

学校代码： 10246  
学 号： 13110240001

復旦大學

博 士 学 位 论 文  
(学术学位)

基于知识图谱的问答系统关键技术研究

**Research of Key Technologies for Question Answering over  
Knowledge Graphs**

院 系： 计算机科学技术学院

专 业： 计算机软件与理论

姓 名： 崔万云

指 导 教 师： 汪卫 教授

完 成 日 期： 2017 年 4 月 1 日



## 指导小组成员名单

肖仰华 副教授 复旦大学

汪 卫 教 授 复旦大学

张 亮 教 授 复旦大学

顾 宁 教 授 复旦大学



# 目录

<b>第一章 绪论</b>	<b>7</b>
第 1 节 问答系统背景介绍	7
1.1. 知识图谱简介	8
1.2. 知识图谱在问答系统上的数据优势	9
1.3. 基于知识图谱的问答系统工作方式	10
第 2 节 研究架构与模块关联	11
2.1. 研究架构	11
2.2. 研究系统性	12
第 3 节 本文组织结构	13
<b>第二章 相关工作</b>	<b>15</b>
第 1 节 问答系统分类	15
第 2 节 基于信息检索的问答系统及其不足	16
2.1. 用户问题处理	16
2.2. 生成搜索关键词	17
2.3. 文章检索	17
2.4. 段落抽取	17
2.5. 答案抽取	18
2.6. 不足之处及使用知识图谱的动机	18
第 3 节 基于知识图谱的问答系统及其不足	19
3.1. 问题分析	19
3.2. 以往研究	19
<b>第三章 基于局部搜索的语义社团挖掘</b>	<b>23</b>
第 1 节 引言	23
第 2 节 问题定义	25
2.1. 问题定义	26
2.2. CSM和CST关系分析	27
第 3 节 mCST的NP完全性	28
第 4 节 全局搜索	29

4.1.	k核和最大核	29
4.2.	CST和CSM的解决方案	29
第 5 节	CST问题的局部搜索解法	30
5.1.	基准算法	30
5.2.	一个CST问题的解决框架	31
5.3.	优化算法	36
第 6 节	CSM的局部搜索方法	38
6.1.	扩展搜索空间	38
6.2.	候选结点生成	40
第 7 节	实验	41
7.1.	数据集	41
7.2.	案例分析	42
7.3.	CST的结果	42
7.4.	CSM的结果	45
第 8 节	小结	46
<b>第四章</b>	<b>基于知识图谱的短文本动词理解</b>	<b>47</b>
第 1 节	引言	47
第 2 节	相关工作	49
2.1.	一般性和特殊性的取舍问题	49
第 3 节	问题模型	51
3.1.	初步定义	51
3.2.	模型	52
3.3.	算法	54
第 4 节	实验	55
4.1.	设置	55
4.2.	动词模板的统计信息	56
4.3.	有效性	56
第 5 节	应用：基于上下文的实体概念化	57
第 6 节	小结	59
<b>第五章</b>	<b>从问答语料库和知识图谱学习问答</b>	<b>61</b>
第 1 节	绪论	61
1.1.	方法概览	62
第 2 节	系统概览	63
第 3 节	本文的方法：KBQA	65

3.1.	问题模型	66
3.2.	概率计算	67
3.3.	在线过程	68
第4节	属性推断	68
4.1.	似然度	68
4.2.	参数估计	70
4.3.	实现	72
第5节	复杂问题回答	72
5.1.	问题陈述	72
5.2.	度量标准	74
5.3.	算法	75
第6节	属性扩展	77
6.1.	对扩展属性的KBQA	77
6.2.	扩展属性的生成	77
6.3.	$k$ 的选择	78
第7节	实验	79
7.1.	实验设置	79
7.2.	概率模型的合理性	80
7.3.	有效性	81
7.4.	效率	84
7.5.	KBQA的具体模块评估	85
第8节	相关工作	87
第9节	小结	88
第六章	针对序列-序列的自然语言处理任务的迁移学习框架	89
第1节	引言	89
第2节	相关工作及其不足	91
第3节	框架	92
第4节	TransferLSTM: 一个具体实现	93
4.1.	开放领域的神经网络	93
4.2.	知识迁移的特定领域模型	94
第5节	实验	95
5.1.	设置	95
5.2.	词性标注	95
5.3.	词分块	97
5.4.	情感分析	97

第6节 小结	100
<b>第七章 领域知识挖掘</b>	<b>101</b>
第1节 概述	101
第2节 相关工作及其不足	103
第3节 系统概览	104
第4节 种子DKS标注	105
第5节 DKS分类器	106
第6节 实验	107
6.1. 实验设置	107
6.2. 中国移动客户服务	109
6.3. 百度百科	110
6.4. 特征贡献	111
6.5. 应用：领域信息抽取	111
第7节 小结	113
<b>第八章 基于Freebase的KBQA问答系统展示</b>	<b>115</b>
第1节 架构	115
1.1. 线上过程	115
1.2. 线下过程	116
第2节 展示	117
2.1. 多种问题类别	117
2.2. 可解释性	118
2.3. 用户反馈	119
<b>第九章 总结和展望</b>	<b>121</b>
第1节 研究总结	121
第2节 研究展望	122
2.1. 常识引入	122
2.2. 领域适配	122
2.3. 文本描述+知识图谱	123



## 摘要

自然语言问答正逐渐成为一种人与机器进行交互的新趋势。从交互形式上，自然语言更接近人的交流习惯；从信息量上，自然语言的语义蕴含更为丰富。最近一段时间，自然语言问答系统获得了人工智能及其相关产业的广泛关注，并已经在互联网、医疗、金融等领域进行了应用尝试。

问答系统依赖一个优质的知识来源。知识来源基于其形式的不同，一般可以分为两类。一类是常见的纯文本形式，如网络文档、问答社区问答对、搜索引擎结果、百科描述文本等。另一类则是知识图谱，通常以RDF三元组的形式结构化表示。由于其结构化、关联化特征，知识图谱相比纯文本语料，具有更丰富的语义表达、更精确的数据内容、更高效的检索方式。这些特征保证了基于知识图谱问答的有效性。另一方面，近些年涌现出了大批十亿甚至更大规模的知识图谱，包括WolframAlpha, Google Knowledge Graph, Freebase等。这些知识图谱为问答系统提供了高覆盖率的开放领域的知识来源。因此基于知识图谱的问答系统正变得可行。

基于知识图谱的问答系统有两个核心问题：问题的理解和表示，语义关联。前者从问题出发，对问题做解析，并将问题转化为机器可以理解的表示形式。后者面向问题表示形式与知识图谱的对接，主要包括将问题的计算机表示，转化为知识图谱中的结构化查询。

除了面向开放领域的问答系统，基于知识图谱的问答系统的应用关键在于领域适配。即将开放领域的问答系统对某一具体领域作自动化适配，使计算机可以对该领域进行深度理解，回答该领域的问题。具体来讲，包括自然语言处理模型在特定领域的适配，以及领域知识的自动抽取。

本文系统性的就以上核心问题进行了研究。其系统性体现在：（1）不同层级的语义理解，包括实体层、短文本层、问题层等；（2）不同领域的问答系统构建，包括开放领域，以及特定领域自动化迁移。本文不止单独研究每一语义层级、每一领域构建，同时注重研究其相互关联，使得低语义层级的研究结果，可以被高语义层级的算法使用；开放领域的问答，可以适配到特定领域的模型积累中。具体而言，本文开展了以下研究工作并作出了相应的贡献：

1. **实体语义理解** 本文研究了面向实体的语义社团搜索模型，特别是从模型的表达性、有效性等方面研究不同社团定义下的社团搜索的合理性。其次，语义社团搜索作为语义理解的原子操作，需要保证在大图上的实时回答。为此，本文研究了大图上的高效社团搜索方法，特别是基于大图的局部性的高效搜索方法。

2. **短文本语义理解** 本文提出了动词模板，用来理解句子中的动词的语义。不同于传统动词理解方式，动词模板是对动词语义更细粒度表示。同时本文利用动词模板，提升了上下文相关的实体概念化。
3. **问题语义理解和表示** 本文提出了问题模板作为问题表示形式，即将问题中的实体作概念化，用来表示问题的语义。该表示方式解决了自然语言描述的多样性问题，完整而一致的表示了问题的语义。
4. **语义关联** 本文提出了基于问答语料（Yahoo! Answers）的语义关联学习，即将问题模板映射到知识图谱的结构化查询中。出于对语义的不确定性的考虑，本文使用了概率图模型来表示这一过程，并利用最大似然估计和EM算法来实现对于语义关联度的参数预测。
5. **领域问答适配** 本文提出了一种基于深度神经网络的迁移学习算法，用于自然语言处理模型从开放领域到具体领域的自动化迁移，以保证相关模型可以在具体领域中使用。同时，本文提出了一种自动化领域知识挖掘方法，利用领域问答语料，实现自然语言的自动领域知识抽取，从而为领域问答提供了知识来源。

**关键字:** 问答系统，知识图谱，实体理解，动词理解，领域适配

**中图分类号:** TP391.1

## Abstract

Natural language question answering (QA) has become a popular way for human computer interaction. Natural language is a more familiar way for human to communicate. And natural language contains rich semantics to represent humans' needs. Recently, question answering has drawn a lot of attention from the artificial intelligence and its related industries. It has been primitively applied to a variety of domains, including internet, medical care, and finance.

The QA system relies on a good source of knowledge. Based on its form, the source of knowledge can be divided into two categories. One is the common form of plain text, such as web documents, community QA, search engine results, and encyclopedia text. The other is the knowledge graph, which is usually in the form of structured RDF triples. Because of the structural and linked data, the knowledge graph contains richer semantics, with more accurate values, and can be indexed more efficiently. These features guarantee the effectiveness of the QA based on knowledge graphs. On the other hand, in recent years, many billion scale knowledge graphs have emerged, including WolframAlpha, Google Knowledge Graph, and Freebase. The size of the knowledge graph guarantees the recall of systems. Under the guarantee of accuracy and recall rate, the question answering systems based on knowledge graphs is now feasible.

A QA system has two core issues: question understanding and representation, semantic matching. The former issue is for questions, which analyses the questions and translate them into forms that a machine can understand. The latter is for the docking of question representations and knowledge graphs. It mainly translate the question representations by computers, into the structured query over knowledge graphs.

In addition to the open domain question answering, the applications of QA systems based on the knowledge graph rely on domain adaptation. The automatical adaptation for a specific domain makes computers understand the domain in depth and answer questions in this domain. Different domains have different focuses. Specifically, it includes the adaptation of natural language processing models in specific domains, as well as the automatic extraction of domain knowledge.

This article systematically studied the above core issues. The systematicness reflects on: (1) Different levels semantic understanding, including entities, short texts, and questions. (2)

Different domains' QA systems, including open domain and automatic adaptation for specific domains. We can not only study each semantic level, each domain level, but also focus on the study of their correlations. The lower semantic level's results can be used in the higher semantic level. Open domain QA can be adapted to a specific domain's model accumulation. Specifically, this paper has carried out the following works and made corresponding contributions:

1. **Entity semantic understanding** We study the entity-oriented semantic community search model, especially from the aspects of the expression and effectiveness of the models. We study the rationales of different definitions of community search. Second, the semantic community search is an atomic operation of semantic comprehension, which needs real time response on large networks. To this end, we studied the efficient community search method on the larger semantic network, especially the local search strategy with high efficiency.
2. **Short test semantic understanding** We proposed a model of verb templates to understand the semantics of verbs in sentences. Different from the traditional verb understandings, verb template is a more fine-grained representation of verbs' semantics. We also use the verb templates to improve the effectiveness of context-aware entity conceptualization.
3. **Question semantic understanding** We proposed the question templates, in which the entity in the question is conceptualized, to represent the semantics of the question. This representation solves the diversity of natural language, and expresses the semantics of the problem in a complete and consistent manner.
4. **Semantic matching** We learn the semantic matching based on Yahoo! Answers, by matching the question templates to the structured queries of the knowledge graph. For the sake of semantic uncertainty, we use the probabilistic graph model to represent this process, and use the maximum likelihood estimation and EM algorithm to achieve the estimation of semantic relevance.
5. **domain QA adaptation** To ensure that the models in the QA system can be still used in specific domains, we propose a transfer learning framework based on the deep neural network to automatically transfer the natural language processing models from the open domain to the specific domain. To provide the knowledge source for the domain QA, we proposed a knowledge mining method, which leverage domain QA corpus, to automatically extract knowledge from specific domains' plain text.

