计算机科学
COMPUTER SCIENCE

使用语义解析构建面向分布式SCADA系统的自然语言接口

王涛, 郭武士, 邓健, 陈亮

引用本文

王涛, 郭武士, 邓健, 陈亮.使用语义解析构建面向分布式SCADA系统的自然语言接口[J]. 计算机科学, 2023, 50(6A): 220300141-9.

WANG Tao, GUO Wushi, DENG Jian, CHEN Liang. Building Natural Language Interfaces for Distributed SCADA Systems Using Semantic Parsing [J]. Computer Science, 2023, 50(6A): 220300141-9.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.220300141

使用语义解析构建面向分布式 SCADA 系统的自然语言接口

王　涛[1,4]　郭武士[3,4]　邓　健[5]　陈　亮[2,4]
1 西南交通大学电气工程学院　成都 610031
2 西南交通大学机械工程学院　成都 610031
3 电子科技大学信息与软件工程学院　成都 610054
4 四川省装备制造业机器人应用技术工程实验室　四川 德阳 618000
5 华润电力投资有限公司　山东 青岛 266071

摘　要　受限于传统的程式固定的视窗界面人机交互方式,大型分布式工业过程 SCADA 系统主要运营于中控机房,配置专业人员维持运行,系统建设和运营维护成本很高,因此探索人机自然交互接口,引导系统自适应服务意义重大。以一种面向多种专业领域的分布式 SCADA 系统为背景,从实际运营的角度分析人机自然交互的核心需求。按照自然语言指令的复杂程度,推荐不同的语义解析算法。首先对指令采取词性标注,确定指令是否包含子指令。对于基本自然语言指令,采用 TF-IDF 关键词提取算法并结合余弦相似度进行结构化抽取,将其解析为 SCADA 操控中间语言后经形式化转换为实际操控指令。对于复杂自然语言指令,采用基于依存句法分析的结构化指令解析算法,实现实时操控接口。实验结果表明,所提出的自然语言接口能较好地解决 SCADA 系统的人机自然语言交互问题,指令解析方面的平均精确率、召回率以及 F 值分别达到了 89.27%,89.28%以及 89.27%,平均响应时间为 1.593 s,特别是为工农业信息化管控提供了更为便捷的交互手段。

关键词:自然语言接口;神经网络语言模型;依存句法分析;SCADA 系统;语义解析
中图法分类号　TP311

Building Natural Language Interfaces for Distributed SCADA Systems Using Semantic Parsing

WANG Tao[1,4],GUO Wushi[3,4],DENG Jian[5] and CHEN Liang[2,4]
1 School of Electrical Engineering,Southwest Jiaotong University,Chengdu 610031,China
2 School of Mechanical Engineering,Southwest Jiaotong University,Chengdu 610031,China
3 School of Information and Software Engineering,University of Electronic Science and Technology,Chengdu 610054,China
4 Robotics Engineering Laboratory for Sichuan Equipment Manufacturing Industry,Deyang,Sichuan 618000,China
5 China Resources Power Investment Company Limited,Qingdao,Shandong 266071,China

Abstract　Due to the traditional program fixed window interface human-computer interaction,large distributed industrial process SCADA systems are mainly operated in the central control room and maintained by professional staff,so the system construction and operation and maintenance costs are very high,and it is significant to explore the natural human-computer interaction interface and guide the system adaptive services. Taking a distributed SCADA system for various professional fields as the background,this paper analyzes the core requirements of natural human-computer interaction from the perspective of actual operation. Different semantic parsing algorithms are recommended according to the complexity of natural language instructions. For basic natural language instructions,TF-IDF keyword extraction algorithm is used and combined with cosine similarity for structured extraction,which is parsed into SCADA manipulation intermediate language and converted into actual manipulation instructions by formalization. For complex natural language instructions,a structured instruction parsing algorithm based on dependency syntax analysis is used to realize the real-time control interface. Experimental results show that the proposed natural language interface can better solve the human-computer natural language interaction problem of SCADA system. The average accuracy,recall and F-value of instruction parsing is 89.27%,89.28% and 89.27%,respectively. The average response time is 1.593s,which provides a more convenient means of interaction,especially for industrial and agricultural information control.

Keywords　Natural language interface,Neural network language model,Dependent syntactic analysis,SCADA system,Semantic parsing

---

基金项目:德阳市重点科技计划项目(2018SZY066);四川工程职业技术学院规划科研项目(KJGH2020G08);德阳市科技计划项目(成果转化)(2022KCZ158);四川省科技计划项目(2022YFG0224)
This work was supported by the Key Science and Technology Project of Deyang(2018SZY066),Sichuan Engineering Technical College Planning Science and Research Project(KJGH2020G08),Deyang Science and Technology Plan Project(Transformation of Results)(2022KCZ158) and Sichuan Province Science and Technology Plan Project(2022YFG0224).
通信作者:王涛(2510392466@qq.com)

1 引言

大型工业过程、电力系统的综合监控通常采用分布式系统控制与数据采集(Supervisory Control And Data Acquisition,SCADA)的技术方案。但是,一般 SCADA 系统建设和运维成本较高,需配置专门的运维和操作人员,且现有的几乎都是基于图形用户界面(Graph User Interface,GUI)进行人机交互的。操作人员通过观察图表数据、点击按钮控件设置和部署 SCADA 系统,这需要掌握专门知识的运维工程师通过视窗界面在固定的程式下完成有限的操作。人机交互的模式对用户不够友好,且操控效率较低。

解决这一问题,根本上还需要从人机交互的自然属性入手。SCADA 系统的自然语言交互接口是一种有效的手段。自然语言接口(Natural Language Interface,NLI)使用人类语言与机器系统交互,降低了交互复杂性[1],这在复杂系统和复杂设备的操控方面显得特别重要,如大型分布式 SCADA 系统、无人机、自动驾驶等[2]。在移动和 Web 应用、数据管理等方面,自然语言接口也正在被广泛应用。例如,面向移动安卓终端提供自然语言交互接口,如微软、苹果、百度的语音助手等;Su 等构建的面向 Web 的自然语言接口,实现了对 Web 邮件等应用的自然语言操作,提高了 Web 应用程序的使用效率[3];Min 对中文 Text-to-SQL 算法的初步研究,为中文自然语言查询数据库提供了一种新的思路[4]。

自然语言接口的任务是将用户表达的语音或者文本转换成可执行的计算机系统调用接口。其核心问题是解决下列两个步骤:

