·体系结构与软件技术· 文章编号：1000-3428(2018)11-0046-10 文献标志码：A 中图分类号：TP391

基于语义要素组合的知识库问答方法

刘飞龙，郝文宁，余晓晗，陈 刚，刘 冲

（陆军工程大学 指挥信息系统学院，南京 210000）

摘 要：为进行知识库问答系统中的问题语义分析，提出基于语义要素组合的问答方法。采取词库识别和规则识别提取问题语义要素并依据预定义模式识别函数，结合问题依存分析树结构和函数类型组合语义要素，生成问题语义表达式后利用映射和联合消歧组成知识库语义表达式，然后将知识库语义表达式转换为 SPARQL 语句后执行问答。实验结果表明，该方法的 F1 平均值为 0.841，能够有效理解并分析问题语义。

关键词：问答系统；问题理解；语义要素组合；联合消歧；语义表示

中文引用格式：刘飞龙，郝文宁，余晓晗，等. 基于语义要素组合的知识库问答方法[J]. 计算机工程，2018，44(11)：46-55.

英文引用格式：LIU Feilong，HAO Wenning，YU Xiaohan，et al. Knowledge base question answering method based on semantic element combination[J]. Computer Engineering，2018，44(11)：46-55.

Knowledge Base Question Answering Method Based on Semantic Element Combination

LIU Feilong，HAO Wenning，YU Xiaohan，CHEN Gang，LIU Chong

（Institute of Command Information System，Army Engineering University of PLA，Nanjing 210000，China）

【Abstract】In order to analyze the question semantics in the knowledge base question answering system，a question answering method based on semantic element combination is proposed. Firstly，the semantic elements of the problem are extracted by lexicon recognition and rule recognition，and recognize functions based on predefined patterns. Then the semantic elements of the problem are combined with the structure of the problem dependency analysis tree and the type of the function to generate the semantic expression of the problem，then the semantic expression of the knowledge base is formed by mapping and joint disambiguation. Finally，the semantic expression of the knowledge base is converted to SPARQL statement and execute question and answer. Experimental results show that，the average F1 value of the method is 0.841，which can effectively understand and analyze the problem semantics.

【Key words】question answering system；question understanding；semantic element combination；joint disambiguation；semantic representation

DOI：10.19678/j.issn.1000-3428.0048633

0 概述

随着万维网的迅速发展，大规模知识库不断出现，如 Yago[1]、DBpedia[2]和 Wikidata[3]等。这些知识库覆盖面广、数据量大，拥有庞大的结构化数据，蕴含大量的知识。为能准确高效地获取这些知识并充分利用结构化数据的优势，专门的知识库查询语言（如 SPARQL[4]）应运而生。但是，只有专业的研究人员才能熟练地掌握该语言的语法，多数普通用户仍通过口语化的问题执行检索来获取知识。由于自然语言具有模糊性和复杂性，因此机器很难准确地理解用户的意图，导致其查询结果不尽人意。因此，如何让多数普通用户便捷地获取知识库中的知识，已成为一个亟待解决的问题。基于知识库的问答系统允许用户使用自然语言描述问题，通过后台问题理解、信息检索和答案生成，返回给用户准确、简洁的知识库知识，这能够降低知识库查询的门槛，提升结构化知识的利用率，因此，该系统成为当前研究的热点。

为深层次获取用户的查询意图，通过对当前知识库问答系统的分析，本文提出基于语义要素组合的知识库问答系统。围绕用户提出的自然语言问题，从问题的依存分析结构入手，首先识别问题中概念、实体、关系、属性、属性值和问题所对应的函数类

基金项目：国家自然科学基金青年基金（71501186）。

作者简介：刘飞龙（1994—），男，硕士，主研方向为军用数据与知识工程；郝文宁，教授；余晓晗，讲师；陈 刚，教授；刘 冲，硕士。

收稿日期：2017-09-11 **修回日期：**2017-11-07 **E-mail：**lflgot@foxmail.com

别(预定义),之后依据由依存分析结构和预定义组合规则组合识别到的语义要素,生成结构化的问题语义表达式,然后依据预先构建的映射集映射语义要素到底层知识库,使用消歧图对映射结果进行联合消歧后得到由知识库元素表示的语义表达式,最后将该语义表达式转换为 SPARQL 语句,在知识库上查询并返回结果。

1　相关研究

早期的知识库问答主要采取信息检索的方法。首先抽取问题中的关键词集合,然后利用该关键词集合在知识库中检索可能的候选答案。但是,简单的关键词集无法准确地表达问题真正的含义,且难以理解较复杂的问题。为提升知识问答的精度,研究人员提出多种解决方法。从已有研究成果来看,知识库问答系统主要包括 3 种类型:基于模板的方法,基于信息抽取的方法,基于语义分析的方法。

基于模板的知识问答方法[5-7]以句型模板为核心,首先将问题和模板库中预构建的句型模板进行匹配,然后依据匹配到的句型模板直接提取问题的语义信息,最后依据该句型模板对应的查询语句执行查询。相比其他方法,该方法不涉及复杂的语义分析,能够直接分析用户的查询意图从而得出语义信息,但是其需要预先构建足够数量的模板,即知识问答的效果受制于句模库的数量和质量。

基于信息抽取的知识问答方法,首先抽取问题中的核心实体,然后在知识库中定位核心词,抽取以核心词为中心的周围几跳之内的实体和关系形成知识库子图,子图中的节点和边中包含着候选答案,最后依据规则和模板等构建问题特征向量和知识库子图特征向量,通过机器学习的方法进行候选答案的筛选从而得出最终答案。文献[8]将问题表层语言信息和知识库子图(其文中称主题图)中节点和边作为特征,以 F1 正则化逻辑回归分类器进行判定,从主题图中选择答案节点。文献[9-10]引入向量建模技术,直接将问题以及和问题相关的候选答案三元组转换为低维向量,结合正确的问答对进行训练,选择相似度最高的候选答案返回。文献[11]提出基于查询图的图匹配方法,首先结合句法分析和关系抽取提取问题当中的三元组并映射成知识库三元组,然后利用指代消解将三元组连接成查询图,最后依据子图同构在知识库中查询匹配子图并返回答案。然而,大范围的图匹配查询空间过大且效率较低,极大地影响到系统的性能。从问答表现来看,该类知识问答方法只能检索知识库中存在的结果,不能理解需要计算或者推理的问题,若查询的结果不是知识库中存在的元素,则无法执行推理且不能计算得到正确答案。