**Keywords:** Question Answering, Knowledge Graph, Entity Understanding, Verb Understanding, Domain Adaptation

**CLC number:** TP391.1



# 第一章 绪论

## 第1节 问答系统背景介绍

2011年10月14日，苹果公司在其iPhone 4S发布会上隆重推出新一代智能个人助理Siri。Siri通过自然语言的交互形式实现问答、结果推荐、手机操作等功能，并集成进iOS 5及之后版本。2012年7月9日，谷歌发布智能个人助理Google Now，通过自然语言交互的方式提供页面搜索、自动指令等功能。2014年4月2日，微软发布同类产品Cortana，2014年10月，亚马逊发布同类产品Alexa。在此之前的2011年9月，由IBM研发的Watson机器人参加智力问答节目“Jeopardy!”，并战胜该节目的前冠军Brad Rutter和Ken Jennings，豪取一百万美金大奖。

问答系统（Question Answering system, QA system）是用来回答人提出的自然语言问题的系统。问答系统的实现涉及到自然语言处理、信息检索、数据挖掘等交叉性领域。问答系统的历史最早可以追溯到1960年代的BASEBALL [40]和1970年代的LUNAR [101]。自那时起，有大量的问答系统涌现 [107, 22]。

智能时代，人类期望有更简单自然的方式与机器进行交互。因此以自然语言为交互方式的智能机器人广受青睐，受到各大IT厂家追捧。而其底层核心技术之一，即为自然语言问答系统。问答系统提供了自然语言形式的人与产品交互，降低了产品使用门槛，大幅提成用户体验。同时，问答系统可以帮助企业极大节省呼叫中心的投入。这些应用已经印证了问答系统的商业价值和社会价值。

问答系统的应用仍然具有新的潜力。人对于互联网的核心诉求之一是知识获取。从更长的时间窗口看，问答系统及聊天机器人，有着成为互联网知识获取新入口的优势。搜索引擎依然是现阶段最重要的互联网入口，也缔造了谷歌、百度等巨头企业。然而，基于关键字的搜索方式，缺乏语义理解，存在着与人的自然需求表达的隔阂，同时其返回结果需要人消耗大量时间剔除无意义的信息。随着人工智能、自然语言理解技术的进步，当问答系统足够智能，使人类的监督最小的时候，人就可以用问答从互联网完成知识获取。

问答系统的研究，是语义计算和自然语言处理的综合性应用。它包含了多种典型自然语言处理的基本模型，例如实体识别、短文本理解、语义匹配等。传统的单一模型研究往往仅关注某一具体问题的效果，而忽视在系统整体中的实用性。问答系统由于其复杂性，需要不同模型间的联通，才能带来综合性、实用性的技术突破。因此问答系统的研究为不同语义理解模型的整合提供了应用出口，为不同模型的关联分析、

数据共享、参数共享等提出了实际需求，为多个自然语言语义理解技术模型的整体突破带来了技术愿景。

另一方面，问答系统研究的核心在于问题语义和知识语义的理解和相似度计算。这是计算机理解人类语言和知识表达的关联，跨越语义鸿沟的关键。这条横亘在计算机面前的语义鸿沟，其关键是计算机和人类在语义表达方式上的不同。人类倾向于使用多样化、非结构化的表达来描述问题和知识，而计算机则偏爱唯一化、结构化的知识。问答系统的研究，直接作用于缩短和跨越这一语义鸿沟，将多样而模糊的问题语义，映射到具体而唯一的计算机知识库中。

优秀的问答系统有两个关键点：精确的问题理解和高质量的知识来源。近年来随着大数据的发展，这两点纷纷迎来了数据层面的发展契机。

- **问题理解** 由于问题的多样性和复杂性，很难人工制定一套规则完成问题理解。因此从数据中进行问题语义学习是必要的。社交类问答网站的兴起，包括Yahoo! Answers, Stack Overflow, 百度知道等。由用户在上面进行提问和回答。这些网站包含了大量的问答对数据集，这成为了问题理解的优质语料。海量的问答语料为问题理解的学习提供了数据基础。
- **知识来源** 由于知识表述的多样性，以及知识关联的复杂性，需要优质而大量的知识来源。近年来，一批高准确率、海量规模的知识图谱涌现，为问答系统提供了结构化、关联化的知识来源。这也为高效的问题回答提供了知识基础。

在数据发展的契机下，如何设定恰当模型学习并使用这一批数据就显得尤为重要。传统的基于规则的模型[72]无法合理利用海量语料；基于关键词的模型[98]则没有进行深入的语义理解。而一些复杂的图模型等[116, 112]，则由于时间复杂度很难直接应用在如此大规模的语料中。本文的研究，即旨在寻求一种优秀的、系统性的问答系统表示和学习模型，并进行成功应用。

### 1.1. 知识图谱简介

2012年5月份，Google花重金收购Metaweb公司，并向外界正式发布其知识图谱(knowledge graph)。自此，知识图谱正式走入公众视野。开放领域大规模知识图谱纷纷出现，包括 NELL [15], Freebase [10], Dbpedia [6], Probase [103]等。

知识图谱本质上是一种语义网络。其结点代表实体(entity)或者概念(concept)，边代表实体/概念之间的各种语义关系。知识图谱的出现是信息技术发展、时代发展的必然结果。语义的本质是关联。只有基于语义的数据互联才能发挥数据集成的非线性效应，才能获取大数据的特有语义。在这一背景下，数据互联(Linked Data)成为了一种运动，在全世界范围内方兴未艾。而数据互联的出现从深层次上来说是由时代精神所决定的。2011年的Science曾经以“互联”为题，出版专刊阐述了一个基本观点：我



们身处在一个“互联”的时代。各种网络，诸如互联网、物联网、社会网络、语义网络、生物网络等等，将各类实体、概念加以互联。网络已经成为刻画复杂性的基本形态。管理、理解和使用各种网络数据，包括知识图谱，已经成为征服复杂性的基本手段。

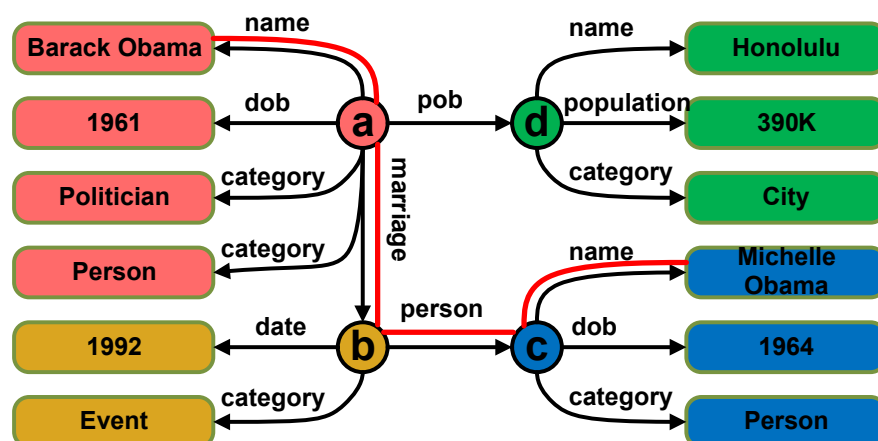


图 1.1: 一个RDF知识图谱示例。这里的“dob”和“pob”分别表示“出生日期”和“出生地”。注意到“spouse”关系是由多条边表示的：name - marriage - person - name

大部分这样的知识图谱采用了RDF作为数据格式，它们包含数以百万记甚至亿记的SPO三元组（*Subject*, *Predicate*, *Object*分别表示主语，属性，宾语）。图1.1是一个奥巴马及其相关实体构成的知识图谱的示例。可以看到，知识图谱具有明显格式化特征，其值往往是一个实体名字或者一个数字、一个日期。这保证了基于知识图谱的问答系统的回答简洁性。另一方面，不同于基于信息检索的问答系统需要考虑数据真实性的问题，知识图谱的高数据质量保证了答案的准确性。

## 1.2. 知识图谱在问答系统上的数据优势

问答系统有多种可能的数据来源。传统的数据来源包括网页文档、搜索引擎、百科描述、问答社区等。无一例外，这些数据来源都是非结构化的纯文本数据。有大量基于信息检索的方法致力于研究从纯文本数据中进行知识抽取和回答。而近年来，基于知识图谱的问答系统则成为学术界和工业界的研究和应用热点方向。相较于纯文本，知识图谱在问答系统中具有以下优势。这些优势都促使本文使用知识图谱来作为问答系统的知识来源。

- **数据关联度-语义理解智能化程度** 问题语义理解程度是问答系统的核心指标。对于纯文本数据，语义理解往往建立在问句与文本句子的相似度计算。然而语义理解和知识的本质在于关联，这种一对一的相似度计算忽视了数据关联。在知识图谱中，所有知识点被具有语义信息的边所关联。从问句到知识图谱的知识点的匹

配关联过程中，可以用到大量其关联结点的关联信息。这种关联信息无疑更为智能化的语义理解提供了条件。

- **数据精度-回答准确率** 知识图谱的知识来自专业人士标注，或者专业数据库的格式化抓取，这保证了数据的高准确率。而纯文本中，由于同类知识容易在文本中多次提及，会导致数据不一致的现象，降低了其准确率。
- **数据结构化-检索效率** 知识图谱的结构化组织形式，为计算机的快速知识检索提供了格式支持。计算机可以利用结构化语言如SQL、SPARQL等进行精确知识定位。而对于纯文本的知识定位，则往往包含了倒排表等数据结构，需要用到多个关键词的倒排表的综合排名，效率较低。

### 1.3. 基于知识图谱的问答系统工作方式

通过知识图谱为知识源回答问题时，一个问题对应于知识图谱的一个子结构。所以其问答过程的核心在于将自然语言问题映射为知识图谱上的结构化查询。例如对于图1.1中的知识图谱，表 1.1展示了一些它可以回答的问题，以及对应的子结构。

自然语言问题	知识图谱属性
① How many people are there in Honolulu?	population
② What is the population of Honolulu?	population
③ What is the total number of people in Honolulu?	population
④ When was Barack Obama born?	dob
⑤ Who is the wife of Barack Obama?	marriage→person→name
⑥ When was Barack Obama's wife born?	marriage→person→name dob

表 1.1: 自然语言问题及其在知识图谱中的属性对应。

基于知识图谱的问答系统，需要解决两个核心问题：（1）如何理解问题语义，并用计算机可以接受的形式进行表示（问题的理解和表示）；（2）以及如何将该问题表示关联到知识图谱的结构化查询中（语义关联）。

- **问题理解和表示：** 知识图谱中有数以千计的关系，而一种关系可以有数以千计的问法。例如，表 1.1中的问题①和问题②都询问了檀香山市的人口，尽管它们的表达方式大相径庭。对于不同的问题形式，问答系统使用不同的表示方法。这些问题表示必须满足（1）归一具有相同语义的问题；（2）区分不同意图的问题。

所使用的问答语料库最终找到了2782种问题意图的约2700万种问题形式。所以问题表示形式设计就是一个巨大的挑战。

- **语义关联：**在获取一个问题的表现示之后，系统需要将这一表现示映射为结构化查询。结构化查询主要依赖于知识库中的属性。由于属性和表现模型之间的跨越，寻找这样的匹配并非直接。例如，在表1.1中，系统需要知道问题①与属性population有着相同的语义。此外，在RDF图中，许多二元关系并不仅仅对应一条边，而是某种复杂的结构：在图1.1中，“配偶”关系由 $marriage \rightarrow person \rightarrow name$ 的路径表示。对于本文使用的知识库，超过98%的关系对应于类似的复杂结构。

