实体类别的识别效果如表 8 所示.

表 8 不同标注类别的实体识别情况 (%)

模型	实体类别	准确率	召回率	*F*1值
BiLSTM-CRF	故障现象	85.42	86.05	85.73
	故障部位	74.31	69.66	71.91
	故障状态	77.19	74.38	75.75
	维修建议	84.56	98.46	90.98
BERT-BiLSTM-CRF	故障现象	91.06	86.37	88.65
	故障部位	82.12	77.10	79.53
	故障状态	81.27	75.16	78.00
	维修建议	85.65	98.48	91.62

可以看到, 在发动机维修数据中, 维修建议与故障现象的整体识别效果较好, 主要由于其结构性较强, 一般由两三个词概括而成, 如“漏油”“动力不足”等. 而故障部位和故障状态实体的准确度较低, 则由于其在句子中出现的位置较随机, 且上下文信息不确定性较强, 在小数据集上表现一般.

2.2 基于贝叶斯推理的辅助决策模型

辅助决策模型即在给定发动机信息和表现的情况下, 推荐其可能出现的故障原因. 以故障部位为例, 根据朴素贝叶斯定理, 给定发动机当前状态 S, 任意一个故障部位 FL_i 出现问题的概率如式 (8) 所示.

$$P(FL_i|S)=\frac{1}{J}\times P(FL_i)\times\prod_{k=1}^{|S|}P(S_k|FL_i) \tag{8}$$

其中, $S=\{\text{Mileage},\text{Model},\text{PrdUse},\text{FalutSym},\cdots\}$为给定发动机的参数信息, $J=P(S_1,S_2,\cdots,S_{|S|})$为参数集合 S 的联合分布.

对于一台给定的发动机, J 值是固定的, 可将其忽略. $P(FL_i)$为该部位发生故障的先验概率, $P(S_k|FL_i)$即三元组$<(h,r,t),M>$的 FF 值 (见式 (2)), 其中 h 为 FL_i, t 为 S_k. 因此, 该值均可以从三元组的属性中直接获取.

对于一个故障部位, 可能存在多个故障状态 FS, 任意一个故障状态 FS_j 的概率如式 (9) 所示.

$$P(FS_j|S,FL_i)=\frac{1}{J}\times P(FS_j)\times\prod_{k=1}^{|S|}P(S_k|FS_j)\times P(FL_i|FS_j) \tag{9}$$

其中, S 为发动机的参数集合, $J=P(FL_i,S_1,S_2,\cdots,S_{|S|})$表示 S 和 FL_i 的联合分布, 且对于不同的故障状态该值固定. 类似的, $P(FS_j)$, $P(S_k|FS_j)$和$P(FL_i|FS_j)$可从对应的三元组属性中直接获取.

故障原因 FR 由故障部位 FL 和故障状态 FS 联合表示, 如式 (10) 所示.

$$P(FR_{ij})=P(FL_i|S)\times P(FS_j|S,FL_i) \tag{10}$$

为了评估该辅助决策模型的有效性, 本文将其与 XGBoost[22] 和 LightGBM[23] 进行对比. 实验为一个多分类任务, 即给定发动机信息, 预测其故障原因. 在实验前, 本文对数据集做了一些预处理, 如缺失值填充、连续值离散化、离散特征编码等.

发动机故障原因有数百种, 遵循帕累托原理, 即大多数事故是由少数原因引起的, 并且由于长尾分布, 某些故障原因的样本数较小, 直接在全体数据集上运行分类模型效果较差. 因此, 本文构建了多个数据集用于测试模型在不同故障原因类别数下的性能, 如表 9.

表 9 不同故障原因类别数据集

故障类别数	数据量	训练集	验证集	测试集
20	33 048	19 828	6 610	6 610
30	59 143	35 485	11 829	11 829
50	85 895	51 536	17 179	17 179
100	127 067	76 241	25 413	25 413

根据故障原因类别数, 将这些数据集称为 *FR*(20), *FR*(30), *FR*(50) 和 *FR*(100). 本文使用 *Recall*@5 作为评价指标, 实验结果如表 10 所示.

表 10 不同模型的 *Recall*@5 值

模型	*FR*(20)	*FR*(30)	*FR*(50)	*FR*(100)
XGBoost	0.792 1	0.744 5	0.708 2	0.642 7
LightGBM	0.806 1	0.762 4	0.715 8	0.633 2
ours	0.852 0	0.822 1	0.776 8	0.703 2

可以看到, 本文设计的辅助决策模型性能比直接在原始数据集上运行多分类模型的效果更好.