基于语义分析的知识问答方法以规范化的语义表达式为核心,首先利用预先设计的语义分析方法解析问题语义,将问题转换为规范化的语义表达式,然后将该语义表达式转化为结构化查询语句执行检索获取答案。其采取逻辑化的规范语义表示式作为对问题的解析形式,逻辑关系清晰,能够推理并解决复杂问题。传统的语义分析[12-13]依赖于人工标注的逻辑词表,只能在小范围内进行有监督的机器学习,当遇到在监督学习中没有学习过的样本时就很难处理。为扩展其应用范围,文献[14]建立半监督模型以扩展语义分析器,其首先通过预定义的逻辑词表训练有监督的语义分析器,然后通过词汇扩展原有语义分析器中的词汇,进而提升语义分析器的适用范围,但是,其仍然依赖人工标注的逻辑词表。文献[15]提出摆脱人工标注的语义分析方式,其使用 DCS 语言得到问题的多个语义表达式,如设计特征、损失函数,利用已有的问题-答案对训练选择模型,选择能得到正确答案的语义表达式。该方法的正确率依赖于问题-答案对的数量和质量,逻辑语义表达式构建过程不够精细,产生的逻辑表达式不准确,对函数类问题描述过于简略,且缺乏专门的映射消歧过程,导致难以理解复杂的用户问题。

相较于基于关键词匹配和信息抽取的方法仅能浅层理解简单的用户问题,本文基于语义要素组合的知识问答方法借助逻辑表达式,能够深入理解复杂用户问题语义,并且不依赖大量人工构建的句模库。相较于基于逻辑词表和问题-答案对的语义分析方法,本文方法通过语义要素分析、语义要素组合 2 个阶段直接生成规范化的问题语义表达式,既能够避免逻辑词表对系统应用范围的限制,也不依赖于难以收集的问题-答案对训练模型。

2　知识库问答方法整体结构

给定一个问答系统中的自然语言问题集 Q:$\{q_i \mid q_i \in Q\}$ 和知识库 K,本文方法接收任意输入的 q_i,经过问题理解输出该问题对应的语义表达式 q_{NLi},通过联合消歧生成由知识库元素表示的语义表达式 q_{FLi},最后将 q_{FLi} 转化为 SPARQL 查询语句 s,用 s 在知识库上查询并输出答案。本文基于语义要素组合的知识库问答方法流程如图 1 所示。

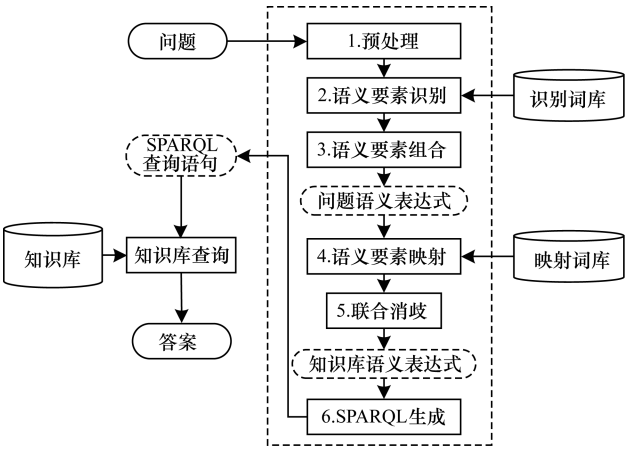

图 1 基于语义要素组合的知识库问答方法流程

该方法流程分为以下 6 个阶段：

1）预处理：执行分词、修正、词性标注、去停用词、依存分析。

2）语义要素识别：采取基于词库的语义要素识别和基于规则的语义要素识别，抽取问题中的语义要素并标记其类型。

3）语义要素组合：首先识别问题中包含的函数类型，然后依据基本语义要素组合规则和函数对应的语义要素组合规则组合语义要素，生成问题语义表达式。

4）语义要素映射：利用预先构建的自然语言词汇-知识库元素之间的映射关系，将识别出的问题语义要素映射到知识库元素。其中，一个语义要素可以对应多个知识库元素。

5）联合消歧：通过构建消歧图进行联合消歧，将上述语义要素和知识库元素之间的一对多关系消歧为一对一，生成知识库语义表达式。

6）SPARQL 生成：将知识库语义表达式转换为 SPARQL 查询语句。

3 知识库问答方法过程分析

3.1 相关概念

本文涉及的相关概念具体定义阐述如下：

定义 1（语义要素）结合当前结构化知识库的数据类型，本文以 E、C、A、R、V 分别表示知识库中的实体、概念、属性、关系、属性值。其中，根据属性值的类型不同，用 V_d 和 V_o 分别表示数值型属性值和对象型属性值（$\{V_d,V_o\}=V$）。在此前提下，本文定义如下 2 类语义要素形式（共用到 $\{e、p、E、C、A、R、V、V_d、V_o\}$ 9 种标记形式）：

1）一元语义要素：e 和 V。其中，$e\in\{E,C\}$。

2）二元语义要素：p。其中，$p\in\{A,R\}$，A 和 R 间的主要区别是 $A=\{A\mid<e,A,V>\}$，$R=\{R\mid<e,R,e>\}$。

从本质上而言，若将知识库三元组表示为 < 主体，“关系”，客体 >，则一元语义要素是集合 { 主体，客体 }，二元语义要素则是集合 {“关系”}（此处“关系”为广义含义，包括属性和关系）。

定义 2（问点块）用户问题包含疑问词但不包含语义要素的连续字符块。经过对大量用户问题的统计分析，问点块中所涉及的疑问词分为以下 3 类：

1）无意义疑问词：{“怎么样”“如何”“怎么”“怎样”“那么”“这么”“多么”}。

2）有意义疑问词：{“多少”“多”“几”“什么”“啥”“谁”“那”“哪”}。

3）原因疑问词：{“为什么”“为何”}。

3.2 预处理

在对问题进行语义分析前，首先要对问句进行预处理。预处理的目的是获取问题的句法结构、分析词性和词汇之间的依存关系、修正不规范的用户描述形式（如在问题中穿插拼音、中英文混用、阿拉伯数字中文写法）、去除对理解句子没有实际帮助的停用词等。