(1)解析操控指令的复杂程度以及复杂操控序列之间的输入输出调用关系,核心是自然操控语言的形式化语义功能描述;

(2)自然语言指令序列与计算机系统调用接口之间的映射。

第(1)步是自然语言处理中的语义理解问题,也是解决自然语言接口的本质和根本问题,通常需要经过文本分词、词性标注、命名实体识别和句法分析等过程,通过模式分类和语义理解识别用户操控指令意图。第(2)步在第(1)步的基础上表达操控序列和系统调用接口后,解决相似度计算和最优化映射问题,其效果很大程度上取决于第(1)步。

操控的意图识别的研究现在主要分为 3 类:基于规则的方法、基于统计特征分类的办法以及基于深度学习的方法。基于规则的方法使用关键字等信息形成规则来检测用户的意图。Chu 等[5]提出了一个基于对话的对象查询系统,利用余弦相似度和 TD-IDF 的混合方法确定用户意图。基于统计特征分类的办法需要对语料文本进行关键特征提取,然后通过训练分类器实现意图分类。常用的方法包括朴素贝叶斯、支持向量机以及逻辑回归等。Muhammad 等[6]提出了一种多项式朴素贝叶斯分类方法来识别意图,在这项研究中,创建的聊天机器人能够理解用户输入的自然语言并根据用户的期望进行回答。基于深度学习的通常做法是先进行文本向量化,常用的方法有 Word2vec[7],Glove[8]以及 BERT[9]等;再使用模型进行特征提取,常用的模型有 LSTM[10],GRU[11]以及 CNN[12]等;最后利用 Softmax 层完成意图分类工作。Zhou 等[13]提出一种 BiLSTM-Attention 的改进结构用于文本分类,但在例如“维度灾难”“语义鸿沟”以及“复杂度高”等方面还需要进一步优化提高[14-15]。通常,深度学习的做法需要大量高质量的训练数据,理论上深度学习模型适用于指令意图分类任务,但是对算力、数据的要求都比较高。

自然语言指令的形式化描述及指令翻译,当前主要依赖于统计与规则方法并举的自然语言处理或自然语言理解技术[16],尤其是随着深度学习理论的发展,分布式特征表示给自然语言处理发展带来了新的突破。文献[17]提出的神经语言模型,输入大规模的语料库作为训练数据,得到语料文本中词语的向量形式——词向量,即词语的分布式表征,再通过获得的词向量进行如自动问答、自然语言接口等的 NLP 下游任务。Sowmya 等[18]提出了一个面向 Web 端的根据用户复杂需求发现可组合服务集的框架,该方法基于自然语言处理和语义理解来解析服务数据集的功能语义,能够有效地为简单和复杂的查询找到相关服务。但其在考虑复杂查询和简单查询的区别时,仅通过几个关键字进行区分,容易造成歧义。Tian 等[19]提出了一种基于依存句法分析的病理报告结构化处理方法,将语句转换为依存句法树的形式,然后依次遍历结点获取关键字。该方法存在结点提取过多而导致信息冗余的问题,但是从依存句法角度解析语言为自然语言接口的设计提供一种新思路。针对数据库的自然语言接口近些年也成为了研究热点。Xu 等[20]利用列注意力机制处理“序列到集合”的生成问题,避免了 where 子句中的顺序相关问题。Tao 等[21]提出的 TYPESQL 架构,和 SQLNet 一样,依旧将 SQL 语句生成问题视为槽填充问题,但是利用了类型信息来更好地理解自然语言问题中的稀有实体和数据。Hwang 等[22]讨论了基于 BERT 的架构的 3 种变体,并且给出了如何在语义分析任务中使用单词语境化的方法。

综上所述,设计面向人机交互的自然语言接口,对于降低大型分布式 SCADA 系统的运维和服务升级成本具有积极作用。但存在的亟待研究的问题:缺少小数据驱动、强泛化能力、低计算量的智能计算模型来指导自然语言接口设计,以解决复杂不确定语言环境下的自然语言交互接口设计问题。需要从面向 SCADA 系统这一特定领域的自然语言查询与控制接口设计出发,结合语义分析和深度学习理论,研究一种科学、合理且泛化能力强的面向多种领域的自然语言接口。本文贡献如下:

(1)设计了一个基于语义理解的自然语言接口,将自然语言转化为 SCADA 可以识别的机器语言(Text-to-SCADA),能有效识别并解析基本自然语言指令和复杂自然语言指令;

(2)实验表明,所提出的 Text2SCADA 框架能够有效解决 SCADA 系统的人机自然交互问题,并取得了较好的准确率,时间性能也符合要求。

本文第 2 节给出了 SCADA 系统的中间语言格式;第 3 节介绍了 Text2SCADA 的模块功能以及相应的算法细节;第 4 节通过实验验证了所使用方法和所提出技术的理论性能;最后全文总结并展望未来。

2 SCADA 系统中间语言格式

在 SCADA 系统中,HTTP 地址分为 3 部分:服务器 IP 地址、视图绝对路径和视图通道 ID。其中,前半部分的 host-id,PATH 以及 Plugs 等是我们不需要关心的无关变量。cnlNum 对应的是采集数据的通道编号(即变量名),viewID 对应

的是视图编号（即地点名），year，month，day 为日期。通过这 3 个属性即可检索到某一日期下的视图编号和通道编号下的变量值。图 1 给出了一个指令参数自动填充草图。

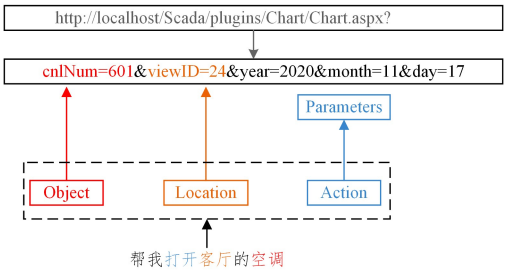

图 1　客厅温度数据的 HTTP 地址

Fig. 1　HTTP address for living room temperature data

如在智能家居系统中，需要抽取的信息有 3 个：对象、地点以及动作。当输入自然语言指令“打开客厅的空调”时，需要将其转化为数据结构如“{Object＝灯，Location＝客厅，Action＝打开}”的中间语言。然后，SCADA 读取上述结构中的 Object 和 Location 等属性的值，即可检索到目标，从而完成相应的操作。