第 2 节 研究架构与模块关联

2.1. 研究架构

本文研究架构见图1.2。依据其理解语义的粒度从小到大，从具体技术到上层应用，主要分为以下五个部分。

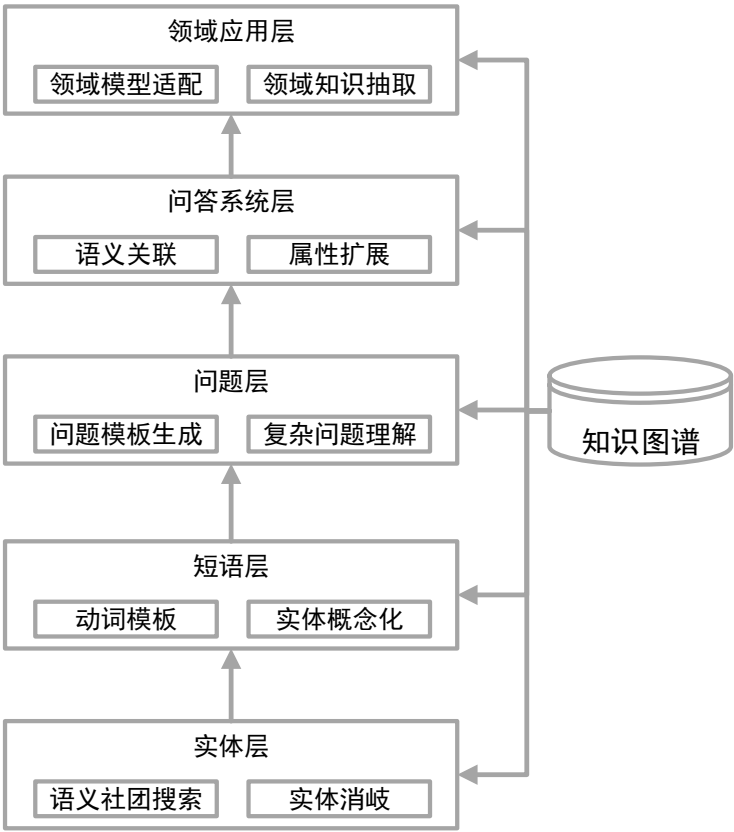


图 1.2: 研究架构图

1. **实体层：语义社团搜索** 实体是自然语言的基本单位之一，基于知识图谱的实体语义理解为上层语义计算，特别是问题中的实体语义，提供基本支持。本文研究了面向实体的语义社团搜索模型，特别是从模型的表达性、有效性等方面研究不同社团定义的合理性。其次，语义社团搜索作为语义理解的原子操作，需要保证在大图上的实时回答。为此，本文研究了大图上的高效社团搜索方法，特别是基于大图的局部性的高效搜索方法。
2. **短文本层：动词理解和实体概念化** 短文本是自然语言的最常见形式之一，起到对实体和更复杂文本单元（如问句）的承接作用。短文本上已经有了句法结构特征和上下文关系。知识图谱的作用，则可以使系统更精确表示句法结构和上下文信息。本文提出了动词模板，用来表示细粒度的动词语义。并且基于动词模板，优化了基于上下文的实体概念化方法。
3. **问题层：问题语义理解和表示** 问题的表示学习，是计算机理解问题的核心步骤。本文提出了问题的模板表示方法，将问题的实体映射到其对应概念，用以表示问题语义。此外，本文还利用问题模板的思路，解决了复杂问句的解析问题。
4. **问答系统层：问题与知识图谱的语义关联** 基于问题的模板表示，需要将其映射为知识图谱的结构化查询，最终实现问答。这一映射主要取决于知识库中的属性。本文提出了基于真实问答语料的学习方法，利用概率图建模问题模板到答案的生成，并进行语义关联的参数估计。同时，本文利用属性扩展，研究知识图谱中的多步属性这一复杂知识表示形式，学习问题模板到复杂知识图谱结构的映射。
5. **应用层：面向应用的领域问答适配** 由于通用人工智能依然是非常困难的，出于具体应用场景需要，大部分真实应用中的问答系统是面向具体领域的。例如IBM的Watson专注于医疗和财经领域，亚马逊的Alexa专注于智能家庭和零售领域，微软的Cortana专注于操作系统助手。为了对问答系统作领域适配，本文有两方面的工作，第一，将开放领域的自然语言模型（例如POS tagging），适配到具体领域，使得问答系统相关自然语言处理模块在特定领域正常运行。第二，从自然语言文本中，为特定领域进行自动化领域相关的知识抽取。

## 2.2. 研究系统性

整个研究强调整个研究的系统性，即不同层级之间的关联。一般来说，低层的结果被使用作为高层模型输入的一部分。其具体系统性关联如下：

1. **从实体到短文本** 实体层为短文本中的实体提供了语义社团。利用该语义社团，短文本层可以得到实体的相关概念，以进行实体语义消歧。从而帮助短文本层得到更精确的实体概念信息和语义信息。

2. **从短文本到问题** 短文本层为问题层提供动词语义以及实体概念信息。这为问题层的问题模板表示学习过程中的实体概念化提供了支持。
3. **从问题到问答系统** 问题层的输出，即问题模板，是整个问答系统语义关联的基础。问答系统以问题模板为输入，学习问题模板到知识图谱结构的映射。
4. **从问答系统到领域应用** 本文会将利用迁移学习将系统的自然语言处理模块应用到领域问答系统中。同时将应用层抽取的领域知识作为问答系统的知识源，以使问答系统可以适应领域问答。

### 第3节 本文组织结构

本文按照架构（图1.2）中自底向上的顺序具体介绍每一项研究内容。第三章介绍实体层的语义社团搜索。第四章介绍短文本层的动词模板，并在应用中讲述如何利用动词模板提升基于上下文的实体概念化效果。第五章提出了问答系统KBQA，包括对问题的模板化理解，以及问题模板到知识图谱的映射学习。在第六章和第七章，本文分别介绍了两个用于问答系统领域适配的技术：自言语言处理模型的领域迁移，和领域知识抽取。第八章给出了一个基于Freebase的具体系统展示。最后的第九章进行了总结和展望。



## 第二章 相关工作

问答系统已经有了大量的相关研究。本章对相关工作进行系统性阐述，并与本文提出的系统进行对比，从而比较和展示本文提出的系统的研究意义及优势。第1节介绍问答系统的不同分类。第2节介绍基于信息检索的问答系统及其不足。第3节介绍基于知识图谱的问答系统，并对比其不足。

### 第1节 问答系统分类

基于问答系统的问题类型和工作方式，可以将其进行多种分类。首先，基于其问题类别，几种典型的分类如下：

- 事实型问题：WH问题，例如when, who, where等。
- 是非型问题：Is Beijing the capital of China?
- 对比型问题：Which city is larger, Shanghai or Beijing?
- 原因/结果型问题：how, why, what等。
- 观点型问题：What is Chinese opinion about Donald Trump?

除了观点型问题，其它问题都询问某一客观事实，答案唯一。这类客观问题以事实型问题为核心。对于是非型问题Is Beijing the capital of China?，事实型问题What is the capital of China?是其回答的基础。对于对比型问题Which city is larger, Shanghai or Beijing?，事实型问题How large is Shanghai?和How large is Beijing?是其回答的基础。故本文主要研究事实型问题。事实型问题也是在问答系统研究中得到最多关注的问题类型。

问答系统的语料数据来源决定了其工作方式。典型的语料数据主要有纯文本语料和知识图谱。纯文本语料一般包括互联网网页、问答社区、维基百科页面等。基于纯文本语料的问答系统一般使用信息检索的技术，一般被称为IR-based QA (Information Retrieval-based Question Answering)。而基于知识图谱的问答系统一般被称为KB-based QA (Knowledge Base-based Question Answering)。常用的知识图谱包括领域知识库（例如医学领域[39]），和通用领域知识图谱（例如Freebase, DBpedia）。

## 第2节 基于信息检索的问答系统及其不足

基于信息检索的问答系统在收到用户的问题之后系统通过一系列流程来最终生成精确的答案。具体有以下几步:

- **用户问题处理** 用户输入自然语言问题,系统对于问题进行处理和分析,并对问题进行分类,确定问题类型。
- **生成搜索关键词** 问题中的一些词不适合作为搜索关键词,另一些词的搜索权重则较高。系统需要对于用户的问题进行分析,来获得不同关键词的权重。
- **文章检索** 系统使用从用户的问题中得到的关键词,对于数据库中的文档与关键词的计算匹配程度,从而获取若干个可能包含答案的候选文章,并且根据它们的相似度进行排序。
- **段落提取** 段落(paragraph)是包含答案的一个小节。问答系统与搜索引擎的区别在于用户期望其返回精确的答案,而不是一个文章或段落。为此首先要从文章中提取出可能包含答案的段落。
- **答案提取** 在答案可能出现的段落被提取到以后,问答系统需要精确抽取段落中所包含的答案。这一步会用到问题分类。同时根据问题的关键词,对于段落中的词进行语义分析,最终找到最有可能是答案的字段。

### 2.1. 用户问题处理

在此阶段系统对于用户的问题进行预处理。主要有以下任务:

**对问题进行词性标注(POS tagging)** 这一步中将问句拆分成若干独立的字或者词,再将这些独立的字词根据其词性进行标注。例如,对于问题“谁是美国总统?”,其词性标注为:谁\_提问词/是\_动词/美国\_名词/总统\_名词。

**答案分类体系设计** 问答系统的设计者对所有可能的答案进行分类。通过寻找与问题类别相一致的答案,系统可以缩小搜索范围。例如在问答系统SiteQ[59]中,将答案分成了18种类别,包括了“数字”、“人物”、“地点”等等。

**问题与答案的类型配对[66]** 依据答案分类体系和问题理解,系统识别问题类型,作为精确答案抽取的重要特征。



## 2.2. 生成搜索关键词

对问题处理以后，系统将问题转化成一组带有权重的关键词，并使用这组词在纯文本语料中进行搜索。系统需要合理地生成关键词，从而优化搜索效果。

**去停用词** 停用词是指出现频率高但检索意义的词。例如：“的、是、啊、太”等等。停用词表可以通过统计文本中各词出现的频率来生成。

**搜索词打分** 系统用剩下的词搜索。为了获得更好的搜索结果，系统给每个关键词一个权重，来表示其与问题的相关性。例如对于问题“谁是美国总统”，“美国”、“总统”这两个词会获得高权重。

## 2.3. 文章检索

文章检索指使用信息检索的方法，通过相似度匹配来搜索与用户问题匹配的文本。典型的基于统计的文档与问题相似度的方法是Opaki BM25[79]。

**Opaki BM25算法** 是一个经典的运用关键词匹配相关文档的算法。它有很多不同变形，其核心公式如下：

$$bm25(s_1, s_2) = \sum_{i=1}^n IDF(w_i) \cdot \frac{f(w_i, s_1) \cdot (k_1 + 1)}{f(w_i, s_1) + k_1 \cdot (1 - b + b \cdot \frac{|s_1|}{avgs_l})} \quad (2.1)$$

这里 $IDF(w_i)$ 指的是inverse document frequency(与该单词在QA语料中出现的文档个数有关)， $f(w_i, s_1)$ 是 $w_i$ 在 $s_1$ 中的词频， $|s_1|$ 是 $s_1$ 的长度， $avgs_l$ 是纯文本语料库的平均句子长度。 $k_1$ 和 $b$ 是超参数。

算法会惩罚在一篇文章中重复多次的关键词，词频率越大，它的新频率对于相似度的贡献的增加越小。例如一个词重复出现了2次，那么它的权重是2，但如果出现了4次，权重仅为3.7。这样可以防止单个高频词对于匹配的影响过大。这个算法也惩罚那些在所有文章中都经常出现的词（例如“的”，“了”），减小它们对结果的影响。

## 2.4. 段落抽取

段落是包含着答案的小节，在具体实现的时候可以使用诸如“段落是包含着答案的4句话”的规定。在系统通过搜索获得了若干篇与问题相关并包含答案的候选文档集后，在文本中匹配最有可能包含答案的3-4句话，即抽取相关段落。

在这一过程中，问答系统计算问题与段落的匹配度。除了基于统计的方法，近年来学术界流行将深度学习的方法运用进来。

**基于卷积神经张量网络(convolutional neural tensor network)的算法** [76]提出了卷积神经张量网络的算法进行问题-答案配对度计算。系统把问题和答案中的词向量化,再对这些向量做卷积。从而得到一句话的定长向量表示。再对向量训练神经网络,计算匹配度。系统使用该模型在百度知道的问题-答案集里地寻找与问题有关的答案,实验取得良好效果。