3.3 语义要素识别

语义要素识别的目的是抽取问题中的语义要素，标记其类型，为后续的语义要素组合提供支撑。通常的识别方法有 2 种：

1）构建相关词库进行识别。该方法的优点在于只要词库信息足够全面，其识别准确率可达任意精度，缺陷在于难以无限制地扩充词库。

2）通过制定规则、算法进行识别。该方法的优点在于通用性，其可以识别复杂的语义要素，覆盖率较高，识别类型广，缺点在于精度受限。

本文采取词库和规则相结合的方法进行语义要素识别，首先由词库识别语义要素，对于词库识别失败的词汇则通过规则识别。该方法能有效保证识别的精度和覆盖率。

3.3.1 词库识别方法

词库识别是语义要素识别的第 1 个步骤，根据待识别的语义要素类型建立各自的词库。本文涉及的词库有以下 3 种：

1）领域词库。这是词库最重要的组成部分，由知识库中元素抽取而来，包括实体库、概念库、关系库、属性库、属性值库，分别存放知识库中实体、概念、关系、属性、属性值，其中，属性值库主要是对象型属性值库。

2)同义词库。该词库对领域词库进行同义扩展。由于领域词库中的元素来自于知识库,其结构性较强且形式单一,而在现实用户表述的问题中,经常会用到同义表达,因此,需要为领域词库进行同义扩充,提升识别的覆盖率和精度。

3)用户词库。该词库对领域词库进行别名扩展。部分用户在提出问题时会夹带缩略词、简写或者别名,如“歼轰-7”缩写为“JH-7”,别名“飞豹”。因此,需要预先搜集相关的缩略词、简写、别名,建立领域词库对应的用户词库。

词库识别语义要素的主要步骤是:接收预处理后的问题,依据建立的词库分别识别并标记{e、p、E、C、A、R、V、V_d、V_o}。其中,子类别可以和父类别同时标记到一个语义要素,比如“歼 10”可标记为 <e,E>。

3.3.2　规则识别方法

尽管词库识别可以覆盖多数语义要素,但是仍有部分复杂的语义要素难以通过词库进行识别。比如,“有多远”对应属性 A“距离”,“50 km”对应数值型属性值 V_d等。这些语义要素由于结构复杂,难以通过词库识别,其识别要借助于词性、预定义文本结构、文本模式等。本文制定如下规则辅助词库识别语义要素:

1)实体 E、概念 C、关系 R 识别。实体识别和关系抽取是信息抽取领域的关键技术,目前已有较成熟的解决方案。通过综合考量,本文选取基于条件随机场(Conditional Random Field,CRF)的命名实体识别与基于规则和启发式算法的关系抽取来进行 E、C、R 识别。

2)属性值 V 识别。待识别的属性值 V 由对象型属性值 V_o和数值型属性值 V_d组成,通过词库识别的方法可以有效识别大部分 V_o,部分未覆盖的 V_o和 V_d则需要定义规则进行识别。因此,本文定义如下的启发式规则对词库识别属性值 V 进行补充:

(1)依据词性识别 V_o。LTP 词性标注会将特殊的名词词性标注出来,比如“现在”词性为“nt(时间名词)”、“城郊”词性为“nl(位置名词)”。经过分析,这类词性的词汇多数对应属性值 V_o,因此,可以在识别时依据词性将其标记为 V_o,其中,主要包括 nd(方位名词)、nl(位置名词)、ns(地理名词)3 种词性。

(2)依据模式识别 V_d。标记为数值型属性值 V_d的词汇中必然包含数值型数据,但是包含数值型数据的词汇不一定就是 V_d。问题中的 V_d一般有其特殊的模式,结合关键词模式和依存分析结构定义如下 2 种模式进行 V_d识别:

① 关键词模式:数值数据 + 计量量词集{“千米”“米”“公里”“吨”“斤”“尺”等},如“车长 5 米的坦克有哪些?”。

② 依存树结构:属性 A-*word_prep* 数值,比如“速度_A 为 100 的坦克有哪些?”。其中,*word_prep* 为介词集合{“是”“有”“为”“属于”“属”}。

3)属性 A 识别。复杂属性识别是语义要素识别中最难的部分,也是本文分析的重点。从目前的研究来看,多数论文工作者很少详细描述如何处理该部分。而本文的实验结果表明,问题理解中由复杂属性未能识别导致组合的语义表达式错误占总错误的 77.08%,由此可见,复杂属性识别非常重要。为解决该问题,本文搜集大量的百度知道问题,通过统计分析制定如下复杂属性识别规则:

(1)结合 E/C 和词性规则识别属性。多数属性都和实体 E、概念 C 有密切联系,极少存在单独的属性单元,因此,可以结合词性和已识别的 E、C 定义如下的识别模式:

(E/C 的)$word_{(n)}$ + ($len(word_{(n)})>1$)

其中,+ 表示多个条件组合,( )表示必需,/表示或者,$word_{(n)}$表示词性为名词的词汇,(n)表示该词汇的词性为名词。上述模式需要满足 2 个条件:① 表示特定的词汇结构;② 词汇结构中的 $word_{(n)}$中文字符长度至少为 2。例如“战斗机_E 的速度(n)是多少”,其中,“的速度”满足上述模式,因此,“速度”会被识别为属性 A。

(2)附加 V_o被省略的属性。有些对象型属性值 V_o所对应的属性 A 在使用时被用户省略,如“城郊_V_o的 机场_E 名字_A 是 什么”,完整的带属性的问句应当是“位置_A 在 城郊_V_o的 机场_E 名字_A 是什么”,其省略了属性“位置_A”,会导致语义组合时“城郊_V_o”不能和“机场_E”组合,因此,要在识别 V_o时附加其对应属性 A。

本文总结的部分词性-属性对应关系如表 1 所示。其中,词性范围为{nd,nl,ns}。

表 1　词性-属性对应关系

词性	对应属性 A	例子
nd	方向	右侧(nd)的坦克有多长?
nl	位置	西郊(nl)的机场叫什么?
ns	国籍	中国(ns)的飞机有哪些?
ns	位置	北京(ns)的防空部队?