3　Text-to-SCADA 架构

要实现自然语言对 SCADA 系统的操控，关键在于抽取自然语言指令中的对象指标、位置指标以及动作指标。本文设计了如图 2 所示的 Text-to-SCADA 架构，即将自然语言转换为 SCADA 系统可以识别的机器语言。首先，指令经过基本、复杂指令识别，若是基本自然语言指令，则推荐 TICS（TF-IDF＋Cos_Sim）算法进行语义解析，否则推荐 SDPA（Semantic Dependency Parsing Algorithm）算法进行解析。

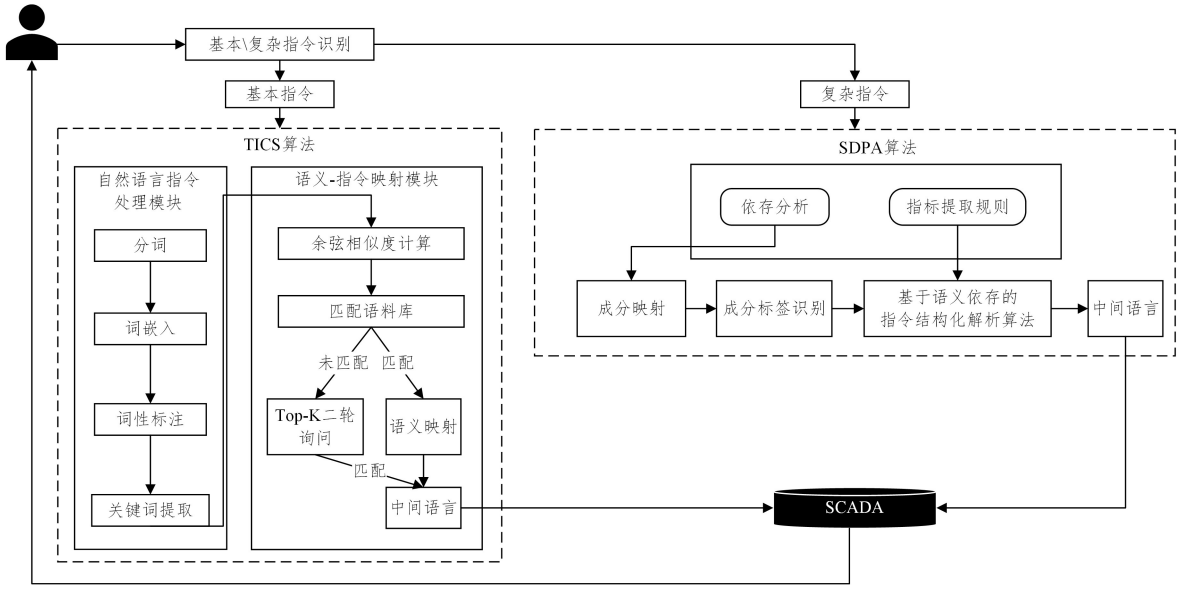

图 2　Text-to-SCADA 框架

Fig. 2　Text-to-SCADA framework

3.1　基本与复杂指令识别

基本自然语言指令可以理解为复杂自然语言指令的子指令。使用依存分析系统 DDParser 对自然语言指令进行词性标注。在词性标注结果中，标注为/c 的术语基本上是指令中的连词。并列连词连接单词、短语以及句子等，如“然后”“和”“或者”“所以”等。目前，在输入的自然语言指令中考虑了是否含有并列连词来确定是否存在子指令。在提出的方法中，如果找到“和”“或”“然后”这样的协调连词，则该指令被视为复杂自然语言指令。例如，如果输入的自然语言指令是“打开大厅的台灯和关闭机房的空调然后查看车床的温度”，则标记的指令如表 1 所列。

表 1　指令词性标注示例

Table 1　Example of command lexical annotation

子指令	标注情况
打开大厅的台灯和	打开/v 大厅/n 的/u 台灯/n 和/c
关闭机房的空调	关闭/v 机房/n 的/u 空调/nz
然后查看车床的温度	然后/c 查看/v 车床/n 的/u 温度/n

3.2　TICS 算法

自然语言指令经过分词与词嵌入、停止词过滤、词性标注和关键词提取 4 个步骤。流程如表 2 所列。

表 2　自然语言指令处理流程

Table 2　Processing flow of natural language instruction

Input		太热了，帮我打开大厅的空调
Step1	分词	太热/了/帮/我/打开/大厅/的/空调
Step2	停止词过滤	太热/打开/大厅/空调
Step3	词性标注	太热，a/打开，v/大厅，np/空调，ns
Step4	关键词提取	打开，v/大厅，np/空调，ns

3.2.1　自然语言处理

首先使用 Word2vec 中的 Skip-gram 模型完成对词语的分布式表示，实现词嵌入算法，要经过中文分词处理，依赖于中文分词工具 Jieba 实现。在经过分词之后，加载哈工大的停用词表，就可以过滤自然语言指令中价值低的字和词语。然后对指令进行词性标注，以限定词语的属性范围，改善关键词提取的效果。本文的词性标注以北大词性标注集为基准。本文定义了一个包含 98 个词的字典，用来匹配词性，在分词的同时完成词性标注。最后，在词性标注的基础上完成关键词提取。主要目的是抽取指令中描述被控对象的相关属性，通过停止词消除、词频-逆文本频率算法和词义相似度计算实现。词频（Term Frequency，TF），表示关键词 w 在文档 D_i 中出现的频率，计算公式为：

$$TF_{w_i D_i}=\frac{count(w)}{|D_i|}$$

其中，$count(w)$表示关键词 w 出现的次数，$|D_i|$为文档 D_i 所有词的数量。

3.2.2 语义-指令映射

(1)余弦词义相似度计算

本文借助 Word2vec 中的神经网络语言模型训练工农业领域中高频词汇的词向量，同时通过词向量间的余弦相似度对词语进行聚类，消除一义多词现象。对于词向量 x_i 和 y_i，其余弦相似度计算公式为：

$$\cos\langle x_i,y_i\rangle=\frac{\sum_{i=1}^{n}(x_i\times y_i)}{\sqrt{\sum_{i=1}^{n}x_i^2\times\sum_{i=1}^{n}y_i^2}}$$

其中，余弦相似度的取值区间为[0,1]。两个词向量夹角的余弦值越接近于 1，这两个词的语义就越接近。

(2)指令匹配与映射

对经过关键词提取的自然语言指令序列 $S=\langle x_1,x_2,\cdots,x_n\rangle$中的词语 x_i 处理，得到其向量形式，然后计算 x_i 与语料库中词语 y_i 的余弦相似度 $\cos\langle x_i,y_i\rangle$，并设定一个阈值 p，当 $\cos\langle x_i,y_i\rangle$超过该阈值时，就可以判断这两个词语是近义的。在匹配了语料库的指令关键词以后，需要转化成第 2 节中定义的中间语言数据结构，供 SCADA 系统读取。参数索引表如表 3 所列，其中包含了可被用户自定义的关键词和对应的参数值。用户可以使用这些关键词来查询对应的参数值，例如地点、对象、动作等。在实现过程中，可以通过调用相应的 HTTP 地址或 JavaScript 方法来进行查询或控制操作。