3 EFKG 系统设计与实现

3.1 系统设计

本文基于构建后的知识图谱, 设计并实现了 EFKG 原型系统, 主要提供以下功能.

1) 知识抽取

厂商们在历年的发动机故障维修工作中已积累大量历史售后数据, 并以文本的形式存储, 然而目前难以利用海量的非结构化数据. 知识抽取从非结构化数据中自动化识别故障实体, 将数据转化为知识, 并以三元组的方式存储, 有利于后续的检索和诊断功能.

2) 知识检索

基于历史故障信息获取故障和故障之间的相似性一直是领域研究重点. 利用知识图谱对实体进行相关性评价指标排序, 可以帮助工作人员充分挖掘历史故障信息中包含的价值.

3) 辅助诊断

发动机结构的复杂性导致其故障难以避免, 而在不拆解发动机的情况下很难预测其故障原因. 本文利用知识图谱提供先验知识, 利用基于贝叶斯推理的辅助决策系统确定故障原因, 有助于在拆解前提高发动机故障诊断的效率和准确性.

系统总体架构如图 4 所示.

系统包括数据持久层、控制层和视图层. 数据持久层使用 Neo4j 图数据库和 MongoDB 非关系型数据库分别保存三元组和算法模型. 控制层采用 Django 框架, 通过 RESTful 风格的 API 接收前端查询请求, 生成 Neo4j 数据库的 DQL 语句后, 通过 Py2neo 接口调用 Neo4j 的引擎, 并将结果返回给前端展示. 对于 MongoDB 的算法模型 (实体识别、辅助故障诊断) 调用也通过控制层进行. 视图层负责前端页面展示, 使用 JavaScript 和 Echart 工具完成图表绘制, 并提供较为简洁的交互功能.

3.2 EFKG 系统查询与可视化

系统从近年来潍柴公司数十万条柴油发动机售后维修报告中抽取 12 534 个实体和 408 972 条三元组, 存储在 Neo4j 图数据库中, 部分示例如图 5 所示.