对于词性不满足上述条件的属性值附加属性,比如“国产_V_o”对应属性“产地_A”,这些 V_o需要另外建立属性值-属性对应词典 V_o-A,在识别属性值时利用词典附加其属性。

(3)结合问点块识别复杂属性

问题中的问点块会暗含属性,比如“火箭弹有多长?”,其中,问点块“有多长”中暗含了属性“长度”。但是有时问点块中暗含的属性会和前面的属性重复,比如“火箭弹的长度_A 有多长?”。因此,结合问点块的复杂属性识别要在识别后进行属性重复判断,若重复则忽略识别的属性。结合问点块的复杂属性识别由 3 个部分组成:问点块确定,复杂属性识别,属性重复判断。各部分详细介绍如下:

①问点块确定。问点块就是问题中包含疑问词的语义块。为提取问题中的问点块,确定问点块覆盖范围,本文提出基于依存分析树的回溯方法,该方法主要包含 2 个步骤:在依存分析树中定位疑问词,依据疑问词集按照最长匹配的原则在依存分析树中匹配疑问词并定位;从疑问词开始,向上或向下回溯到第一个遇到的 E、C、R、A 并停止回溯,正向输出回溯过程中经过的节点,得到问点块。

②复杂属性识别。复杂属性的识别基于得到的问点块,首先从问点块中抽取其暗含的属性,然后判断问点块中是否包含动词,对抽取出的属性进行扩展,比如“歼 10 在哪里制造(v)?”((v)表示词性为动词),其中,“哪里制造”是得到的问点块,“哪里”对应属性“地址”,结合动词“制造”,扩展的属性为“制造地址”,最终返回的属性为“制造地址”。复杂属性识别步骤为:

步骤 1 依据问点块-属性对应词典,根据最长匹配规则抽取问点块中的属性。本文统计的部分问点块-属性对应词典如表 2 所示。

表 2 问点块-属性对应词典

属性 A	问点块
时间	{“有多久”“多久”“什么时间”“什么时候”“何时”…}
地址	{“哪里”“哪儿”“何处”“什么地方”“什么地点”…}
速度	{“跑多快”“多快”“飞多快”“快多少”“哪个快”…}
射程	{“打多远”“能打多远”…}
尺寸	{“有多大”“多大”…}

步骤 2 判断问点块中是否包含动词,若存在动词,则扩展上述得到的属性,扩展方法为动词在前、属性在后。如问点块“多久完成(v)”→属性“完成时间”,“何时服役(v)”→“服役时间”,“哪里制造(v)”→“制造地址”等。

③属性重复判断

通过上述复杂属性识别得到的属性 A'可能和问题中其前面的属性 A 重复,因此,要进行属性重复判断。判断的过程主要依据依存分析树,通过依存分析树判断识别到的属性 A'和其在依存分析树结构中最近连接的属性 A 是否重复,若重复,则忽略属性 A';若不重复,则返回属性 A',并将依存分析树中问点块节点部分替换为属性 A'。

3.4 语义要素组合

语义要素组合的目的是通过一定的规则组合识别到的语义要素,形成规范化的语义表达式以表示问题语义。为保证语义要素组合的精度,本文提出基于依存分析树的组合方法。利用语义要素之间的依存关系和依存分析树的结构,首先进行函数识别并替换函数单元,然后对依存分析树进行简化,去除非函数单元和语义要素的节点,最后按照树的深度,依据规则从叶子节点开始逐层向上组合(若叶子节点为属性或关系,则允许跳过部分节点后连接到实体或概念),最终得到问题的语义表达式。语义要素组合分2 类:基本组合和函数组合,其中,基本组合优先执行,函数组合靠后执行。通过 2 类组合的执行,可以准确地生成问题语义表达式,从而有效表示问题语义。

3.4.1 基本组合

基本组合是语义要素组合的基础,其主要对非函数单元的一元语义要素和二元语义要素执行简单的连接和组合,不涉及复杂的结构组合。本文制定基本组合规则如下:

1)连接。连接关系是最常用的组合规则,主要涉及一元语义要素和二元语义要素之间的连接,具体的规则如表 3 所示。其中,“—”表示双向均可的依存连接,“→”表示单向的依存连接,“/”表示或者,$A.V$ 表示等待结合到 E/C,$C[e?]$ 表示类型为 C 的 E/C,在 e、V 组合中,若之后没有 A 结合,则丢弃 V。

表 3 连接关系的组合规则

组合单元	依存树结构	组合结果
e/V、p	$V—A/A—V$	$A.V$
	$C—A/A—C$	$C[e?].A$
	$E—A/A—E$	$E.A$
	$C→R$、$R→C$	$C[e?].R$、$R.C[e?]$
	$E→R$、$R→E$	$E.R$、$R.E$
e、V	$e—V—A$	$e.A.V$
$e.p/p.e/A.V$、e/V	$R.e—e/e.R—e$	$e.R.e$
	$e.A—V/A.V—e$	$e.A.V$

2)交。该规则符号为“∩”,涉及 2 个条件,意义为 2 个条件需要同时满足。比如,“舰载机_C[e?]∩战斗机_C[e?]”表示某个“e?”其类别既要属于“舰载机”也要属于“战斗机”。组合规则“∩”的激活条件如下:

(1)e_1、e_2、e_3 在依存分析树中共同连接到某个 e,如 $e_1—e_2—e_3$,其语义表达式为 $e_1 \cap e_2 \cap e_3$。

(2)三元组连接到同一个节点。比如 $e_1—R—e_2—A—V$,其语义表达式为 $e_1.R.e_2 \cap e_2.A.V$。

(3)依存树中结构 $C_1—C_2$,其语义表达式为 $C_1[e?] \cap C_2[e?]$。

3)否。该规则主要是对后续条件或语义组合结果的否定,其通过否定关键词集激活,符号为“!”。

如“不装备_*R* 99 式步枪_*E* 的部队_*C*”,其语义表达式为“部队[*e*?].装备.! 99 式步枪”。部分否定关键词有{“不是”“非”“不包括”“不包含”}等。

4)类包含。该规则是 *C* 和 *E* 之间的组合规则,在基本组合中具有最高优先级,但是只有满足激活条件的 *C* 和 *E* 才能执行组合,组合结果为 *C*[*E*],表示类别为 *C* 的 *E*,其激活规则如下:

(1)依存树结构为 $E \rightarrow C$ 并且依存关系为“*ATT*”。

(2)在用户问题中 *E*、*C* 相邻。

(3)依存树结构为 *E*—[“有”“是”]—*C*—疑问词,其中,“[ ]”为可选。

5)其他特殊结构

部分特殊的依存树结构要进行修正或者执行特殊的语义组合。该类结构主要有以下 4 种:

(1)$E—A_1—A_2$ 修正为 $A_1—E—A_2$。

(2)$e—R_1—R_2$,其执行特殊的语义组合规则:$e.R_1.R_2$。

(3)*e*—*R*—*A*,其执行特殊的语义组合规则:*e*.*R*.*A*。

(4)*e*—*e*,并且依存关系为“*COO*”,若后续没有指定函数接收,则在组合时将其看作单个 *e*,最后拆分成 2 个表达式。

3.4.2　函数组合

函数组合是语义要素组合的高阶操作,其利用依存分析树结构,在基本组合的基础上结合预定义的函数组合规则,生成带函数体的语义表达式,以理解复杂的问题语义。比如,“速度最快的坦克是什么”,其语义表达式为“*MEntity*(坦克[*e*?].速度,MAX)”,其中,MAX 参数表示函数 *MEntity*( )取最大,即最大属性值的实体。函数组合主要包含 2 个步骤:函数识别,定位替换、语义要素组合。其难点在于函数识别,为准确识别问题中的函数,本文以问题中语义要素类型和数目、问点块类型、关键词集、依存树结构、依存关系等作为特征,提出基于模式的函数识别方法,并根据不同的函数制定其语义要素组合规则。函数组合所含元素具体介绍如下:

1)函数类别

本文函数组合包含 8 类函数,共 17 种,其详细的函数类型及功能如表 4 所示。

表 4　函数类型及功能

函数类型	子函数	函数功能
概念类	*Concept*( )	返回实体的所有属性、属性值
区别类	*Different*( )	返回输入实体之间不同属性值的属性及其属性值
原因类	*Why*( )	返回原因
比较类	*CompareBase*( )	基础比较函数,是后续 2 个函数的支撑
	CompareSelect( )	返回满足比较条件的实体
	CompareValue( )	返回实体属性值之间比较的结果
最值类	*MEntity*( )	返回满足最值条件的实体
	MValue( )	返回满足最值条件的实体的属性值
选择类	*Exist*( )	判断当前输入是否成立
	Select( )	选择多个输入中正确的输出
统计类	*List*( )	列举出满足条件的所有结果
	Count( )	统计满足条件的结果数目
计算类	*Inequality*( )	不等式计算
	Calculate( )	数值计算
	Sum( )	求和
	Average( )	平均值
	UnitConvert( )	单位转换

2)问点块类型确认

问点块暗含了用户关注的重点,比如“歼 10 和歼 15 比速度哪个快?”和“歼 10 和歼 15 比速度快多少?”,前者关注的是速度快的实体,对应函数 *CompareSelect*( ),而后者关注速度快了多少,对应函数 *CompareValue*( )。因此,要在函数识别前确认问点块类型。结合关键词集,本文定义如下问点块类型:

(1)可省略。识别条件:仅有疑问词 +[*word*_*prep*]。

(2)? *Value*。识别关键词:{“多少”“几”},“? *notValue*”表示非“? *Value*”的问点块。

(3)? *Num*。识别条件:{“多少”“几”} + {“个”“架”“辆”“发”“艘”…}。

(4)? *listnum*。识别条件:{“多少”“几”} + {“种”“类”“种类”“类型”}。

(5)其他。

3)基于模式的函数识别及语义要素组合

为取得理想的函数组合效果,本文分析并归纳大量的用户问题,结合问点块类型、依存关系等特征,为每个函数建立一套准确的能够覆盖大多数用户问题表达方式的识别模式,并运用以下符号来准确描述模式结构:“[ ]”表示可选,“( )”表示必需,“/”表示或者,“ + ”表示多个条件组合,“—”表示双向均可的依存连接,“→”“←”表示单向的依存连接,“*NumE*”表示问题中实体 *E* 的数目。以比较类中的函数为例,本文建立的函数识别模式及语义要素组合规则如下:

(1) *CompareBase*(*E*. *A*/*A*)。

① 函数功能:基础的比较类函数,输出多个 *E* 各自的属性对比,若有属性输入则只比较该属性,否则比较 *E* 所有属性。如“歼 10 和歼 15 相比怎么样”,语义表达式为 *CompareBase*(歼 10,歼 15)。

② 识别规则:*NumE* > =2 + *key_cmp* + [*C*/*A*] + 不满足 *CompareSelect*( ) 和 *CompareValue*( )。

③ 定位替换:*key_cmp* 替换为 *CompareBase*( )。

④ 组合规则:结合依存分析树结构,其组合规则如图 2 所示。其中,属性一般连接在实体之后,其会通过基本组合规则组合到实体 *E* 上,因此,在图 2 中并未标注属性 *A* 的位置,仅标注和函数相连的实体结构。

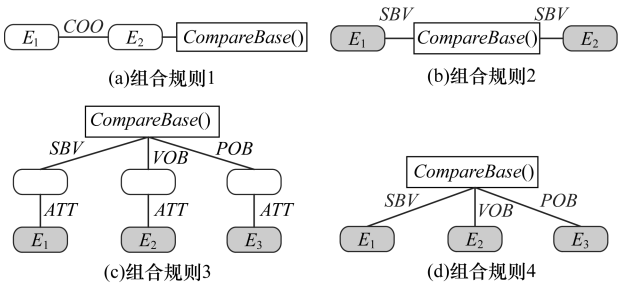

图 2 *CompareBase*( )依存分析树结构组合规则

(2) *CompareSelect*(*E*. *A*,0/1)。

①函数功能:若有属性则输出经过属性比较的 *E*,否则默认输出属性“排名”。0/1 表示输出较大的 *E* 还是较小的 *E*, *words_high*/*words_cmp* 对应 1, *words_low* 对应 0。

②识别规则:

NumE > =2 + [*key_cmp*] + [{“哪/那个”“哪/那一个”“谁”“哪/那种”}] + [*A*] + [“更”] + *words_cmp*/*words_high*/*words_low* + 有问点块。定位替换:*key_cmp*/问点块替换为 *CompareSelect*( )。

word_cmp—*word_cmp*(且依存关系为“*COO*”) + 没有问点块。定位替换:该结构替换为 *CompareSelect*( )。

③组合规则:同 *CompareBase*( )。

(3) *CompareValue*(*E*. *A*)。

①函数功能:输出实体属性值比较的结果,如“歼 10 比歼 15 速度快多少”, *CompareValue*(歼 10,速度,歼 15,速度)。

②识别规则:*NumE* > =2 + *key_cmp* + [*A*] + *words_high*/*words_low* + “多少”。

③定位替换:*key_cmp* 替换为 *CompareValue*( )。

④组合规则:同 *CompareBase*( )。

上述提到的关键词集具体如下:

key_cmp:{“比较”“对比”“相比”“比起”“比”“改进”“更差”“更优越”“更好”“更佳”“更加”“更强”“更牛”“更”}。

words_cmp:{“先进”“狂”“厉害”“出色”“好”“流行”“强悍”“强”“棒”“优越”“高级”}。

words_high:{“快”“大”“长”“高”“远”“宽”“久”“前”“重”“好”}。

words_low:{“慢”“小”“短”“矮”“近”“窄”“后”“轻”“差”“坏”}。

3.5 语义要素映射

由问题语义表达式转换的 SPARQL 查询语句并不能在知识库中有效执行,因为其中的语义要素以用户词汇描述,需要映射到底层知识库元素。本文采取基于词库的方法映射问题语义要素,通过预先构建用户词汇-知识库元素之间的映射词库,计算其相似度值,在执行时将每个语义要素 V_{qi} 映射到多个知识库元素,生成该语义要素的候选项集 $M(V_{qi})$ 和对应的相似度集 $W(V_{qi})$。

3.6 联合消歧

消歧的目的是计算语义要素和候选知识库元素之间的映射,使得语义要素和知识库元素之间的一对多关系转变为一对一,进而生成知识库语义表达式。本文采取联合消歧的方法,将所有待消歧的语义要素整合到一个大的消歧任务中。首先依据语义要素及其候选项集构建消歧图,然后计算其边权值,最后为消歧图添加约束条件,通过最大化其目标函数来生成知识库语义表达式。

3.6.1 消歧图构建

消歧图的定义及构建过程如下:

1)消歧图定义

以 G 表示消歧图,V 表示顶点集,E 表示边集,本文定义消歧图 $G = <V, E>$,其中,$V = VK \cup VQ$, $E = E_{\text{sim}} \cup E_{\text{sup}}$。各元素含义如下:

(1) VQ 是识别到的语义要素集合,$V_{qi} \in VQ$ 是以自然语言描述的语义要素节点。

(2) VK 是问题中语义要素映射的结果集,$VK = M(V_{q1}) \cup M(V_{q2}) \cup \cdots \cup M(V_{qi}) \cup \cdots \cup M(V_{qn})$, $Vk_{ij} \in VK$ 是语义要素 V_{qi} 映射后的节点。

(3) $E_{\text{sim}} \subseteq VQ \times VK$,该边的权值是语义要素节点 V_{qi} 和其映射节点 $M(V_{qi})$ 之间的相似度量。

(4)$E_{\sup} \subseteq VK \times VK$,该边的权值是不同语义要素映射节点之间(如 $M(V_{q1}) \times M(V_{q2})$)的语义一致性度量。

2)消歧图生成

消歧图的生成以问题语义表达式中基本组合的“连接”为依据。首先,连接所有的问题语义要素节点 V_{qi} 及其映射节点集 $M(V_{qi})$;其次,若 V_{qi} 和 V_{qj} 之间存在“连接”,则将 $M(V_{qi})$ 和 $M(V_{qj})$ 中的节点相互连接;最后,若存在函数,则以函数作为分隔生成不连通的子图,分别进行消歧。以“X 军装备的国产 96 坦克速度是多少?”为例,其消歧图如图 3 所示,其中,无方框节点表示 V_{qi},虚线方框表示 $M(V_{qi})$,带阴影方框/椭圆节点表示 Vk_{ij}。

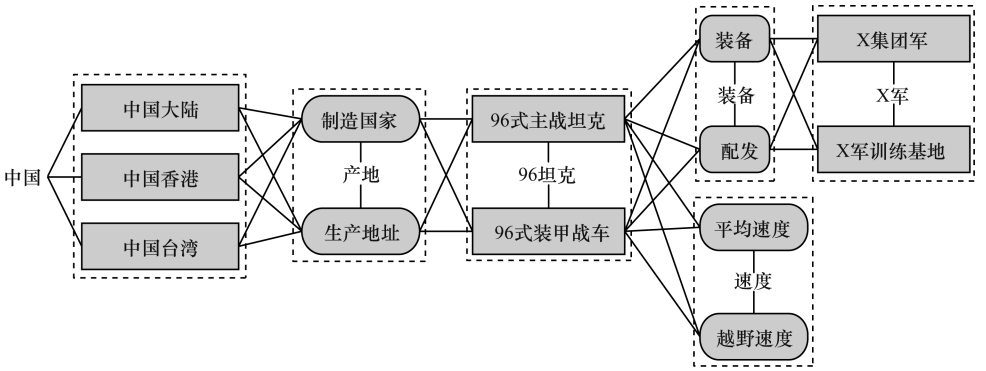

图 3　“X 军装备的国产 96 坦克速度是多少?”消歧图

3.6.2　边权值

边 $E_{\sup}$ 反映知识库对于消歧结果的支撑性,即与该边连接的节点是否符合知识库结构。以 V_{qm}、V_{qn} 表示与 $E_{\sup}$ 连接的 2 个映射集的语义要素(如“96 坦克”“装备”),$Vk_{mk} \in M(V_{qm})$、$Vk_{nl} \in M(V_{qn})$ 表示与 $E_{\sup}$ 连接的 2 个节点(如“96 式主战坦克”“配发”),$freq(Vk_{mk}, Vk_{nl})$ 为知识库中 $<Vk_{mk}, Vk_{nl}>$ 连接出现的频次,$|M(V_{qm})|$ 为 V_{qm} 映射集的数目。则在消歧图中,边 $<Vk_{mk}, Vk_{nl}>$ 的权值 v_{kl} 计算公式如下:

$$v_{kl} = \frac{freq(Vk_{mk}, Vk_{nl})}{\sum_{k,l} freq(Vk_{mk}, Vk_{nl})}$$

其中,$k \in |M(V_{qm})|$,$l \in |M(V_{qn})|$。

边 E_{sim} 反映问题中的语义要素和知识库元素之间的匹配性。以 V_{qi} 表示语义要素,根据第 3.5 节的语义要素映射,可以得到其候选项集为 $M(V_{qi})$,对应相似度集为 $W(V_{qi})$。则对于候选项集中的 Vk_{ij},其消歧图中边 $<V_{qi}, Vk_{ij}>$ 的权值为 w_{ij},且 $w_{ij} \in W(V_{qi})$。