综上所述，算法 1 描述了 TICS 算法的语义解析过程。

算法 1 TICS 基本自然语言指令解析算法

输入：基本自然语言指令序列 String

输出：结构化键值对集合[{Object：word_1}，{Location：word_2}，{Action：word_3}]

```
1. RESULT={ Ø } /* 初始化一个空集合 */
2. StringList=Keyword_Extract(String,'TF') /* TF-IDF 关键词提取 */
3. FOR List IN StringList/* 遍历提取后的关键词列表 */
4.    IF List[0] IN Location_dict OR List[1]=='ns' /* 若关键词在位置属性词典或者词性为 ns，则认定该词语为位置属性 */
5.       Location ← List[0]
6.    ELSE IF cos_sim>=0.65
7.       Location ← List[0]
8.    END IF
9.    IF List[0] IN Object_dict OR List[1]=='nz'OR'nc'
10.      Object ← List[0]
11.   ELSE IF cos_sim>=0.65
12.      Object ← List[0]
13.   END IF
14.   IF List[0] IN Action_dict OR List[1]=='v'
15.      Action ← List[0]
16.   ELSE IF cos_sim>=0.65
17.      Action ← List[0]
18.   END IF
19. RESULT ←[{Object：word_1}，{Location：word_2}，{Action：word_3}]/* 更新结果 */
20. END FOR
```

表 3 参数索引表

Table 3 Parameter index

Keywords	ViewID
卧室	1
客厅	2
厨房	3
卫生间	4
…	…

3.3 SDPA 算法

SDPA 算法的解析目标是复杂自然语言指令，利用 DDParser 对复杂自然语言指令依存句法分析。将自然语言指令进行成分映射和成分标签识别，分析复杂指令中词原型之间的依赖关系，并且定义指标提取规则，从而实现复杂自然语言指令的语义解析。本文构建了一个自下而上的解析框架，如图 3 所示。

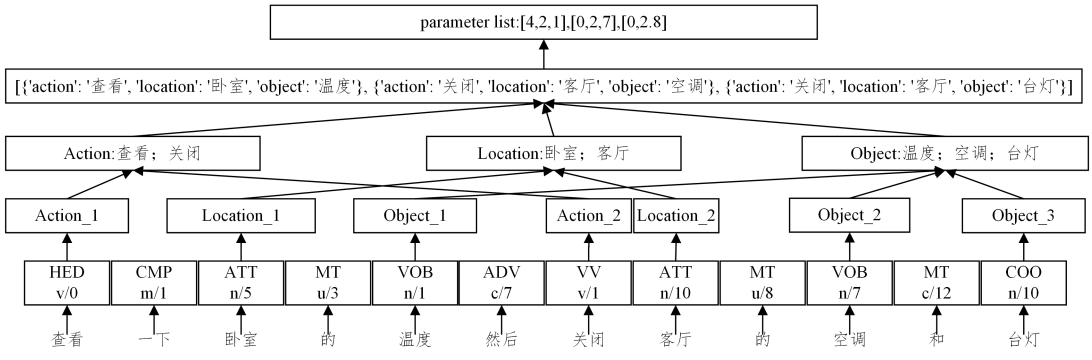

图 3 SDPA 解析框架

Fig. 3 SDPA parsing framework

3.3.1 依存句法分析

复杂自然语言指令“先打开大棚的空调然后关闭大棚的灯光”，生成的依存关系如列表[{'word'：['先'，'打开'，'大棚'，'的'，'空调'，'然后'，'关闭'，'田间'，'的'，'灯光']，'postag'：['d'，'v'，'n'，'u'，'n'，'c'，'v'，'n'，'u'，'n']，'head'：[2，0，5，3，2，7，2，10，8，7]，'deprel'：['ADV'，'HED'，'ATT'，'MT'，'VOB'，'ADV'，'VV'，'ATT'，'MT'，'VOB']}]。将自然语言指令成分映射为 4 个指标，即“word”“postag”“head”以及“deprel”，其依次表示自然语言指令分词后的结果、分词后的词性、词语语义之间的位置依赖关系以及语义关系。在对复杂自然语言指令进行归纳总结以后，发现指令中常出现的语义依存关系主要有 4 种：核心关系

(HED)、定中关系(ATT)、动宾关系(VOB)和并列关系(COO)。表 4 给出了自然语言常见依存关系的示例。

表 4　依存关系示例

Table 4　Example of dependency relationship

关系	描述	举例
HED	指整个句子的核心	打开卧室的台灯(ROOT,打开,HED)
ATT	定语与中心词之间的关系	关闭大棚的传感器(传感器,大棚,ATT)
VOB	宾语与谓语之间的关系	请帮我打开厨房的油烟机(打开,油烟机,VOB)
COO	同类型词语之间的关系	打开卧室的空调和台灯(空调,台灯,COO)

然后进行成分标签识别。在自然语言指令中,通常以动词为整个句子的核心,不依赖于任何词语,所以“打开”下面的数字为 0,即不与任何词语形成依存关系,定义为 ROOT 结点。同时,定中关系(ATT)描述的是定语与中心词之间的关系。在本例中,“客厅”下所对应的数字为 5,即与第 5 个单词“空调”构成定中关系(ATT),视“客厅”为“空调”的定语;同理,“卧室”下所对应的数字为 10,即与第 10 个单词“灯光”构成定中关系(ATT)。根据这一特性,可以选择提取定中关系(ATT)所对应词原型作为中间语言数据结构范式中的位置(Location)属性。同时,在示例中“空调”下所对应的数字为 2,即与第 2 个单词“打开”构成动宾关系(VOB);“灯光”下所对应的数字为 7,即与第 7 个单词“关闭”构成动宾关系(VOB)。根据这一特性,可以选择提取动宾关系(VOB)所对应的词原型作为中间语言数据结构中的对象(Object)属性。指令“打开厨房的油烟机和洗碗机”中的“油烟机”与“洗碗机”之间会构成并列关系(COO),即同类型词语之间的关系,因此在遇到这种情况时,可以将并列关系(COO)所对应的词原型进行提取,合并为对象(Object)属性。而针对动作(Action)属性,可以考虑动宾关系(VOB),找到宾语即对象属性以后,根据动词与宾语之间的依赖关系,提取宾语所依赖的关键词为动作属性,例如对于“关闭灯光”,“关闭”和“灯光”之间构成动宾关系,即可确定动作属性为“关闭”。经过以上步骤完成语法组合以后,会得到复杂自然语言’先打开客厅的空调然后关闭卧室的台灯’的中间语言数据结构范式,即[{‘action’:‘打开’,‘location’:‘客厅’,‘object’:‘空调’}{‘action’:‘关闭’,‘location’:‘卧室’,‘object’:‘灯光’}]。最终进行语义-指令映射,完成参数转换。其中,指令“先打开客厅的空调然后关闭卧室的灯光”和“打开厨房的油烟机和洗碗机”的依存关系树如图 4 所示。