## 2.5. 答案抽取

系统从相关段落中抽取精确的答案。目前最先进的算法依然使用了深度神经网络。

**基于循环神经网络模型的算法** [81]提出用循环神经网络模型来训练用来抽取答案的算法。循环神经网络由若干结构一致的神经元串联而成。每一个神经元的输入不仅包括上一层的输出,还包括了同一层上一神经元的输出。这一模型对序列数据处理有很大优势。而自然语言文本正是典型的序列数据。

## 2.6. 不足之处及使用知识图谱的动机

由于自然语言文本的复杂性,利用信息检索实现精确的答案抽取是很困难的。这也导致了在TREC 2007的问答竞赛[22]中,除第一名、第二名的准确率达到70%和49%,其余系统准确率都在30%之下。造成这一问题主要有以下两个原因:

- **知识表示多样性** 文本中对于同一知识的表示是多样的。但当前的问题及文本理解,其核心还是关键字匹配。该方法无法处理问题和文本语料中知识表示方式的不一致。尽管基于深度学习等的方法已经可以从语义角度学习问题和文本语料的对应关系,但是数据的稀疏性依然使得这一学习过程非常困难。
- **知识不一致性** 同一知识点可能在文本中多次提到。但是由于文本质量无法做到100%的准确,一个知识点对应的值在不同文本片段中是不同的。因此系统还需要进行正确值的判别 [70]。这进一步增加了基于文本问答的难度。
- **海量本文查询性能要求** 由于文本数据的海量特性,对于给定问题查询其答案需要快速精确的候选答案定位。现有方法往往利用关键词的倒排表来作为索引,这导致很多正确答案无法被查询处理到。

相对而言,计算机更擅长处理结构化的数据。因此使用一个高质量的结构化数据源作为问答语料,其效果往往好于基于自然语言语料的信息检索方法。而知识图谱正是这类数据的代表。具体而言知识图谱有以下几个优势。这些优势促使问答系统使用知识图谱作为问答系统的数据源。

- **知识表示语义性** 知识图谱不止有实体，还有丰富的属性和语义关系。这些语义关系为问题理解和回答提供了更充分的语义特征。
- **知识的高准确率** 知识图谱基于人工构建，或高质量数据源的结构化爬取[6]。这保证了其高准确率。
- **检索友好** 结构化的知识表示，使得计算机可以通过结构化查询进行知识检索。例如对于常见的RDF型知识图谱，计算机只需要对subject建立索引，即可直接检索查询所有给定实体的相关属性。

### 第3节 基于知识图谱的问答系统及其不足

#### 3.1. 问题分析

给定问题和知识图谱时，系统面临两个核心挑战：用何种方式来表示问题语义（问题表示设计），以及如何将该问题表示映射到知识图谱上的结构化查询（语义匹配）。

- **问题表示设计**：不同的问题对应到知识图谱中数以千计的关系中，一种关系可以有数以千计的问题模板。例如，表1.1中的问题①和问题②都询问了檀香山市的人口，但它们的表达方式大相径庭。对于不同的问题形式，问答系统需要不同的问题表示。这些问题表示必须满足（1）使相同意图的问题具有一致的问题模型；（2）区分不同意图的问题。在所使用的问答语料库中，系统找到了2782种问题意图的约两千七百万种问题形式。如何设计问题表示模型来处理这一情况是一个巨大挑战。
- **语义匹配**：在实现一个问题的表示之后，系统需要将这一表示映射为结构化查询。对于事实型问题，结构化查询主要依赖于知识图谱中的属性。由于属性和表示模型的不一致，寻找这样的匹配并非一件直接的工作。例如，在表1.1中，系统需要知道问题①与属性人口数有着相同的语义。此外，在RDF图中，许多二元关系并不仅仅对应一条边，而是某种更为复杂的结构：在图1.1中，“配偶”关系由 $marriage \rightarrow person \rightarrow name$ 的路径表示。对于所使用的知识图谱，超过98%的关系对应于类似的复杂结构。

#### 3.2. 以往研究

基于结构化数据的自然语言问答，在上世纪六十年代就已经开始被研究了。在知识图谱出现前，这一研究往往关注在知识本体（ontology）、语义网络（semantic web）上作问答[63]，或者被称为对数据库的自然语言交互[3]。其中的佼佼者包

括Lunar[102]、Rendezvous[18]、Ladder[43]等。在大规模知识图谱出现之后,研究的重点开始转向基于知识图谱的问答系统。对于以往的研究工作,本文根据其问题表示方式,粗略地将其划分为以下几个种类。

1. **基于规则模板** [72]。基于规则模板的实现通过人工构造规则模板将问题映射到属性。这导致了较高的准确率和较低的召回率(对问题多样性的较低覆盖率),因为为人工构建规则对通用问题是不可行的。
2. **基于句法规则** [102]。基于句法规则的实现,依然源自人工定义规则。只是此时的规则建立在句法上,即规则预先定义好句法的表示及其对应的属性。这样可以更深入的理解一个问题。然而人工构建的句法规则虽然可以带来高准确率,但是无法覆盖多样的问题形式。
3. **基于语义文法** [43, 100]。基于语义文法的实现,与基于句法规则的实现相类似。只是此时规则建立在语义文法上。语法解析树是一个典型的语义文法特征。规则会建立在对于语法树的描述上。与基于句法规则的实现类似,此方法也存在覆盖率的问题。
4. **基于关键词** [98]。基于关键词的实现利用关键词匹配将问题中的关键词映射到属性。通过识别出问题中“population”这一关键词并将其映射到相应的属性,这一方法可能可以回答类似表1.1中问题⑥这样的简单问题。但是一般而言,由于知识图谱中属性的单一描述并不能匹配自然语言中的不同问题描述形式,基于关键词的方式寻找问题与属性的映射是困难的。例如,在问题①和③中不能寻找到“population”这一关键词。
5. **基于同义词** [97, 107, 116, 112]。基于同义词的方法将属性的同义词纳入考虑,从而扩展了基于关键词的问题表示。这一方法先生成各个属性的同义词,再寻找问题与这些同义词间的映射。DEANNA [107]是一个典型的基于同义词的问答系统。其主要思想是将问答系统简化为对属性和候选同义词(问题中的词语、词组)之间的语义相似度计算。它使用维基百科来计算相似度。例如,通过理解问题中total number of people是属性“population”的同义词,可以回答表1.1中的问题③。gAnswer [116, 112]更进一步地提高了同义词表示的精确度。它学习同义词的更加复杂的子结构。然而,所有这些方法都不能回答表1.1中的问题①,因为how many、people、are there都没有与属性“population”的明显对应关系。

因此,这些工作都不能解决前文所述的挑战。对于基于规则的实现,它需要无法承担的人工标签工作。对于基于关键词或同义词的实现,单个词语或词组不能完全地表达问题的语义意图。系统需要整体地理解问题。而当问题变得更加复杂,或是需要

映射到知识图谱中的复杂结构属性（例如问题⑥、⑦）时，更会让先前的方法变得无从施展。

另一方面，当前的这些工作都缺乏系统性。它们直接研究从问题到知识图谱的映射。本文认为对于问题的理解本身的研究，和对于知识图谱的研究，以及如何将它们应用在问答系统中，将会大幅提升系统的回答效率。但是这些研究在之前工作中是缺失的。本文研究的系统性构成了的一大研究优势。对于问题理解的研究，论文研究了实体理解和动词理解。对于知识图谱的研究，论文研究了如何自动化的构建领域知识图谱，以及如何将问答系统中的自然语言处理技术迁移到具体领域。



## 第三章 基于局部搜索的语义社团挖掘

问答系统对问题理解的基础是对问题实体的理解。社团搜索有助于实体语义分析。对于一个语义网络或知识图谱中的结点，本章的目标是找到其所属的最佳社团。直觉上，该社团是它的邻居区域的一部分。现有方法使用全局搜索来寻找最佳社团。这些算法尽管非常直接，但是时间消耗极大，尤其是整个网络中的结点都需要被遍历到。

本文提出了一个局部搜索的策略，在给定查询点的邻居中进行搜索来找到最佳社团。这个最佳社团的度量是它的最小度。论文会说明最小度在局部搜索中是非单调的，并为此提出了局部搜索的相关理论证明和算法。不同数据集上的充分实验验证了算法效果。

### 第1节 引言

语义网络 and 知识图谱是复杂网络，它们具有社团结构。也就是说，一个网络可以被划分成若干个社团，社团内部的点连接紧密，而社团外的点连接稀疏[71]。在真实语义网络中社团结构是和节点语义紧密相连的。它的社团中的点描述了这个点的相关特征。基于语义网络的实体语义计算为上层语义计算提供基本支持。

由于语义社团的重要意义，本章提出了社团查找问题，即对于给定结点，找到其属于的最可能社团。语义社团查找问题对于很多真实应用都是有意义的[88]，特别是语义扩展和实体语义理解：

- **语义扩展** 在信息检索中，当一个用户提交一个查询时，例如“image”，他可能同时对其它诸如“picture”，“photos”等查询结果也感兴趣。使用语义链接网络 [115]，查询可以扩展为语义社团中的其它单词，从而扩充查询结果。
- **实体语义理解** 实体理解的一个基本问题是语义理解，即理解给定实体/概念的主题。语义社团是比传统的“词袋”(bag of words)模型更为准确的一种主题模型。给定知识图谱，实体语义理解可以转换为社团搜索问题：即给定某个实体/概念，查找其所隶属的最佳社团。

为了高效的进行语义社团搜索，需要首先确定什么是一个好的语义社团。一个语义社团的好坏，其评估标准在于其内部结点的粘结度。一个常见的粘结度的度量方式是最小度，即，该社团结点的导出子图的度的最小值。一个好的语义社团就是一个具有较大最小度的社团。例3.1说明了这一点。



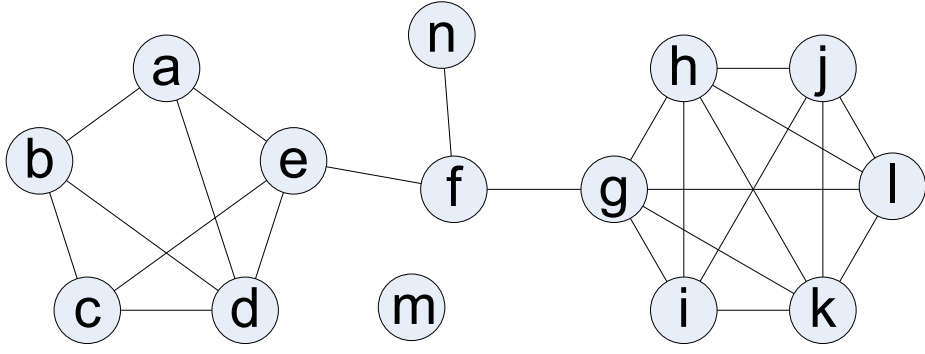


图 3.1: 网络示意图

**例 3.1 (最小度).** 假设用户希望在图3.1中找到包含结点 $a$ 的最佳社团。直觉上来说,  $V_1 = \{a, b, c, d, e\}$ 是一个包含 $a$ 的高度关联的社团。 $V_1$ 所对应结点集的导出子图的最小度是3。当加入一个新的结点 $f$ 进来, 它的最小度就会降低为1。

另外一个度量方式是平均度。在这种度量下, 对于结点 $a$ 的最佳社团就会进一步包含结点 $f$  和一个稠密子图 $V_2 = \{g, h, i, j, k, l\}$ , 这样 $V_1 \cup \{f\} \cup V_2$ 的平均度数就达到了3.8, 大于了 $V_1$ 的平均度3.2。然而直觉上,  $V_1$ 和 $V_2$ 更倾向于是作为两个分隔的社团, 因为它们之间是通过一个非常弱的连接 $f$ 来关联的。因此, 最小度更好的捕捉了关于社团粘结度的度量。所以本章中使用最小度作为社团搜索的度量。这样, 语义社团搜索的问题, 就转变为了查找包含查询点的点集, 使其导出子图的最小度最大。

在以上的论述之外, 下属理由同样促使本章使用最小度作为衡量社团结构是否良好的标准。首先, 最小度是一个图的最基本特征之一。例如, 它被用来描述随机图的演化[11], 以及图的可视化 [34]等问题。在本文的关于最小度的相关性质研究, 也可以被应用在其他相关问题中。第二, 在复杂网络分析中, 最小度已经被广泛的应用为描述结点间粘结程度的度量[83, 88, 25]。这最早可以追溯到Seidman 在1983年的工作[83]。作者比较了最小度和很多其他的关于粘结度的度量, 例如连接度和图的直径, 最终发现最小度是建模图的粘结度的一个优秀度量。在最近的社团查找研究中[88], 作者同样将最小度作为社团的度量。他们同样发现最小度是比其它度量更为优秀, 包括子图的平均度数, 密度 (度量为 $\frac{2|E|}{|V|(|V|-1)}$ , 这里 $|V|$ 和 $|E|$ 分表表示图的结点数和边数) 等。