3.6.3　消歧处理

联合消歧的结果是生成消歧图的子图,该子图跨越消歧图中所有虚线框,以图 3 的消歧图为例,其消歧结果如图 4 所示。

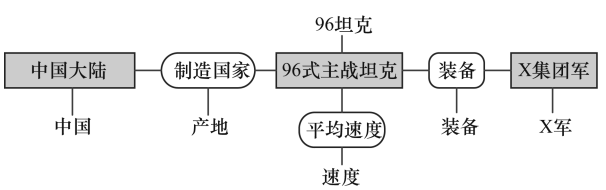

图 4　“X 军装备的国产 96 坦克速度是多少?”消歧结果

本文的消歧目标是在满足约束条件的情况下最大化目标函数值。在描述具体的消歧方法前,进行如下定义:

$GX_{ij} \in \{0,1\}$:消歧图上 VQ 中节点 i 和 VK 中节点 j 之间是否存在边。

$GY_{kl} \in \{0,1\}$:消歧图上 VK 中节点 k 和节点 $l(k \neq l)$ 之间是否存在边。

$X_{ij} \in \{0,1\}$:VQ 中节点 i 和 VK 中节点 j 之间的边是否被选中。

$Y_{kl} \in \{0,1\}$:VK 中节点 k 和节点 $l(k \neq l)$ 之间的边是否被选中。

v_{ij}:$E_{\sup}$ 的权值。

w_{ij}:E_{sim} 的权值。

本文目标函数为:

$$\text{Max } Score(Q, G, K)$$

其中,$Score(Q, G, K) = \alpha \sum_{i,j} w_{ij} X_{ij} + \beta \sum_{k,l} v_{kl} Y_{kl}$。

约束条件为:

1)对任意的一个 VQ 节点,有且仅有一个映射节点。

$$\sum_{j} X_{ij} = 1, \forall i$$

2)如果 VK-VK 之间的某个边被选中($Y_{kl} = 1$),那么必定存在 2 个 VQ 节点分别映射到 Vk_{ik} 和 Vk_{jl}($\exists i, j, X_{ik} = 1 \cap X_{jl} = 1$)。

$$Y_{kl} \leqslant \sum_{i} X_{ik} \cap Y_{kl} \leqslant \sum_{j} X_{jl}$$

3)选中的边必须在图中存在。

$$X_{ij} \leqslant GX_{ij} \cap Y_{ij} \leqslant GY_{ij}$$

3.7　SPARQL 生成

经过联合消歧后生成知识库语义表达式,对于消歧图中基本组合构成的三元组集,可以直接转换为 SPARQL 查询语句,而函数组合的部分则根据预定义的函数类型进行相应转换,生成对应的

SPARQL 查询语句。以“X 军装备的国产 96 坦克速度是多少?”为例,其 SPARQL 查询语句如下:

```
Select o? where{
X 集团军  装备  96 式主战坦克
96 式主战坦克  制造国家  中国大陆
96 式主战坦克  平均速度  o?
}
```

在知识库上执行该 SPARQL 查询语句,最终得到用户问题的答案。

4 实验验证

4.1 实验设置

由于中文没有类似于 QALD 的测试集,为测试本文的问题理解方法,借助百度知道收集问题集:首先结合知识库构建实体集、实体-关系集,在百度知道中检索,得到问题集后以该问题集为基础在百度知道中检索并收集同义问题,然后去除无关问题和重复问题。本次实验共收集问题 8 732 个,去除结构类似的问题后得到 6 215 个问题,随机选取 1 000 个问题作为测试集,5 215 个问题作为辅助分析集。

在实验实现方面,由于哈工大语言技术平台(LTP)[16]分词工具不支持附加词库辅助分词,本文采取 jieba 分词工具[17]对问题进行分词,附加语义要素识别当中的所有词库,词性标注以及依存分析则采取哈工大语言技术平台(LTP)-3.4.0,同义词库采用哈工大社会计算与信息检索研究中心同义词词林扩展版[18]。

4.2 评测标准

本文采取的评测方法为:首先人工给出每个测试问题的答案,然后使用本文方法执行问答,最后将问答系统给出的答案与人工构造的标准答案进行比较,若问答结果与人工答案一致,则判定回答正确。采取精确率 P、召回率 R 和 F1 值对实验效果进行评价,各值定义如下:

精确率 P 指问答系统正确回答的问题数和问答系统返回答案不为空的问题数的比率,其计算公式如下:

$$P=\frac{\text{问答结果正确的问题数}}{\text{问答结果不为空的问题数}}$$

召回率 R 指问答系统正确回答的问题数和总问题数的比率,其计算公式如下:

$$R=\frac{\text{问答结果正确的问题数}}{\text{总问题数}}$$

一般情况下,精确率 P 和召回率 R 相互制约,精确率提高会导致召回率下降,反之亦然。因此,为综合衡量系统性能,避免精确率和召回率的片面性,本文采取 F1 值进行综合评价,其计算公式如下:

$$\text{F1}=\frac{2\times P\times R}{P+R}$$

4.3 实验结果与分析

对 1 000 个测试问题进行实验分析,设置 3 组实验进行对比:第 1 组仅使用基于关键词的信息检索(IR)方式,不执行语义组合及后续过程,直接利用识别到的语义要素作为查询关键词执行信息检索;第 2 组为 E_{sim}-only 方式,其在本文方法的基础上,去除联合消歧中的 E_{sup},仅使用 E_{sim} 进行消歧;第 3 组运用本文方法进行实验。实验结果如表 5 所示。

表 5 不同语义要素识别方法实验结果

实验方法	精确率	召回率	F1
IR 方法	0.259	0.237	0.248
E_{sim}-only 方法	0.689	0.534	0.602
本文方法	0.848	0.731	0.841

从表 5 可以看出,本文基于语义要素组合的问答系统平均 F1 值达到 0.841,说明其能够有效理解问题语义并回答问题。

进一步分析系统中联合消歧的性能,采取本文方法将 $Score(Q,G,K)$ 排名前 k 的消歧图子图转换为 SPARQL 语句并进行问答实验,结果如图 5 所示。