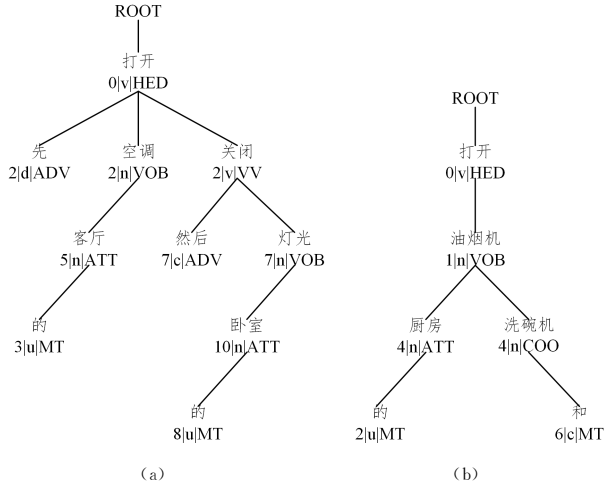

图 4　依存句法图

Fig. 4　Dependency syntax diagrams

3.3.2　关键属性提取规则

通过依存句法分析以及词性特征提取复杂自然语言指令中的关键属性,并将其填充上文定义的中间语言数据结构{‘action’:‘’,‘location’:‘’,‘object’:‘’}中去。提取步骤如下:

(1)生成复杂自然语言指令的依存分析结果;

(2)遍历依存分析结果,根据语义关系以及词性特征提取关键指标;

(3)将其自动填充到事先定义好的中间语言数据结构形式中去。

本文利用语义特征进行关键属性提取时,主要寻找的语义关系有 3 种:定中关系(ATT)、动宾关系(VOB)、并列关系(COO)。在进行关键指标提取的时候,需要遵循以下规则。

规则 1　指令中存在多个动词的时候,动词具有核心关系(HED)和连谓关系(VV),不利于直接提取,因此首先寻找动宾关系(VOB),根据语言习惯以及词性特征,借助宾语找到对应的动词。

规则 2　定语主要为修饰作用,本文考虑的是负责操纵指令的语义解析。通常其中代表定中关系(ATT)的是位置信息,因此可以将其提取出来作为位置属性。

规则 3　并列关系(COO)在操控指令中起到的作用相同,即操控对象的动作一致。针对该种关系,使其操控动作和位置信息保持一致,与动宾关系(VOB)对应的词原型可以进行合并。

基于以上规则,算法 2 描述了 SDPA 算法的语义解析过程。

算法 2　SDPA 复杂自然语言指令解析算法

输入:复杂自然语言指令依存关系字典 ParserDict,包含{word,postag,head,deprel}。其中,word 表示词原型,postag 表示词性,head 表示两个词语之间的位置依赖关系,deprel 表示两个词语之间的依存关系

输出:结构化键值对集合[{Object:word_1},{Location:word_2},{Action:word_3}]

```
RESULT={ Ø }/*初始化一个空集合*/
number=-1/*初始化计数变量*/
WordList=ParserDict['word']
PostagList=ParserDict['postag']
HeadList=ParserDict['head']
RelationList=ParserDict['deprel']
FOR relation IN RelationList
    number=number+1
    IF relation=='ATT' /*遍历定中关系*/
        Location ← WordList[number]
        Object ←WordList[HeadList[number]-1]
    ELSE IF
    (relation=='VOB') OR(relation=='COO')
```

```
13.     IF relation==‘VOB’
14.       Action←WordList[HeadList[number]-1]
15.       ELSE Object←WordList[number]
16.     END IF
17.   END IF
18. RESULT←[{Object: word_1}, {Location: word_2}, {Action: word_3}]/*更新结果*/
19. END FOR
```

3.3.3 解析算法优化策略

通过分析复杂自然语言在进行依存句法分析中经常出现的噪声数据，将提取失败的结果自动剔除。汉语本身十分复杂，同一个词语在不同语境中形成的依存关系可能有所不同，例如“请帮我打开卧室的台灯”和“打开卧室的台灯”，这两条自然语言指令所表达的含义完全一致，但是关键词“打开”所表示的依存关系却不同。在第一条指令中，“打开”与“帮”构成双宾语关系(DOB)。在第二条指令中，“打开”和“ROOT”构成核心关系(HED)。同样，在同一条自然语言指令中，同一种依存关系所对应的词原型也可能不一致，例如“请帮我分别打开卧室的台灯和客厅的空调”中的“帮”和“台灯”均与其他词语构成动宾关系(VOB)。此外，人类在说话时会掺杂一些其他词汇，例如“打开卧室的扫地机器人，卫生需要保持干净”，经过算法 2 的结果为：[{‘action’:‘打开’,‘location’:‘卧室’,‘object’:‘扫地机器人’},{‘action’:‘打开’,‘location’:‘卧室’,‘object’:‘保持’]。可见第二个字典为多余的，且将“保持”误认为对象(object)属性。主要原因在于“保持”与“需要”构成动宾关系(VOB)，并且“扫地机器人”与“打开”也构成动宾关系(VOB)。算法 2 会误将“保持”认定为对象属性并完成指标提取。因此在优化算法模块，需要自动删除结构化结果中多余无效的信息，并且结合词性特征将错误提取信息剔除，从而提高指令解析的准确率，增强依存句法分析的指令结构化解析算法的科学性、适用性以及可拓展性，使得 SDPA 算法在复杂自然语言指令解析上具有更高的准确率。