一个社团查找的直接方法被认为是全局搜索。但该方法的应用前景不乐观, 因为为了确定某个结点的社团结构, 它需要检索整个网络。注意到在最差情况下, 整个网络可能是最佳的社团。这种情况发生在整个网络构成一个完全图的时候。

例3.2展示了一个典型的图3.1中的全局搜索过程。它从整个图出发, 迭代的删除图中的不可能成为答案的结点。这个过程不断重复, 直到没有结点可以被删除。

**例 3.2 (全局搜索).** 假设用户希望找到图3.1中对于结点 $j$ 的最佳社团。算法会迭代的从图中删除度数最低的结点。这样结点 $m, n, f, a, b, c, d, e$ 就会被轮流删除。这样算法



就得到了社团  $V' = \{g, h, i, j, k\}$ ,  $j$  在社团中, 并且保证图的最小度最大。容易发现所有  $V'$  的子图的最小度都低于  $V'$ 。因此,  $V'$  就是  $j$  的最佳社团。

由于需要检索整个图, 全局搜索的复杂度是非常高的。这对于一个大的网络是无法接受的。本文提出了一种基于局部搜索的策略。其直觉在于, 一个结点的最佳社团, 一定在这个结点的局部邻居范围中。所以不必利用整个网络来寻找这样一个社团。局部搜索的过程如下: 首先从查询结点出发, 最开始时, 目标社团就是查询结点本身。随着算法不断探索当前社团的邻接结点, 新的结点被加入, 社团被扩展。当算法找到最优社团时, 过程停止。

问题在于, 算法是如何知道当前找到的社团已经是最优社团的? 如果最小度的度量方式是随着节点加入而单调的, 即当新的结点加入时, 最小度变低了, 那问题很容易解决: 算法可以在最小度开始变低的时候停止进一步搜索。然而遗憾的是, 最小度这个度量是不单调的。这被展示在例3.3中。

**例 3.3 (局部搜索和不单调性).** 假设算法希望找到图3.1中结点  $a$  的最佳社团。假设当前枚举到的社团包含了  $a$  和它的直接邻居  $b, d, e$ 。当前社团  $S = \{a, b, d, e\}$  的导出子图的最小度是2。为了增大社团, 算法考虑引入结点  $c$  或者  $f$ 。加入  $f$  会使得社团的最小度变为1, 而加入  $c$  则使得社团的最小度变为3。

很明显, 如例3.3所示, 最小度是不单调的。即对于当前社团  $S$  来讲, 加入一个新的邻接点进入  $S$ , 其新社团的最小度既可能变大, 也可能减小。这就对算法何时停止局部搜索和社团扩张提出了挑战。

本章给出了局部搜索的理论证明和相关算法, 研究了一个邻接点被加入社团的充分条件。如果没有邻接点满足这一充分条件, 那么算法停止这一搜索过程。本章证明了这个充分条件是存在的。但是在最差情况下, 对它的计算会和全局搜索的复杂度一致。然而, 一般来讲, 一个典型的局部搜索总是可以找到一个远比全局搜索复杂度低的算法。

本章的内容被组织如下: 第2节介绍一些基本的背景信息和问题定义。论文明确提出了两个语义社团查找问题: CST和CSM。第4节讨论了如何用全局搜索解决此问题。第5节提出了一些局部搜索策略来解决CST问题。第6节中提出了对CSM问题的解法。第7节包含了相关实验。第8节做小结。

## 第2节 问题定义

本节定义社团的良好程度, 以及社团搜索的具体问题, 同时也会讨论此问题的挑战。这项工作只考虑简单图, 即无自环、无重复边的图。另外, 本章只考虑无权无向图。

$G(V, E)$	一张结点集为 $V$ 边集为 $E$ 的图
$G[H]$	由 $G$ 中结点集 $H$ 。它包括边集 $(H \times H) \cap E$
$deg_G(v)$	结点 $v$ 在图 $G$ 中的度
$\delta(G)$	$\min\{deg_G(v)   v \in V\}$
$H^*(G, v)$	图 $G$ 中结点 $v$ 的最佳社团
$m^*(G, v)$	$H^*(G, v)$ 的社团度量
$V_{\geq k}$	$\{v   deg_G(v) \geq k\}$
$C$	候选结点集
$maxcore(G, v_0)$	$G$ 中结点 $v_0$ 的最大核

表 3.1: 符号标记

### 2.1. 问题定义

用 $G(V, E)$ 表示无向图 $G$ ，包括结点集 $V$ 和边集 $E$ 。对于任意结点集 $H \subseteq V$ ，它对于 $G$ 的导出子图被表示为 $G[H]$ 。 $G[H]$ 的结点集为 $H$ ，边集为 $(H \times H) \cap E$ 。另外，用 $deg_G(v)$ 来表示结点 $v$ 在图 $G$ 中的度。很明显，因为 $G[H]$ 是 $G$ 的子图，有 $deg_{G[H]}(v) \leq deg_G(v)$ 。在表3.1中总结了本文中涉及的符号表示。

**定义 3.4** (社团度量). 假设图为 $G(V, E)$ ，对于它的结点子集 $H \subseteq V$ ，考虑以 $H$ 作为一个社团的社团好坏程度度量。这个度量被定义为它的最小度：

$$\delta(G[H]) = \min\{deg_{G[H]}(v) | v \in H\} \quad (3.1)$$

最小度是最常见的社团度量方式之一[88]。就像例3.3中展示的那样，最小度的一个重要特性是它的非单调性。也就是说， $\delta(G[H \cup \{v\}])$ 的最小度不一定比 $\delta(G[H])$ 的最小度大。由于最小度的非单调性，使用局部搜索策略找到最小社团的问题是很困难的。

**问题定义 3.5** (CSM). 对于图 $G(V, E)$ 以及图中的一个结点 $v_0 \in V$ ，找到 $H \subseteq V$ ，使得(1) $v_0 \in H$ ;(2) $G[H]$ 是一个连通子图;(3) $\delta(G[H])$ 在 $H$ 的所有可能选择中最大。本章将这个问题定义为CSM，即有最大社团度量的社团搜索 (community search with the maximality constraint)。

对于图 $G$ 和结点 $v$ ，假设 $m^*(G, v)$ 表示其最大可能的最小度， $H^*(G, v)$ 表示任意一个具有该最小度的社团，有

$$0 \leq m^*(G, v) \leq deg_G(v) \quad (3.2)$$

注意这里最优解不一定是唯一的。一般来讲， $m^*(G, v)$ 是被 $v$ 和 $G$ 的图结构决定的。

不同于直接找到社团最优解，在一些应用中，用户可能对于找到满足度量大于某一阈值的社团感兴趣，即 $\delta(G[H]) \geq k$ 。这里 $k$ 是一个给定的阈值限制。通过控制 $k$ ，用户可以控制所找到的社团的大小。

**问题定义 3.6 (CST).** 对于给定图 $G(V, E)$ 以及图中的一个结点 $v_0 \in V$ ，找到 $H \subseteq V$ ，使得(1) $v_0 \in H$ ; (2) $G[H]$ 是一个连通子图; (3) $\delta(G[H]) \geq k$ 。定义这个问题为带阈值限制的社团搜索问题 (*community search with the threshold constraint*)  $CST(k)$ 。

**例 3.7.** 以图3.1中的语义网络作为例子。假设查询点为 $a$ 。在CSM问题中， $H = \{a, b, c, d, e\}$ 是最终的最佳社团，因为 $\delta(G[H]) = 3$ ，而其他候选社团的最小度都小于3。在 $CST(k)$ 问题中，当 $k = 3$ 时，它的解依然是 $H$ 。如果 $k = 2$ ，问题会有多个解，包括 $\{a, b, d\}$ ,  $\{a, d, e\}$ 和 $\{a, b, c, d, e\}$ 等。

例3.7展示了CSM和CST问题的例子。它们都可能对应多个解。实际上，CST可能会产生关于图的大小的指数规模的解。因此本文只注重找到它的一个解。同时，用户可能希望这个解的结点个数最少。这个新问题被定义为mCST如下：

**问题定义 3.8 (mCST).** 对于给定图 $G(V, E)$ 以及图中的一个结点 $v_0 \in V$ ，找到 $H \subseteq V$ ，使得(1) $v_0 \in H$ ; (2) $G[H]$ 是一个连通子图; (3) $\delta(G[H]) \geq k$ ; (4) $H$ 的结点个数最少。本文将这个问题称为 $mCST(k)$ 。

不幸的是，mCST问题是NP完全的。本章将会在第3节证明这一点。本章会聚焦解决CST和CSM问题。

## 2.2. CSM和CST关系分析

CSM是一个最优化问题，它的对应的判定问题是：判定是否存在 $H \subseteq V$ ，满足上述CST问题定义的三个条件。很明显，CST就是CSM问题的构造版本。也就是说，算法不仅需要判断有效解是否存在，也需要在解存在时，构建或找到一个具体解。在这些明显的关系之外，本节进一步建立了一些数值上的CSM和CST的相关性分析，作为解决这两个问题的基础。

**命题 3.9 (CST( $k$ )的向下传导性).** 如果 $H$ 是 $CST(k)$ 的一个解，那么对于 $k' < k$ ， $H$ 也是 $CST(k')$ 的一个解。

**命题 3.10.** 对于图 $G(V, E)$ 和查询点 $v$ ，如果 $H$ 是 $CST(k)$ 的一个解，那么 $m^*(G, v) > k$ 。

**命题 3.11 (剪枝规则).** 对于结点 $v$ ，满足 $\deg_G(v) < k$ ， $v$ 不属于任何 $CST(k)$ 的解。

命题3.9和3.10导致了，如果有一个关于CST问题的多项式解法，那么可以得到一个关于CSM的多项式解法。由于 $m^*(G, v)$ 在 $[0, \deg_G(v)]$ 的范围内，算法可以使用二分法，从 $\lceil \frac{N-1}{2} \rceil$ 开始迭代的访问范围的中值，并判断该中值是否是有效解。在这个方法下，算法可以在复杂度 $O(\log \deg_G(v) f(N))$ 下解决CSM问题，这里 $f(N)$ 是CST问题的复杂度。

对于CST和CSM问题，本章只期望寻找其中的一个解，因为可能有指数级别的解。为了说明这一点，考虑包括 $N$ 个度为1的结点和1个度为 $N$ 的结点的情况（图3.2）。假设结点 $v_c$ 的度为 $N$ ，那么很显然有 $m^*(G, v_c) = 1$ 。但是 $H^*(G, v_c)$ 可能是任何包含 $v_c$ 的子集。这样这个图中最优解的个数就是 $\Theta(2^N)$ 。为了避免这种指数级的枚举，在本文的问题中，只查找其中的一个解。

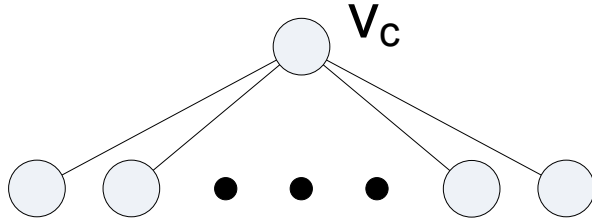


图 3.2: 一个具有1度为 $N$ 和 $N$ 个度为1的结点的图

### 第3节 mCST的NP完全性

现在证明第2.1节提出的mCST问题是NP完全的。为了证明这一点，先证明一个相关问题是NP完全的，然后将此问题规约到mCST。

**引理 3.12.** 对于给定图 $G$ ，查询点 $v_0$ ，和整数 $k$ ，如果存在一个团 $C$ ，即任意 $v \in C$ ， $|C| = k + 1$ ，那么 $C$ 是 $CST(k)$ 问题的一个结点集最小的解。

**问题定义 3.13 (MCC).** 对于图 $G(V, E)$ 以及结点 $v_0 \in V$ ，寻找包含 $v_0$ 的最大团。

**引理 3.14.** MCC是NP完全问题。

**证明.** 很明显，MCC属于NP类问题。此证明将一个经典的NP完全问题，最大团问题，规约到MCC问题。对于任意图 $G(V, E)$ ，构建一个新的图，方式如下：加入一个新的结点 $v_0$ ，并且将 $v_0$ 关联到 $G$ 中的所有结点。也就是说， $V' = V \cup \{v_0\}$ ， $E' = E \cup \{(v_i, v_0) | v_i \in V\}$ 。这样，一个 $G'$ 中MCC的解，就是 $G$ 中的一个最大团。□