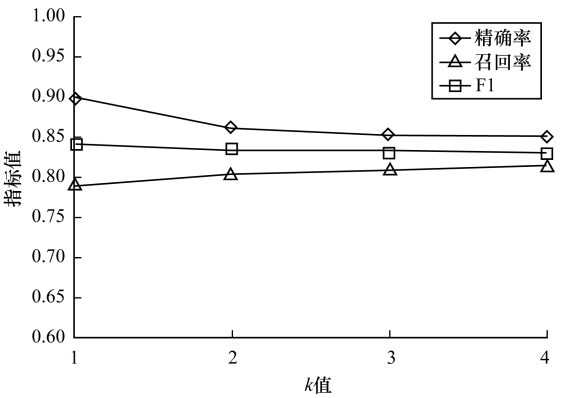

图 5 $Score(Q,G,K)$ 排名前 k 的子图问答结果

从图 5 可以看出,本文方法在 k 取 1 时 F1 值最大,为 0.841,随后,随着 k 的增大,SPARQL 查询语句增多,F1 值逐渐降低。该结果表明本文的联合消歧方法能够有效消歧并从消歧图中选择最优的子图。

4.4 错误分析

进一步进行分析,可以发现导致本文系统回答出错的原因主要有以下 4 点:

1)用户问题过于复杂。某些复杂的用户问题可能包含多个子问题和复杂的指代关系,导致语义要素识别阶段出现错误,进而使得语义表达式生成出错。

2)知识库不够丰富,映射并转换成功的 SPARQL 查询语句无法在知识库中找到结果。

3)依存分析错误导致语义组合出错。概念类、基本组合类、计算类问题不涉及复杂的依存分析结构,因此,这 3 类问题不受影响,而其他函数对依存分析结构依赖较强,受影响较大,容易出现较多错误。

4)复杂函数识别错误。如比较类函数,其识别规则复杂繁多,容易因为要求过严而导致未能识别,进而使得语义组合出错。

5　结束语

本文提出基于语义要素组合的问答方法,该方法无需大量的人工标注数据和正确的问题-答案对进行训练,而是采取语义要素分析、语义要素组合来直接分析问题语义,然后产生规范化的问题语义表达式,再通过映射和联合消歧生成知识库语义表达式,最后将知识库语义表达式转换为 SPARQL 语句执行查询获取知识。该方法能较好地进行问答系统中问题语义分析,有效理解和推理复杂问题,降低知识获取的门槛。实验结果表明,该方法的平均 F1 值为0.841。虽然本文方法可以理解绝大多数的问题,但是仍对部分复杂问题理解出错,通过错误分析发现,该类问题多数和函数模式以及指代关系有关,因此,今后将进一步优化函数模式并添加指代消解模块,以提高本文方法理解复杂问题的准确度。

参考文献

[1] SUCHANEK F M, KASNECI G, WEIKUM G. Yago: a core of semantic knowledge[C]//Proceedings of International Conference on World Wide Web. New York, USA: ACM Press, 2007: 697-706.

[2] AUER S, BIZER C, KOBILAROV G, et al. DBpedia: a nucleus for a Web of open data[C]//Proceedings of ISWC'07. Berlin, Germany: Springer, 2007: 722-735.

[3] VRANDEĈIC D, KRÖTZSCH M. Wikidata: a free collaborative knowledge base[J]. Communications of the ACM, 2014, 57(10): 78-85.

[4] PRUD'HOMMEAUX E, SEABORNE A. SPARQL query language for RDF[EB/OL]. [2017-09-05]. https://www.researchgate.net/publication/225070173_SPARQL_Query_Language_for_RDF.

[5] WANG D S. A domain-specific question answering system based on ontology and question templates[C]// Proceedings of the 11th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing. Washington D. C., USA: IEEE Computer Society, 2010: 151-156.

[6] ZHENG W, ZOU L, LIAN X, et al. How to build templates for RDF question/answering: an uncertain graph similarity join approach[C]//Proceedings of 2015 ACM SIGMOD International Conference on Management of Data. New York, USA: ACM Press, 2015: 1809-1824.

[7] 马　莉,唐素勤,陈立娜,等. 改进的基于句模匹配算法的问句理解方法[J]. 计算机工程, 2009, 35(20): 50-52.

[8] YAO X, DURME B V. Information extraction over structured data: question answering with freebase[EB/OL]. [2017-09-05]. http://www.cs.jhu.edu/~xuchen/paper/acl14-ie-freebase.pdf.

[9] BORDES A, WESTON J, USUNIER N. Open question answering with weakly supervised embedding models[C]// Proceedings of European Conference on Machine Learning and Knowledge Discovery in Databases. Berlin, Germany: Springer, 2014: 165-180.

[10] BORDES A, CHOPRA S, WESTON J. Question answering with subgraph embeddings[EB/OL]. [2017-09-05]. http://www.thespermwhale.com/jaseweston/papers/fbqa.pdf.

[11] ZOU L, HUANG R, WANG H, et al. Natural language question answering over RDF: a graph data driven approach[C]//Proceedings of ACM SIGMOD International Conference on Management of Data. New York, USA: ACM Press, 2014: 313-324.

[12] UNGER C, CIMIANO P. Pythia: compositional meaning construction for ontology-based question answering on the semantic Web[C]//Proceedings of International Conference on Natural Language Processing and Information Systems. Berlin, Germany: Springer, 2011: 153-160.

[13] UNGER C, LEHMANN J, NGOMO A C N, et al. Template-based question answering over RDF data[C]// Proceedings of International Conference on World Wide Web. New York, USA: ACM Press, 2012: 639-648.

[14] CAI Q, YATES A. Large-scale semantic parsing via schema matching and lexicon extension[C]// Proceedings of Meeting of the Association for Computational Linguistics. Berlin, Germany: Springer, 2013: 423-433.

[15] BERANT J, CHOU A, FROSTIG R, et al. Semantic parsing on freebase from question-answer pairs[EB/OL]. [2017-09-05]. https://nlp.stanford.edu/joberant/homepage_files/publications/semparse EMNLP13.pdf.

[16] CHE W, LI Z, LIU T. LTP: a Chinese language technology platform[C]//Proceedings of International Conference on Computational Linguistics: Demonstrations. [S.l.]: Association for Computational Linguistics, 2010: 13-16.

[17] Chinese words segementation utilities[EB/OL]. [2017-09-05]. https://pypi.python.org/pypi/jieba/.

[18] 同义词词林(扩展版)[EB/OL]. [2017-09-05]. http://www.ltp-cloud.com/download/.

编辑　吴云芳