算法 3 复杂指令解析结果优化算法

```
输入：RESULT
输出：优化后的 RESULT
1. i=0
2. Object_value={ Ø }/*初始化一个空列表*/
3. IF Action !=Ø AND Location !=Ø AND Object !=Ø
4. FORitem IN RESULT
5.     Object_value←item(key_value)/*将表示对象属性的指标值存放在列表里面*/
6. END FOR
7. FOR string1 IN WordList
8.     i=i+1
9. FOR string2 IN Object_value
10.   IF string2==string1
11.     IF(PostagList[i-1]!=‘nz’) AND (PostagList[i-1] !=‘n’)
12.       Delete(dict←object)/*对象属性如果不是 nz 或者 n，则将该对象所在的字典删除*/
13.     END IF
14.   END IF
15. END FOR
16. RESULT ← RESULT
17. RETURN RESULT
18. END FOR
```

4 实验设置及结果分析

4.1 测试数据集构造

Text-to-SCADA 是面向特定领域的自然语言指令解析框架。针对本文提出的应用领域，我们构建了中文测试指令数据集——CNLQTS。关于语料库的收集，不是采用对话文本的直接整合，而是预先定义好 SCADA 系统中可能出现的各种指令，然后通过科研工作人员众包的形式展开成自然的、口语化的句子。这种搭建方法能够有效地提高 Text2SCADA 理解特定领域中自然语言指令的效果。具体做法为：首先，限定一个符合 SCADA 系统中间语言格式所提到的中间语言规范的指令{‘object’:‘空调’,‘location’:‘机房’,‘action’:‘打开’}；然后，将伪语言问题描述改写为自然语言问题 E=“打开机房空调”，并将语句 E 扩充为更加复杂、口语化的自然语言语句。我们在实验室召集了 20 个硕士生，花费了 1 个月众包了该数据集。语料库中的一个案例如表 5 所列。

表 5 语料库示例

Table 5 Corpus examples

中间语言		自然语言
{object=空调,location=机房,action=打开}	语句①	帮我打开机房空调
	语句②	把机房的空调打开
	语句③	请去把机房的空调打开

最终，CNLQTS 数据集提供了来自不同领域的 600 条自然语言指令集合，包括农业领域(200)、工业领域(200)、智能家居领域(200)。每个领域均包含 100 条基本自然语言指令和 100 条复杂自然语言指令。使用如表 6 所列的各种类型数据进行实验测试。

表 6 CNLQTS 数据集示例

Table 6 Example of CNLQTS dataset

领域	指令示例
农业	大棚的温度是多少？ 请帮我查看田间的甲醛含量是多少？ 打开大田的灯光，然后查看一下温室的温度。 请帮我分别打开大棚的灯光和空调。
工业	机房现在甲醛浓度是多少？ 中控机房的二氧化碳浓度是多少？ 先打开机房的吸尘机再关闭无尘室的空调。 请帮我查看一下中控机房的甲醛浓度。 查看位于广大东汽机器人 A1 轴电流。 查看位于二重的工业机器人工作站用电量。
智能家居	打开厨房的油烟机。 查看卫生间的温度是多少？ 先打开卧室的台灯，再关闭卫生间的浴霸。 打开客厅的扫地机器人和吊灯。

4.2 Text-to-SCADA 框架评估指标

Text2SCADA 性能的评价方法主要包括 3 个：精确率 Precision、召回率 Recall 以及 F-score 值。本文通过手工判定方式获取。构建的数据集总共包含 600 条自然语言指令，考虑指令的数量对实验结果的影响，根据具体实验，选择不同数量的指令集来对算法进行性能测试，并且以手工判定的方式获取精确率 P、召回率 R 以及 F 值。接下来给出相关变量的定义：A 表示正确识别的指标以及对应指标值个数；B 表示

结构化结果的记录总数；C 表示原自然语言指令中包含指标的总数。相关计算公式如下：

$$P=\frac{A}{B}$$

$$R=\frac{A}{C}$$

$$F=\frac{2\times P\times R}{P+R}$$

4.3　训练和测试

4.3.1　词嵌入

采用 skip-gram 模型，损失函数为交叉熵，词向量维度设置为 150，窗口大小为 8，采用随机梯度下降法进行训练，训练过程中的 loss 曲线如图 5 所示。

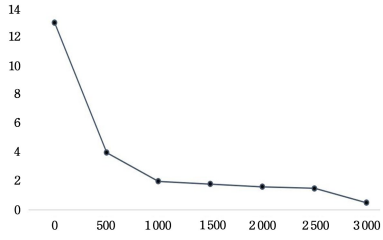

图 5　训练 Loss 曲线

Fig. 5　Loss curve

得到中文词语的向量形式以后，就可以通过公式计算输入词和语料库中各个例句的相似度，并取相似度最高的、同词性的词匹配预料库中的词语；再通过语义-指令映射模块检索指令描述的对象、地点等属性，将其映射成 http 地址或调用 JavaScript 方法执行指令。

4.3.2　基本自然语言指令解析算法测试实验

为了评估 TICS 基本自然语言指令解析算法的性能，使用 CNLQTS 数据集汇总的基本自然语言指令进行实验。采用控制变量以及随机分配的方法，依次输入 50，100，150，…，300 条指令进行测试，得到的评估指标值和返回结果的最长时间如表 7 所列。图 6 能够清晰反映出各项评估指标的变化情况。

表 7　基本自然语言指令测试结果

Table 7　Basic natural language instruction test results

No.	*Number*	*Precision*/%	*Recall*/%	*F-score*/%	*Time*/s
1	50	92.15	94.25	93.19	0.82
2	100	91.57	92.14	92.85	0.74
3	150	91.24	91.71	91.47	0.67
4	200	90.25	91.14	90.19	0.89
5	250	89.65	88.54	89.06	0.87
6	300	87.58	87.91	87.73	0.73
Avg		**90.37**	**90.79**	**90.58**	**0.79**