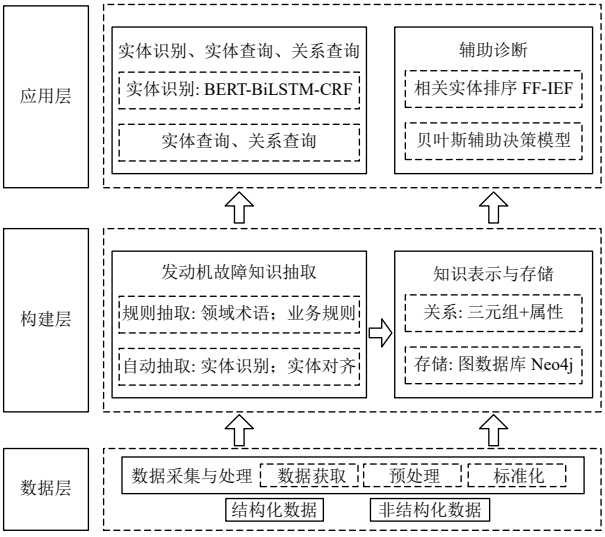

图 4 系统总体架构

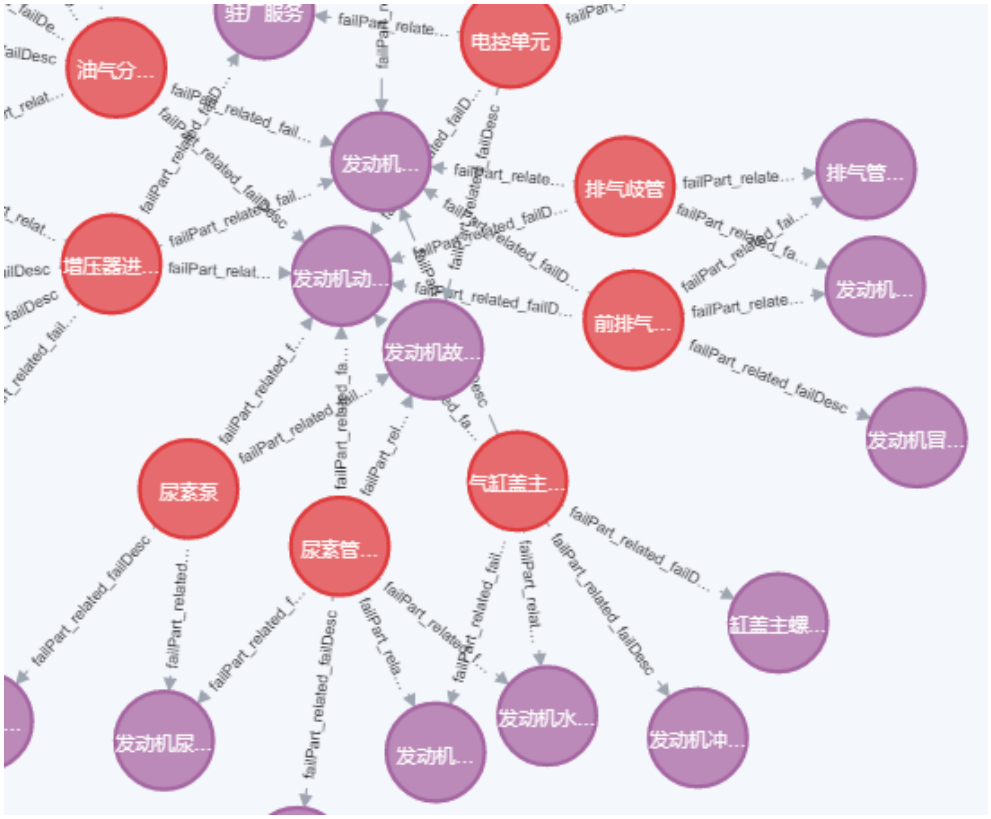

图 5 图数据库示例

实体识别模块负责从输入语句中识别发动机故障实体, 如图 6 所示. 当前端页面输入发动机维修相关语句时, 后台调用已训练好的模型进行实体识别, 并将标注结果返回到前端 (蓝色字体标识), 鼠标点击对应文本可查看其所属的实体类别. 该模块实现了维修报告的自动化录入.

实体查询模块可查询实体与实体间的关系, 也可直接输入 Cypher 查询语言进行更灵活的自定义查询, 如图 7 所示, 查询“前排气歧管”, 返回与之相关的实体和关系并进行可视化展示. 前端页面通过 Echarts 渲染, 点击实体或关系可以查看对应三元组的属性.