**定理 3.15.** mCST问题是NP完全的。

**证明.** 很明显，mCST是属于NP类问题。此证明通过将MCC规约到mCST的方式，来证明它的NP完全性。MCC是一个最优化问题，考虑它的判定问题版本：判定是否 $G$ 是一个大小不小于 $k$ 的包含 $v_0$ 的团。可以为mCST构造如下判定问题。判定是否

存在一个解  $H \subseteq V$ ，使得  $|H| = k$  并满足 mCST 的条件：(1)  $v_0 \in H$ ；(2)  $G[H]$  是连通的；(3)  $\delta(G[H]) \geq k - 1$ 。如果  $H$  是它的解，那么很明显  $G[H]$  是一个团，因为任何  $H$  中的点的度都满足  $\geq k - 1$ 。□

## 第4节 全局搜索

本节阐述 CST 和 CSM 问题的全局搜索解决方案。全局搜索需要遍历图中的所有结点和边，因此对于大图来讲时间消耗巨大。

### 4.1. k核和最大核

为了理解全局搜索，首先定义 k 核和最大核的概念。

**定义 3.16 (k核).** 图  $G$  的一个子图，如果该子图是结点个数最多的满足子图中所有结点的度数都至少为  $k$  的子图，则称之为  $G$  的  $k$  核。注意  $k$  核可能有多个连通分支。

**定义 3.17 (最大核).** 对于结点  $v$ ，它的最大核是所有包含  $v$  的，满足  $k$  最大的  $k$  核，本文用  $\text{maxcore}(v)$  来表示它。

**例 3.18.** 考虑图 3.1 中的图  $G$ 。它的  $\{a, b, c, d, e, g, h, i, j, k, l\}$  对应的导出子图是一个 3 核子图，它的  $\{g, h, i, j, k, l\}$  对应的导出子图是一个 4 核子图，同时也是图  $G$  的最大核。而  $\{a, b, c, d, e\}$  的导出子图则是图  $G$  中关于结点  $e$  的最大核，即  $\text{maxcore}(G, e)$ 。

例 3.18 说明了 k 核和最大核的概念。可以使用全局搜索的方式寻找 k 核如下：迭代的从  $G$  中删除结点的度数小于  $k$  的结点，直到没有更多的结点可以被删除。这一过程需要访问每个结点和每条边来确定其度数。最大核的算法可以类似设计，每次删除度最小的结点。因此，k 核和最大核的算法复杂度都是  $O(|V| + |E|)$ 。

### 4.2. CST和CSM的解决方案

对于图  $G$  和给定查询点  $v_0$ ，接下来介绍如何利用 k 核和最大核的算法结果得到 CST 和 CSM 的解。首先，考虑问题 CST(k)。如果结点  $v$  有任何包括 CST(k) 的解，则称  $v$  是可接受点。假设  $A$  是所有对于 CST(k) 的可接受点的集合。类似的，定义  $A'$  为所有 CSM 的可接受点。例 3.19 展示了可接受点集的例子：

**例 3.19.** 考虑图 3.1 中图  $G$  和查询点  $e$ 。对于 CSM 问题，有  $m^*(G, e) = 3$  和  $H^*(G, e) = \{a, b, c, d, e\}$ 。由于没有其他的  $H^*(G, e)$  存在，可接受点集即为  $A = \{a, b, c, d, e\}$ 。对于  $\text{CST}(2)$ ，它的解包括  $\{a, b, c, d, e\}$  和  $V - \{m, n\}$ 。因此有  $A = V - \{m, n\}$ 。

接下来会说明  $A$  是  $G$  的 k 核的子集。同样， $A'$  也是  $\text{maxcore}(v_0)$  的一个子集。更具体的，有：



**引理 3.20.** 对于给定图 $G$ ，查询点 $v_0$ ，包含点 $v_0$ 的 $k$ 核的连通分支是 $CST(k)$ 问题的一个解。同时对于任意 $CST(k)$ 的解 $H$ ，都有 $H \subset C_k$ 。

**引理 3.21.** 对于给定图 $G$ ，查询点 $v_0$ ，包含点 $v_0$ 的 $maxcore(v_0)$ 的连通分支被标记为 $C_{max}(v_0)$ 。 $C_{max}(v_0)$ 是 $CSM$ 问题的一个解。另外，对于任何其他的 $CSM$ 问题的解 $H$ ，有 $H \subset C_{max}(v_0)$ 。

**证明.** 根据定义， $C_{max}(v_0)$ 是 $CSM$ 问题的一个解。如果存在另一个解 $H \neq C_{max}(v_0)$ ，那么 $H \cup C_{max}(v_0)$ 将会成为一个 $maxcore(v_0)$ 中的连通分支，并且 $H$ 和 $C_{max}(v_0)$ 都包含 $v_0$ 。这与 $C_{max}(v_0)$ 的定义矛盾。□

引理3.20表明，一个结点 $v$ 是 $CST(k)$ 问题的可接受点，当且仅当 $v \in C_k$ 。引理3.21表明，一个结点 $v$ 是 $CSM$ 问题的可接受点，当且仅当 $v \in C_{max}$ 。不幸的是，检查这个重要条件的复杂度，等同于全局搜索本身。

为了解决 $CST$ 问题，引理3.20表明可以通过迭代去掉度小于 $k$ 的结点的方式。剩下的包含查询点的连通分支就是一个有效的解。

为了解决 $CSM$ 问题，引理3.21表明了，需要找到 $v_0$ 对应的最大核的连通分支。本文通过贪心算法[88]来解决此问题。假设 $G_0 = G$ 。算法从 $G_0$ 开始删除其邻接点最少的结点，并不断迭代。逐渐得到图序列 $G_0, G_1, \dots, G_t$ ，直到下一个要被删除的结点是 $v_0$ 。然后， $G_t$ 中包含 $v_0$ 的连通分支中，拥有最大的 $\delta(G_t)$ 的那一个即为最优解。

以上两个问题的解法的复杂度都是 $O(|V| + |E|)$ ，因为它需要遍历到图中的所有结点和边。

## 第5节 CST问题的局部搜索解法

本节设计对于社团搜索的局部搜索算法。它的最大挑战在于解决最小度这一度量的非单调性，这样才可以保证算法可以在图的局部完成整个过程。本节首先提出一个基准局部搜索算法，它的复杂度是指数级的。接着会展示线性模型的一般框架。最后，在第5.3.3节和第5.3.3.3节，论文会对模型给出优化实现。就像之前提到过的， $CSM$ 可以基于 $CST$ 的解法来解决。因此，本节首先解决 $CST$ 问题。

### 5.1. 基准算法

本节首先给出关于 $\delta(\cdot)$ 的单调性的深度分析，并介绍一些符号定义。考虑一个从查询结点的探索过程。在每一步，算法探索一个新的结点，直到得到解 $H$ 。假设 $v_0, v_1, \dots, v_t$ 是在得到 $H$ 之前探索的一系列结点。设 $H_i = \{v_0, \dots, v_i\}$ 。本节将说明，大体上 $\delta(\cdot)$ 是一个关于 $H$ 的非单调函数。更明确的， $\delta(H_i)$ 和 $\delta(H_{i+1})$ 的大小关系是不确定的。显然， $\delta(\cdot)$ 的单调性取决于结点被加入 $H$ 的顺序。一个有趣的现象是，对于任

意  $v_0 \in H$ ，总是可以找到一个结点序列（每个结点都在  $H$  中），从结点  $v_0$  开始，满足此序列的  $\delta(H_i)$  是关于  $i$  单调不减的。

**定理 3.22.** 对于图  $G$  中任意结点  $v_0 \in H$ ，总是存在一个  $H$  的结点序列  $v_0, v_1, \dots, v_t$ ，满足  $\delta(H_i) \leq \delta(H_{i+1})$ 。

**证明.** 这等价于证明，算法可以从  $H$  出发，一个接一个的删除结点，并保证每次结点删除之后，剩下结点的导出子图最小度不减。假设当前的点集为  $H'$ 。如果  $H' = \{v_0\}$ ，算法已经找到了此结点序列。如果  $H' = \{v_0, v_i\}$ ，那么删除结点  $v_i$ ，或者降低最小度（如果  $(v_0, v_i) \in E$ ），或者最小度不变（如果  $(v_0, v_i) \notin E$ ）。接下来，考虑当  $H'$  中有更多  $v_0$  之外的结点的情况。在这种情况下，势必存在结点  $v \in H'$ ， $v \neq v_0$ ， $\delta(G[H']) \geq \delta(G[H' - \{v\}])$ 。这样算法移除结点  $v$  即可。如果这样的结点不存在，即对于  $\forall v \in H', v \neq v_0$ ，都有  $\delta(G[H']) < \delta(G[H' - \{v\}])$ 。这仅仅发生在  $v$  是  $H'$  中度最小的结点的时候，因为移除一个度非最小的结点，只会使图的最小度不变或者降低。这样，至少有两个或者更多这样的  $v$ 。而删除其中的一个结点，并不会造成一个更大的最小度。  $\square$

定理3.22表明，总是有一个单调的探索顺序来得到  $H$ 。定理3.22同时表明，任何  $CST(k)$  的解  $H$  都可以被从  $v_0$  开始的一个结点序列所构造，并且满足对于任意的  $i$ ，都有  $\delta(H_i) \leq \delta(H_{i+1})$ 。这个特性的存在，为算法按照单调的方式找到解  $H$  提供了必要而非充分的条件。为了表示这个不充分性，参见图3.1中的图，对于  $CST(3)$  问题和查询点  $e$ ，任意的从  $e, f$  开始的结点序列都不会产生一个有效的解。但是很明显  $\delta(G[e, f])$  是比  $\delta(G[e])$  要大的。

定理3.22导致了一个非常直接的算法，它被展示在算法1中。从  $H' = \{v_0\}$  开始，通过调用 *search* 函数来进行。*search* 函数不断地枚举每个结点  $v$ ， $v$  来自当前枚举的  $H'$  的邻居，使得  $\delta(H' \cup \{v\}) \geq \delta(H')$ 。如果一个解被找到了，那么这一过程就停止。否则，它会回溯并继续调用 *search* 函数。定理3.22保证了这一解法的正确性，即不断迭代总会找到  $CST(k)$  问题的最优解的。而这个解也可以被直接扩展到  $CSM$  问题的解中，本文将这部分细节隐去。很明显，这个算法的复杂度依然是指数级别的。这就促使本文开发一个更为高效的算法。接下来介绍一个线性算法模型。

## 5.2. 一个 CST 问题的解决框架

这一节介绍一个 CST 问题的局部搜索解法框架。如算法2所示，在一个较高的层面上，它包含了三个简单的步骤。首先，算法检查当前的子图是否已经满足了  $CST(k)$  的必要条件。第二，算法执行 *candidateGeneration()*。即，算法从当前结点集合的邻居中进行探索，并生成候选答案集合  $C$ ，此集合可能包含了问题的一个解。在大部分情况下，第二步已经找到了有效的解并且算法可以结束了。而如果没有，在最后一步，算法会采用全局搜索的  $k$  核解法，来找到其导出子图并返回结果。

**Algorithm 1:** *Search()*输入:  $G(V, E), H', k$ 输出:  $H$ 

```

1: if  $\delta(G[H']) = k$  then
2:    $H \leftarrow H'$ 
3:   return
4: end if
5: for all  $H'$ 的邻居结点 $v$  do
6:   if  $\delta(G[H' \cup \{v\}]) \geq \delta(G[H'])$  then
7:      $Search(H' \cup \{v\})$ 
8:     if  $H \neq \emptyset$  then
9:       return
10:    end if
11:   end if
12: end for