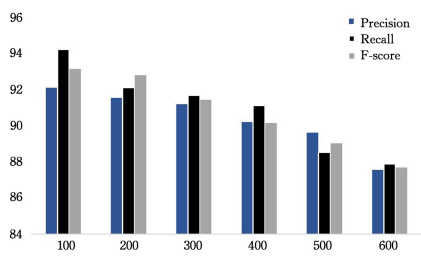

图 6　TICS 性能评价柱形图

Fig. 6　Bar chart of TICS performance evaluation

(1)在测试集上，三大评估指标的平均值分别 90.37%，90.79%，90.58%，均在 90%以上，都取得了不错的效果。

(2)随着指令数量的增多，评估指标值均呈现下降趋势，其下降的幅度在 2%到 5%之间，属于可接受范围。

(3)平均响应最长时间大约为 0.79 s，响应时间能够满足实时控制和查询功能，符合人机交互的实时性。

(4)平衡的 F 值最初表示良好的精度-召回性能，性能随着实验中考虑的自然语言指令数量的增加而恶化。

4.3.3　复杂自然语言指令解析算法测试实验

为了提高复杂自然语言指令结构化解析算法的科学性、适用性以及可拓展性，本文在 SDPA 算法后增加了结果优化算法。本节依旧采用控制变量以及随机分配的方法，依次输入 50，100，150，…，300 条指令进行测试，得到的评估指标值和返回结果的最长时间如表 8 所列。图 7 能够清晰反映出各项评估指标的变化情况，其中 $P1$，$R1$，$F1$ 表示优化前的性能评价指标，$P2$，$R2$，$F2$ 表示优化后的性能评价指标。

表 8　复杂自然语言指令测试结果

Table 8　Complex natural language instruction test results

No.	*Number*	*Precision*/%	*Recall*/%	*F-score*/%	*Time*/s
1	50	89.14	87.64	88.59	1.12
	95.10	**93.27**	**94.18**	**1.47**	
2	100	80.09	81.36	80.66	1.46
	89.81	**90.65**	**90.22**	**1.48**	
3	150	78.02	79.37	78.68	1.47
	87.58	89.91	88.73	1.48	
4	200	76.24	77.36	76.80	1.52
	85.43	**86.67**	**86.05**	**1.48**	
5	250	74.21	74.49	74.35	1.57
	84.57	**85.56**	**85.06**	**1.47**	
6	300	71.17	71.29	71.23	1.70
	82.29	**84.41**	**83.34**	**1.49**	
Avg		78.15	78.59	78.37	1.47
		87.63	**88.41**	**87.93**	**1.47**

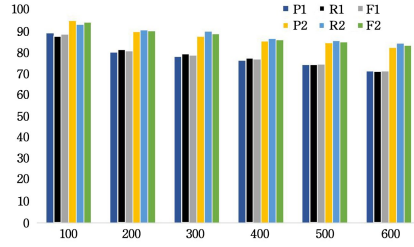

图 7　SDPA 性能评价柱形图

Fig. 7　Bar chart of SDPA performance evaluation

(1)无论是 SDPA 算法单独在复杂指令测试集上进行实验，还是优化后的 SDPA 算法在复杂指令测试集上进行实验，都取得了不错的效果，其中平均 $P1$，$R1$ 达到了 78.15%和 78.59%，$P2$，$R2$ 达到了 87.63%和 88.41%。

(2)随着复杂控制指令数量的增多，算法性能均有一定程度的下降。未优化前的 F 值下降幅度分别为 7.93%，1.98%，1.88%，2.45%以及 3.12%，平均下降幅度为 3.47%。优化后的 F 值下降幅度分别为 3.96%，1.49%，2.68%，0.99%以及 1.72%，平均下降幅度为 2.17%。明显看出，优化后算法的平均下降幅度明显小于优化前的算法，说明了优化算法在复杂自然语言指令解析的过程中起了很大的作用，提升了算法的可拓展性以及科学泛化能力。

(3)其中平均响应时间约为 1.474 s，满足人机交互的实时性要求。

4.3.4 Text2SCADA 组合服务性能测试实验

本节选取 600 条测试指令，采用随机分配以及控制变量的方法，分别将 100，200，300，…，600 依次输入带 Text2-SCADA 的系统中，实验结果如表 9 所列。图 8 能够反映各项评估指标的变化情况。

表 9 组合算法服务性能测试

Table 9 Combined algorithm service performance testing

No.	*Number*	*Precision*/%	*Recall*/%	*F-score*/%	*Time*/s
1	100	98.26	97.37	97.81	1.521
2	200	94.62	93.31	93.96	1.571
3	300	91.23	90.39	90.81	1.585
4	400	85.23	84.91	85.07	1.597
5	500	83.58	85.91	84.73	1.619
6	600	82.68	83.78	83.23	1.662
Avg		**89.27**	**89.28**	**89.27**	**1.593**

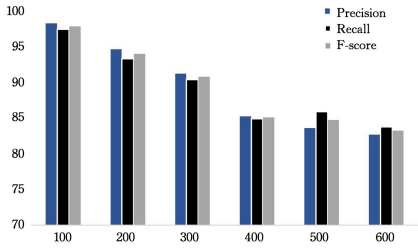

图 8 组合算法性能评价

Fig. 8 Combinatorial algorithm performance evaluation

(1)Text2SCADA 在指令测试集下的平均 *Precision* 值和 *Recall* 值分别为 89.27%和 89.28%，整体在 90%附近，准确性较高。

(2)*F* 值逐渐下降，下降幅度依次为 3.91%，3.15%，5.74%，0.34%，1.5%，平均下降幅度为 2.93%。这主要是因为随着测试指令数量的增加，其测试效果会出现一定程度的恶化，从而导致 *F* 值逐渐下降。

(3)Text2SCADA 的平均响应时间约为 1.593s，对比我们在现实世界中实际考察的平均响应时间为 3s 有了很大的提升，满足人机交互实时性的要求。

4.3.5 对比实验

随机抽取 300 条指令，分别使用 TF-IDF 关键词提取算法，基于 Word2vec 的相似度算法、基于 LSTM 句子相似度的算法和模板匹配的方法和本文算法进行对比实验。余弦相似度阈值为 0.65。以成功识别率为测试指标，分子为成功识别指令的个数，分母是总指令条数，具体结果如表 10 所列。

表 10 对比实验

Table 10 Comparison experiments

Method	Success rate
TF-IDF	38.67%(116/300)
Word2vec	40.33%(121/300)
LSTM	43.67%(131/300)
Template Matching	63.67%(191/300)
Our Method(TICS)	**70.33%(211/300)**
Our Method(SDPA)	**85.67%(257/300)**
Our Method(TICS+SDPA)	**96.67%(287/300)**