图 6 实体识别模块

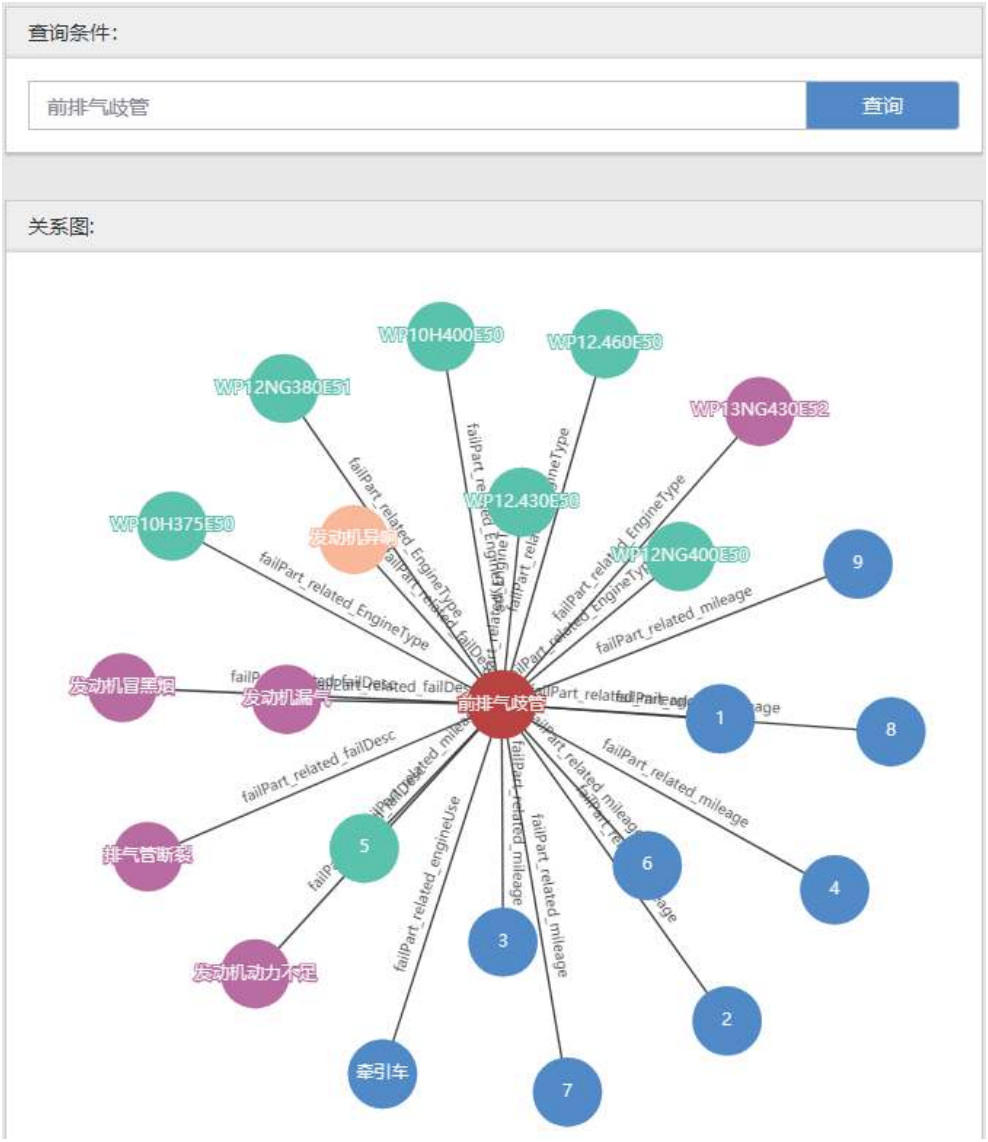

图 7 实体查询模块

相关实体会在下方以表格形式展示, 并通过 FF-IEF 指标排序. 图 8 展示了与“活塞”相关的部分故障现象, “发动机机油耗高”现象与该部位的 FF-IEF 值为 2.07, 说明二者相关性较高.

辅助诊断模块自定义输入发动机特征 (里程、型号、用途、故障现象等), 特征数可通过“新增条件”按钮控制, 通过调用基于贝叶斯的辅助诊断模型预测其可能出现的故障原因, 如图 9 所示.

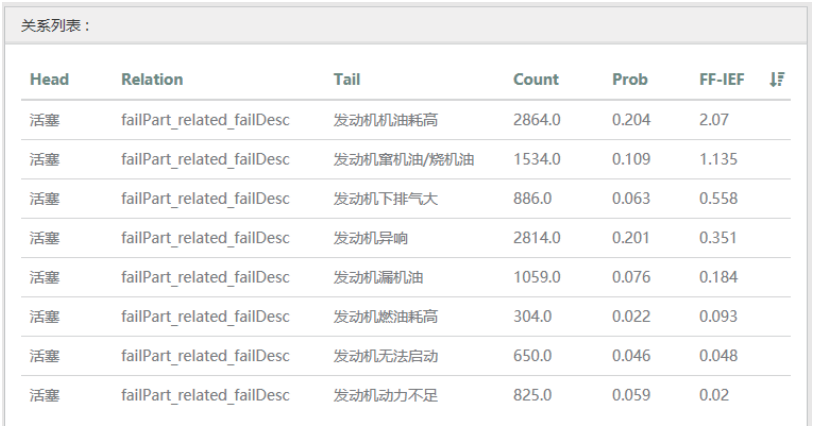

图 8 实体相关性列表

图 9 辅助诊断模块

4 结论与展望

为解决发动机维修过程中极度依赖维修人员个人经验、缺乏定量事实依据等问题, 本文利用发动机故障报告构建发动机维修领域知识图谱, 深度挖掘设备之间共性问题, 从数据和知识层面指导发动机故障诊断和维修工作, 主要成果如下.

1) 建立了从真实发动机维修数据集中构建知识图谱的系统流程和本体设计, 构建了发动机故障知识图谱 EFKG, 共包含 12 534 个实体和 408 972 条三元组.

2) 对发动机维修领域文本做了较全面的命名实体识别对比实验. 整体而言, BERT-BiLSTM-CRF 模型基于词的标注粒度和 BIOES 标注方案效果更好.

3) 设计了实体相关性评价指标 FF-IEF 和基于贝叶斯推理的辅助决策模型, 相比基于机器学习的多分类模型取得更好的推理效果.

4) 设计并实现 EFKG 原型系统, 基于 Neo4j 图数据库存储和 Django Web 框架, 实现了查询和可视化等功能, 为 EFKG 的落地应用提供技术参考.

后续研究一方面可聚焦在整个发动机维修领域的大规模数据集的构建, 另一方面可在故障原因推理模型中, 考虑扩充数据来源和影响因素, 提高推理效果.