```

只要 $candidateGeneration()$ 的过程没有删除任何可接受点, 算法2保证可以返回一个有效的解。这在命题3.23中被说明。

**命题 3.23.** 给定图 $G$ 和查询点 $v_0$ , 如果 $H \subseteq V$ 是一个 $CST(K)$ 的解, 那么对于任意 $H' \subseteq V$ , 一个 $G[H \cup H']$ 的 $k$ 核中包含 $v_0$ 的连通分支即为 $CST(k)$ 的一个有效解。

接下来会在第5.2.1节进行算法复杂度上界估计。论文在第5.2.2节展示直接的 $candidateGeneration()$ 过程, 并深度分析其复杂度和效率。

### 5.2.1. 上界

在算法开始搜索之前, 是否能直接判定一个图对于 $CST(k)$ 和结点 $v$ 是有解的? 显然, 如果 $v$ 的度是小于 $k$ 的, 那么算法直接知道 $CST(k)$ 对于结点 $v$ 是没有解的。本节会估计 $m^*(G, v)$ 的上界。如果 $k$ 是大于这个上界的, 那么算法直接知道是无解的。这个上界的计算在定理3.24中。

**定理 3.24.** 给定连通简单图 $G(V, E)$ , 对于任意 $v \in G$ , 有

$$m^*(G, v) \leq \left\lfloor \frac{1 + \sqrt{9 + 8(|E| - |V|)}}{2} \right\rfloor \quad (3.3)$$



**Algorithm 2:** 一个CST的基本框架

输入:  $G(V, E), v_0, k$   
 输出: CST( $k$ )的解  
 1: **if**  $k > \text{upperBound}(G)$  **then**  
 2:     **return**  
 3: **end if**  
 4:  $C \leftarrow \text{candidateGeneration}(G, v_0, k);$   
 5: **if** 没有找到解 **then**  
 6:     在 $G[C]$ 的 $k$ 核中使用全局搜索;  
 7: **end if**  
 8: **return**

**证明.** 由于 $G$ 是连通图, 有 $|E| \geq |V| - 1$ 。为了简便起见, 本文使用 $H^*$ 和 $m^*$ 来表示这个最优社团及其最小度。那么 $G[H^*]$ 的边数至少为 $\lceil m^*|H^*|/2 \rceil$ 。对于其他的结点 $V - H^*$ , 至少有 $|V| - |H^*|$ 条边来确定图 $G$ 的连通性。因此有:

$$\lceil \frac{m^*|H^*|}{2} \rceil + |V| - |H^*| \leq |E| \quad (3.4)$$

同时容易发现,  $|H^*| \geq m^* + 1$ 。因此, 根据公式3.4, 有 $(\frac{m^*}{2} - 1)(m^* + 1) \leq |E| - |V|$ 。通过简单的计算, 可以得到 $(\frac{m^*}{2} - 1)(m^* + 1) \leq |E| - |V|$ 。对以上公式求解, 可以得到:

$$m^*(G, v) \leq \lfloor \frac{1 + \sqrt{9 + 8(|E| - |V|)}}{2} \rfloor \quad (3.5)$$

□

**5.2.2. 一个朴素的候选答案生成方法**

本节将展示一个朴素的 $\text{candidateGeneration}()$ 的实现方式。论文会在第5.3节给出一个更复杂而高效的解法。

例3.27中给出了这个伪代码的一个示例。在这个朴素策略中, 算法通过从结点 $v_0$ 开始的BFS来生成候选结点。就像在算法3中显示的那样, 算法在搜索过程中剪枝掉所有度小于 $k$ 的结点。算法3的运行时间复杂度为 $\Theta(n' + m')$ , 这里 $n'$ 是 $G[C]$ 的结点的个数,  $m'$ 是 $G[C]$ 的边的个数。

**5.2.3. 复杂度分析**

接下来研究当使用算法3中的对于 $\text{candidateGeneration}()$ 的实现时, 算法2的时间复杂度。最后一步的全局搜索的复杂度, 和候选答案生成的复杂度一致。算法2的复

**Algorithm 3:** 朴素 candidateGeneration()**输入:**  $G(V, E), v_0, k$ **输出:**  $C$ 

```

1:  $v_0$ 入队列;  $C \leftarrow \emptyset$ ;
2: while 队列非空 do
3:    $v$ 出队列;
4:    $C \leftarrow C \cup \{v\}$ ;
5:   if  $\delta(G[C]) \geq k$  then
6:     找到一个有效解, 返回 $C$ ;
7:   end if
8:   for each  $(v, w) \in E$  do
9:     if  $w$ 没有被访问过且 $\deg_G(w) \geq k$  then
10:       $w$ 入队列;
11:    end if
12:  end for
13: end while
14: return  $C$ 

```

杂度为 $\Theta(n' + m')$ 。因此, 为了降低复杂度, 需要降低 $n' = |C|$ 以及 $m'$ 。第5.3节提出了一个优化方法来降低 $n'$ 以及 $m'$ 。在这之前, 本节首先对 $n'$ 以及 $m'$ 做深度分析, 以评估朴素候选答案生成的剪枝力度。

**$n'$ 的大小估计** 首先,  $n'$ 的值有一个明显的上界:  $|V|$ 。在最差情况下, 有 $C = V$ 。例如, 当 $k = N - 1$ 时, 图 $G$ 是一个拥有 $N$ 个结点的完全图。这时对于任意一个查询点, 只有整个图才是最优的社团。因此在最差情况下, 局部搜索相比全局搜索不会有任何的优势。

其次,  $n'$ 有一个更为紧凑的上界:  $|V_{\geq k}|$ 。假设 $V_{\geq k} = \{v | \deg_G(v) \geq k\}$ 表示图 $G$ 中所有度不小于 $k$ 的结点。很明显,  $G[C]$ 属于 $G[V_{\geq k}]$ 的一个连通分支。例如, 如图3.1所示的 $G$ , 查询点为 $g$ 。对于 $k = 4$ , 有 $C = \{g, h, i, j, k, l\}$ , 它的导出子图是一个有 $V_{\geq 4} = \{d, e, g, h, i, j, k, l\}$ 得到的导出子图的连通分支。

图3.3展示了对于 $|C|$ 的上界的模拟计算, 以及根据朴素候选答案生成方法找到的 $|C|$ 的真实值。实验生成了不同尺寸、同样参数的模拟图。模拟图随机选择了10个结点作为查询结点, 并记录了它的平均度量。图中同样给出了在第5.3.1节提出的通过提升的局部搜索策略得到的社团大小(表示为“local search”)。这个模拟值表示 $|C|$ 的值和真实网络的社团大小是非常接近于 $|V_{\geq k}|$ 的, 但是与 $|V|$ 的值差距很大。这表明了 $|V_{\geq k}|$ 是

一个对于  $C$  的上界的良好估计。 $m'$  的预测使用  $|V_{\geq k}|$  作为  $|C|$  的预测值。

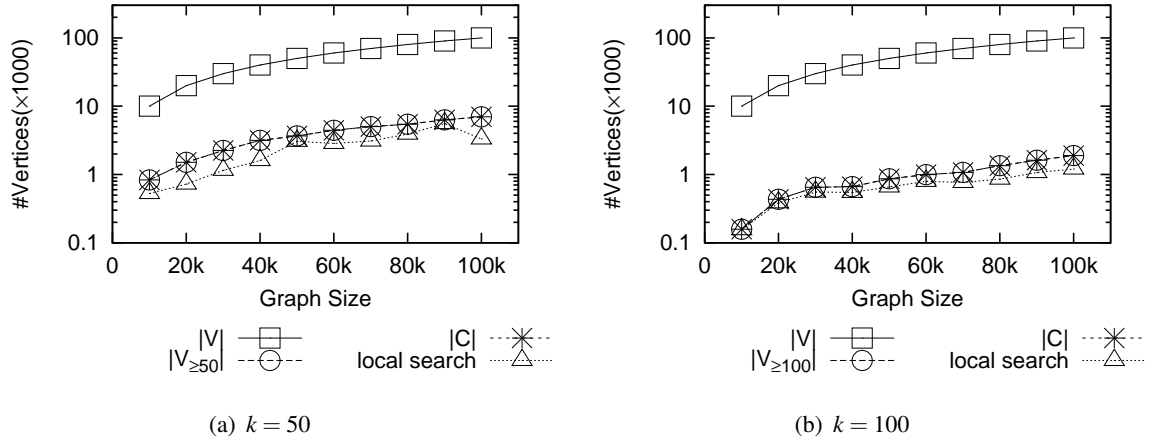


图 3.3:  $|C|$  的上界的模拟

**$m'$  的值预测** 很明显,  $G[V_{\geq k}]$  中边的个数是  $m'$  的上界。度分布是一个真实网络的重要特性, 所以本文会基于度分布来预测  $G[V_{\geq k}]$  中边的个数。

假设  $p_k$  表示从图中随机选择一个结点, 其度数为  $k$  的概率。假设  $P = \{p_0, p_1, p_2, \dots, p_\omega\}$  表示图  $G$  的度分布, 这里  $\omega$  是图  $G$  的结点的最大度, 同时  $\sum_{0 \leq i \leq \omega} p_i = 1$ 。一般来说,  $\omega \in o(|V|)$  的假设对于真实图是合理的。给定以上假设, 其主要结论是: 给定图  $G$  和其结点度分布  $P = \{p_0, p_1, p_2, \dots, p_\omega\}$ , 以及最大度  $\omega \in o(|V|)$ ,  $m'$  可以被预测为:

$$|V_{\geq k}| \sum_{t=1}^{\omega} t \times q_t = n \sum_{i=k}^{\omega} p_i \sum_{t=1}^{\omega} t \times q_t \quad (3.6)$$

这里  $q_t$  被公式 3.7 定义。这个公式基于定理 3.25 和引理 3.26。定理 3.25 给出了  $G[V_{\geq k}]$  的度分布 (在渐进意义下, 即图的结点个数被视为足够大)。引理 3.26 给出了  $G[V_{\geq k}]$  中度数最大的点的估计。为了节约空间, 本文隐去了这两个证明。

**定理 3.25.** 给定图  $G$ , 其度分布  $P = \{p_0, p_1, p_2, \dots, p_\omega\}$ , 和最大度  $\omega \in o(|V|)$ 。假设  $q_t$  表示从  $G[V_{\geq k}]$  中随机选择一个点, 其度分布为  $t$  的概率。则:

$$q_t = \sum_{i=t}^{\omega} p_i \binom{i}{t} p^t (1-p)^{(i-t)} \quad (3.7)$$

这里  $p = \frac{\zeta(k)}{\zeta(0)}$ ,  $\zeta(x) = \sum_{i=x}^{\omega} i \times p_i$ .