TF-IDF 关键词提取算法，可以将指定关键词进行信息抽取，但是并未考虑一义多词的情况，同义词之间为构建联系，所以只能识别 116 条。Word2vec 方法主要是计算词语之间的相似度，但是无法提取合适的关键词，也只能识别基本自然语言指令中的部分指令。LSTM 以句子为单位，计算句子意思，识别情况有一定的提升，但是信息抽取能力一般。人工的模板匹配方式是基于人工规则，根据经验编写提取算法，有着不错的成功识别率，但是该方法需要耗费大量人力阅读文本，不适用于实际人机交互系统。TICS 是关键词提取和余弦相似度的结合，在关键词抽取的基础上利用了词义信息，因此成功率达到了 70.33%，但是无法对复杂长指令进行解析。SDPA 算法是针对复杂自然语言指令，运用语义依存遍历结点的思想，识别率高达 85.67%。最后是整体算法的测试 TICS+SDPA，根据词性标注是否含有连词来区分基本自然语言指令和复杂自然语言指令，以此来推荐不同的算法进行语义解析。Text2SCADA 的整体可拓展性和科学泛化能力得到了较大的提升，指令识别成功率高达 96.67%。

结束语 本文提出了一个面向分布式 SCADA 系统的自然语言接口(Text-to-SCADA)，该接口通过词性标注来区别基本自然语言指令和复杂自然语言指令，针对基本自然语言指令采用 TICS 算法进行解析，针对复杂自然语言指令采用 SDPA 算法进行解析，以此来理解用户输入的自然语言指令；通过众包的形式提供了一个 CNLQTS 数据集来测试接口的性能。之后会研究更加复杂的自然语言指令，并且丰富中间语言格式；此外，未来工作中将会考虑图神经网络的形式来对指令进行编码以搭建知识图谱的方式来提升复杂查询的准确率，并支持更高的可拓展性；也会对算法的复杂度进行优化，缩短接口的服务响应时间，提升人机交互的实时性。总之，我们的方法为工农业信息化管控提供了更为便捷的交互手段。

参 考 文 献

[1] BELLEGARDA J R, SILVERMAN K E A J I T O S, PROCESSING A. Natural language spoken interface control using data-driven semantic inference[J]. IEEE Transactions on Speech and Audio Processing, 2003, 11(3): 267-277.

[2] ZHENG Z, ZHAI M, PENG H, et al. Architecture and key technologies of distributed SCADA system for power dispatching and control[J]. Dianli Xitong Zidonghua/Automation of Electric Power Systems, 2017, 41: 71-77.

[3] SU Y, AWADALLAH A H, KHABSA M, et al. Building natural language interfaces to Web APIs[C]// Proceedings of 26th ACM International Conference on Information and Knowledge Management(CIKM 2017). Singapore, 2017: 177-186.

[4] MIN Q, SHI Y, ZHANG Y. A pilot study for Chinese SQL semantic parsing[C]// Proceedings of 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, EMNLP(IJCNLP 2019). Hong Kong, 2019: 3652-3658.

[5] CHU E T H, HUANG Z Z. Dbos: A dialog-based object query system for hospital nurses[J]. Sensors(Switzerland), 2020, 20, 1-15.

[6] SETYAWAN M Y H, AWANGGA R M, EFENDI S R. Comparison Of Multinomial Naive Bayes Algorithm And Logistic Regression For Intent Classification In Chatbot[C]// Proceedings of 2018 International Conference on Applied Engineering (ICAE 2018). Batam, Indonesia: 2018 IEEE Indonesia CSS/RAS Joint Chapter. 2018.

[7] LE Q, MIKOLOV T. Distributed representations of sentences

and documents[C]// Proceedings of 31st International Conference on Machine Learning(ICML 2014). Beijing,2014:2931-2939.

[8] PENNINGTON J,SOCHER R,MANNING C D. GloVe:Global vectors for word representation[C]// Proceedings of 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014). Doha,2014:1532-1543.

[9] DEVLIN J,CHANG M W,LEE K,et al. BERT:Pre-training of deep bidirectional transformers for language understanding [C]// Proceedings of 2019 Conference of the North American Chapter of the Association for Computational Linguistics:Human Language Technologies(NAACL HLT 2019). Minneapolis,2019:4171-4186.

[10] HOCHREITER S,SCHMIDHUBER J J N C. Long Short-Term Memory[J]. Neural Computation,1997,9(8):1735-1780.

[11] CHO K,VAN MERRIENBOER B,GULCEHRE C,et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[C]// Proceedings of 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014). Doha,2014:1724-1734.

[12] LECUN Y,BOTTOU L,BENGIO Y,et al. Gradient-based learning applied to document recognition[C]// Proceedings of the IEEE. 1998:2278-2323.

[13] ZHOU P,SHI W,TIAN J,et al. Attention-based bidirectional long short-term memory networks for relation classification [C]// Proceedings of 54th Annual Meeting of the Association for Computational Linguistics(ACL 2016). Berlin,2016:207-212.

[14] KHOMENKO V,SHYSHKOV O,RADYVONENKO O,et al. Accelerating recurrent neural network training using sequence bucketing and multi-GPU data parallelization[C]// Proceedings of 1st IEEE International Conference on Data Stream Mining and Processing(DSMP 2016). Lviv,2016:100-103.

[15] VERLEYSEN M,FRANCOIS D. The curse of dimensionality in data mining and time series prediction. In Proceedings of 8th International Workshop on Artificial Neural Networks[C]// Computational Intelligence and Bioinspired Systems(IWANN 2005). Vilanova i la Geltru,2005:758-770.

[16] GOODMAN B A,GROS Z,et al. Research in knowledge representation for natural language communication and planning assistance[R]. 1988.

[17] BENGIO Y,DUCHARME R,VINCENT P,et al. A Neural Probabilistic Language Model[J]. Journal of Machine Learning Research,2003.

[18] SOWMYA KAMATH S,ANANTHANARAYANA V S. Discovering composable web services using functional semantics and service dependencies based on natural language requests[J]. Information Systems Frontiers 2019,21:175-189.

[19] TIAN C Y,CHEN D H,WANG M,et al. Structured Processing for Pathological Reports Based on Dependency Parsing[J]. Journal of Computer Research and Development,2016,53:2669-2680.

[20] XU X,LIU C,SONG D. SQLNet:Generating Structured Queries From Natural Language Without Reinforcement Learning [J]. arXiv:1711.04436,2017.

[21] YU T,LI Z,ZHANG Z,et al. TypeSQL:Knowledge-based type-aware neural text-to-SQL generation[C]// Proceedings of 2018 Conference of the North American Chapter of the Association for Computational Linguistics:Human Language Technologies. 2018:588-594.

[22] HWANG W,YIM J,PARK S,et al. A Comprehensive Exploration on WikiSQL with Table-Aware Word Contextualization [J]. arXiv:1902.01069,2019.

WANG Tao, born in 1982, master. His main research interests include industrial internet and internet of things.