致谢: 潍柴动力股份有限公司张明国工程师的支持.

参考文献

1 张栋豪, 刘振宇, 郑维强, 等. 知识图谱在智能制造领域的研究现状及其应用前景综述. 机械工程学报, 2021, 57(5): 90–113.

2 曹正志, 叶春明. 改进 CNN-LSTM 模型在滚动轴承故障诊断中的应用. 计算机系统应用, 2021, 30(3): 126–133. [doi: 10.15888/j.cnki.csa.007830]

3 刘兆炜, 王汉军, 李丹, 等. 改进 SOM 神经网络在电力调度故障诊断中的应用. 计算机系统应用, 2018, 27(3): 179–185. [doi: 10.15888/j.cnki.csa.006279]

4 Ji SX, Pan SR, Cambria E, *et al*. A survey on knowledge graphs: Representation, acquisition, and applications. IEEE Transactions on Neural Networks and Learning Systems, 2021. 1–21. [doi: 10.1109/TNNLS.2021.3070843]

5 Li LF, Wang P, Yan J, *et al*. Real-world data medical knowledge graph: Construction and applications. Artificial Intelligence in Medicine, 2020, 103: 101817. [doi: 10.1016/j.artmed.2020.101817]

6 罗明. 教育测评知识图谱的构建及其表示学习. 计算机系统应用, 2019, 28(7): 26–34. [doi: 10.15888/j.cnki.csa.006977]

7 Hubauer T, Lamparter S, Haase P, *et al*. Use cases of the industrial knowledge graph at siemens. Proceedings of the ISWC 2018 P&D/Industry/BlueSky Tracks. Monterey: CEUR-WS.org, 2018.

8 Schmid S, Henson C, Tran T. Using knowledge graphs to search an enterprise data lake. European Semantic Web Conference. Portorož: Springer, 2019. 262–266.

9 Huang ZH, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging. arXiv: 1508.01991, 2015.

10 Qiu JH, Wang Q, Zhou YM, *et al*. Fast and accurate recognition of Chinese clinical named entities with residual dilated convolutions. 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). Madrid: IEEE, 2018. 935–942.

11 Yan H, Deng BC, Li XN, *et al*. Tener: Adapting transformer encoder for named entity recognition. arXiv: 1911.04474, 2019.

12 Li Z, Ding N, Liu ZY, *et al*. Chinese relation extraction with multi-grained information and external linguistic knowledge. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence: Association for Computational Linguistics, 2019. 4377–4386.

13 Sun ZQ, Hu W, Zhang QH, *et al*. Bootstrapping entity alignment with knowledge graph embedding. Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI-18). Stockholm: IJCAI.org, 2018. 4396–4402.

14 Cao YX, Liu ZY, Li CJ, *et al*. Multi-channel graph neural network for entity alignment. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence: Association for Computational Linguistics, 2019. 1452–1461.

15 Zhao H, Yao QM, Li JD, *et al*. Meta-graph based recommendation fusion over heterogeneous information networks. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2017. 635–644.

16 Zhu GM, Bin CZ, Gu TL, *et al*. A neural user preference modeling framework for recommendation based on knowledge graph. 16th Pacific Rim International Conference on Artificial Intelligence. Cuvu: Springer, 2019. 176–189.

17 Wang HW, Zhang FZ, Xie X, *et al*. DKN: Deep knowledge-aware network for news recommendation. Proceedings of the 2018 World Wide Web Conference. Republic and Canton of

Geneva: International World Wide Web Conferences Steering Committee, 2018. 1835–1844.

18 Zhang FZ, Yuan NJ, Lian DF, *et al*. Collaborative knowledge base embedding for recommender systems. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2016. 353–362.

19 Wang HW, Zhang FZ, Wang JL, *et al*. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. Proceedings of the 27th ACM International Conference on Information and Knowledge Management. New York: ACM, 2018. 417–426.

20 Wang HW, Zhang FZ, Zhao M, *et al*. Multi-task feature learning for knowledge graph enhanced recommendation. The World Wide Web Conference. New York: ACM, 2019. 2000–2010.

21 Devlin J, Chang MW, Lee K, *et al*. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv: 1810.04805, 2018.

22 Chen TQ, Guestrin C. Xgboost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2016. 785–794.

23 Ke GL, Meng Q, Finley T, *et al*. LightGBM: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems. Red Hook: Curran Associates Inc., 2017. 3149–3157.

(校对责编：牛欣悦)