**引理 3.26.** 对于一个图  $G$  及其最大度  $\omega$ , 度分布  $P = \{p_0, p_1, p_2, \dots, p_\omega\}$ 。在渐进意义下,  $G[V_{\geq k}]$  的最大度依然是  $\omega$  几乎一定是对的, 当  $n \rightarrow \infty$  时。这里  $n$  是图  $G$  的结点个数。

### 5.3. 优化算法

本节介绍两种用来优化算法并降低 $n'$ 和 $m'$ 的技术。

#### 5.3.1. 智能候选点选择

算法3的朴素在于，它是盲目的从当前点集的邻居中选点的。如下的例子展示了它是如何因为错误的选择，从而导致更多的枚举迭代步数的。

Step1: $C=\{e\}$	$Q=\{a,c,d,f\}$	Step1: $C=\{e\}$	$Q=\{a,c,d,f\}$
Step2: $C=\{e,a\}$	$Q=\{b,c,d,f\}$	Step2: $C=\{e,a\}$	$Q=\{b,c,d,f\}$
Step3: $C=\{e,a,f\}$	$Q=\{b,c,d,g,n\}$	Step3: $C=\{e,a,d\}$	$Q=\{b,c,f\}$
...		Step4: $C=\{e,a,d,b\}$	$Q=\{c,f\}$
Step12: $C=V-\{n,m\}$	$Q=\{\}$	Step5: $C=\{e,a,d,b,c\}$	$Q=\{f\}$

(a) 朴素选择方法

(b) 智能选择方法

图 3.4: 朴素&智能候选点生成

**例 3.27** (候选点生成过程). 考虑图3.1所示网络。对于结点 $e$ 和问题 $CST(3)$ ，使用朴素候选点生成方法， $f$ 可能被加入到 $C$ 中（正如图3.4(a)中第三步所示），这导致了一个无效答案。但是算法流程依然持续，直到所有的其它结点都被枚举到（总共需要12步）。另一方面，如果算法总是选择到当前 $C$ 有最多条关联边的点来加入，那么可以在5步内找到一个有效解。两个不同的候选点生成选择的结果被展示在图3.4中，这里 $Q$ 是当前选择的结点队列。

为了减少 $n' = |C|$ ，本文提供了两个候选点选择策略。其基本思想是通过优先队列，优化候选点生成过程，每次选择最有希望导致有效解的结点。理论上，引理3.20可以用来计算一个结点的有效程度。然而，它的计算依然太过耗时。本节专注于更为轻量级的候选点生成策略，期望它具有常数级别复杂度。接下来会具体提出说明两个选点策略是如何实现的。

**最大度量提升(lg)** 很直接的，由于问题要求社团的度量达到或超过某一阈值，即 $\delta(G[C]) \geq k$ 。所以在每一步选择中，算法朝着使该度量上升最大的方向来选择结点。在这种策略下，一个结点 $v$ 的优先度可以被计算为：

$$f(v) = \delta(G[C \cup \{v\}]) - \delta(G[C]) \quad (3.8)$$

这是一个贪心的方法，因为它只考虑了每一步对于  $\delta(C)$  的提升。注意到，每次一个结点被新加入  $C$  时，它的最小度最多提升 1。换句话说，对于任意的结点  $v$ ，都有  $f(v) = 1$  或  $f(v) = 0$ 。因此，这个策略等同于随机选择一个邻接点，这个点和当前  $C$  中的最小度的点相连。

**最大邻接数(ii)** 这是一个更为智能的选择策略。每个点  $v$  的优先度被定义为：

$$f(v) = \deg_{G[C \cup \{v\}]}(v) \quad (3.9)$$

在这个策略中，算法选择和当前已选择结点具有最大连接数的结点。这会导致对于  $G[C]$  的平均度的最大提升。一般来说，当一个图的平均度提升的时候，它的最小度也会期望提升。并最终在一个有限的步数内，得到一个满足  $\delta(G[C])$  限制的有效解  $C$ 。

正如例 3.27 中说明的，这些选择策略一般都是非常高效的。然而，例 3.28 依然说明了局部信息有时候是不足以在算法 3 的 WHILE 循环中构建一个有效解的。在这种情况下，算法依然需要使用全局搜索来找到  $G[C]$  的  $k$  核（算法 2 中的第 6 行）。根据命题 3.23，这一步总是可以保证算法找到有效解。

**例 3.28 (选择困难).** 接着例 3.27 中的例子，通过最大邻接数策略，很容易在  $e$  之后选择结点  $f$ 。这依然导致了更多的结点被访问遍历，而且没有解会被找到。

### 5.3.2. 复杂度分析

假设  $n'$  和  $m'$  分别表示图  $G[C]$  中的结点数和边数。整体上，**lg** 和 **li** 策略都可以在  $O(n' + m' \log n')$  时间内实现，因为每一次一个新的结点被加入时，最多  $d$  个结点的优先度被改变（这里  $d$  是该结点的度）。其结果是，最多  $m'$  个更新操作会被执行。每一个队列的相关操作（添加、删除、优先度更新）都具有  $O(\log n')$  的复杂度。

但是通过特殊设计，**li** 可以在  $O(n' + m')$  时间内完成计算，并且每次扩展的复杂度变为  $O(1)$ 。算法维护一个列表集合，每个列表包含具有相同  $f(v)$  的  $V$  中结点。每次一个新的结点被加入到  $C$  中时，它的邻居的  $f(\cdot)$  值（除了已经在  $C$  中的结点）将会提升 1。对于每个受影响的结点  $v$ ，算法将  $v$  从它的原始对应的列表  $f(v)$  中移动到  $f(v) + 1$  对应的列表中。这样，算法总是可以在  $O(1)$  时间内定位到一个  $f(\cdot)$  最大的结点。图 3.5 和例 3.29 说明了这一过程。

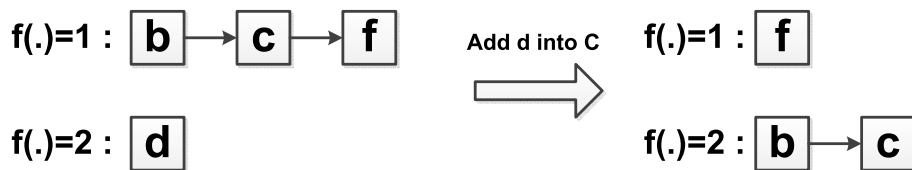


图 3.5: li 策略使用的数据结构的例子。

**例 3.29.** 考虑图3.1中的例子。假设结点 $e$ 和 $a$ 已经加入了 $C$ 中（候选结点集）。那么有 $f(b) = f(c) = f(f) = 1$ ,  $f(d) = 2$ 。根据 $f()$ 函数的计算方式（即所有点根据其邻接数插入对应的列表中），算法可以把 $C$ 的邻居划分到两个列表中，如图3.5左侧所示。算法同时会记录一个指向最大 $f()$ 的指针。这个指针帮助其找到具有最大连接数的候选结点（算法3，第3行）。接着，算法将 $d$ 从它的列表中移除，并加入 $C$ 中（算法3，第4行）。同时算法更新这两个列表，将 $b$ 和 $c$ 从 $f(.) = 1$ 的列表中移动到 $f(.) = 2$ 的列表中。这样，算法就实现了对于该数据结构的更新操作。

### 5.3.3. 智能扩展

算法通过设计扩展策略，对需要访问的邻居做了剪枝，从而降低 $m'$ 。剪枝的基本思想是对结点的邻接表的排序。算法通过邻接表来表示整张图。对于每个结点，将它邻接表中的邻居按照降序排序。接着，在候选结点集生成过程中，当算法扩展到一个结点的邻居时(算法3的9-11行)，如果该结点的度小于 $k$ ，算法立刻停止这个扩展。这样，算法避免了对所有邻居的搜索。

对于一个具有 $d$ 个邻居的结点，算法需要 $O(d \log d)$ 的时间来对它的邻接表作排序。为了避免这样的花费，算法在线查询之前的预处理过程中完成这一行为。在大部分真实的网络查询中，有大量来自用户的查询。因此对于邻接表的预处理是一种提前而有效的方式。当一个图是在动态变化时，它的邻接表维护是更为耗时的。这时可以通过二分搜索树来表示这一邻接结构，并每次使用 $O(\log(d))$ 的时间来维护图的更新。

很明显，这个优化值对局部搜索有效，而对全局搜索是无效的。实验结果表明，这个优化效果明显。

## 第6节 CSM的局部搜索方法

CSM问题的目标是找到对于给定结点的最佳社团。其主要挑战在于，社团的度量 $\delta(\cdot)$ 是非单调的。通过暴力枚举来解决的方法，其复杂度是指数级的。第2.2节引入了通过二分CST来解决CSM的方法。这一节介绍一个更为高效的自底向上的方法。该算法有3步。首先，它从查询点 $v_0$ 开始扩展搜索空间。第二，它在此查询空间中，生成一个候选结点集 $C$ 。第三，它通过最大核算法在候选结点集中来找到最终的答案。

### 6.1. 扩展搜索空间

在这一步（算法4的1-15行中），算法的目标是通过扩展搜索空间，来找到一个子集 $H$ ，使得在线性复杂度下，当剪枝掉尽量多的无效点后，它的 $\delta(G[H])$ 尽可能大。算法从结点 $v_0$ 开始，每一步通过局部搜索策略选择新的点，并加入当前结果中。这里算法使用最大邻接数来作为策略选择局部最优结点（第6-7行）。然后，在每一轮的最后，由



于所有的结点度数小于 $\delta(G[H]) + 1$ 的结点无法存在于更优的解中。所以在第14行，算法用 $H$ 作为过滤依据，来扩展需要访问的结点集合。

**Algorithm 4:** 一个CSM的局部搜索框架

**输入:**  $G(V, E), v_0, -\infty < \gamma < \infty$   
**输出:**  $H$

{第1步: 迭代的搜索和过滤.}

- 1:  $H \leftarrow \emptyset$  /\* 当前最优解 \*/
- 2:  $A \leftarrow \{v_0\}$  /\* 已访问结点 \*/
- 3:  $B \leftarrow \{v | (v, v_0) \in E\}$  /\* 需要访问的结点 \*/;
- 4:  $s \leftarrow 0$
- 5: **while**  $B \neq \emptyset$  and  $s \leq e^{-\gamma}(\lfloor \frac{|E|-|V|}{(\delta(G[H])+1)/2-1} \rfloor - |H|)$  **do**
- 6:   另 $v$ 为 $B$ 中有最多从 $A$ 边相连的结点;
- 7:    $s \leftarrow s + 1; A \leftarrow A \cup \{v\}; B \leftarrow B - \{v\};$
- 8:   **if**  $\delta(A) > \delta(H)$  **then**
- 9:      $H \leftarrow A; s \leftarrow 0;$
- 10:   **if**  $\delta(H) = \min\{deg_G(v_0), \lfloor \frac{1+\sqrt{9+8(|E|-|V|)}}{2} \rfloor\}$  **then**
- 11:     **Return**  $H;$
- 12:   **end if**
- 13: **end if**
- 14:   将 $v$ 的度大于 $\delta(G[H])$ 的邻居加入 $B$ ;
- 15: **end while**

{第2步: 生成候选结点集}

- 16:  $C \leftarrow$  基于 $H$ 或 $A$ 生成候选结点集;

{第3步: 寻找解}

- 17:  $H \leftarrow maxcore(G[C], v_0);$
- 18: **return**  $H;$

一个严重的问题在于，算法何时可以停止这个扩展过程。很明显，当满足以下条件时， $H$ 就是最优解：

$$\delta(H) = \min\{deg_G(v_0), \lfloor \frac{1 + \sqrt{9 + 8(|E| - |V|)}}{2} \rfloor\} \quad (3.10)$$

然而，公式3.10是一个充分而非必要条件。例如，如果一个无效结点在扩展的早期被引入 $H$ 中，那么 $H$ 可能永远也达不到这个上界，即使它可能已经包含了最优解。为了解决这个问题，本文给出了另一个上界证明。假设有一个包含当前 $H$ 的解存在，考虑为了

使 $G[H]$ 提升而被进一步加入 $H$ 中的结点数。其上界在推论 3.31 中。这个推论是从定理 3.30 中推导得到的。

**定理 3.30.** 对于一个连通图 $G(V, E)$ ，并且 $v \in V$ 是一个查询点。如果 $H$ 是 $CST(k)$ 问题的一个解，那么

$$|H| \leq \lfloor \frac{|E| - |V|}{k/2 - 1} \rfloor \quad (3.11)$$

**证明.** 由于 $G$ 是连通的，有 $|E| \geq |V| - 1$ 。同时， $G[H]$ 有至少 $k|H|/2$ 条边。所以存在至少 $|V| - |H|$ 条从 $H$ 连向 $V - H$ 的边，用来保持图的联接性。因此有 $k \cdot |H|/2 + |V| - |H| \leq |E|$ ，它导致 $|H| \leq \frac{|E| - |V|}{k/2 - 1}$ 。□

**推论 3.31.** 对于连通图 $G(V, E)$ ，假设 $H$ 是当前的算法 4 的最优解。如果存在 $H' \supset H$ 使得 $\delta(G[H']) = \delta(G[H]) + 1$ ，算法需要添加至多

$$\lfloor \frac{|E| - |V|}{(\delta(G[H]) + 1)/2 - 1} \rfloor - |H| \quad (3.12)$$

个结点来得到 $H'$ 。

推论 3.31 说明，越大的 $\delta(G[H])$ 会导致越紧的上界。因此这个上界估计对于有较大的 $m^*$ 的解的剪枝效果更佳。

给定如上上界，算法利用两个参数 $s$ 和 $\gamma$ （后者由用户指定）来控制搜索空间。 $s$ 表示当前已经被加入 $H$ 的结点数（见第 7 和第 9 行）。在第 5 行，算法使用：

$$s \leq e^{-\gamma} (\lfloor \frac{|E| - |V|}{(\delta(G[H]) + 1)/2 - 1} \rfloor - |H|) \quad (3.13)$$

以及 $-\infty < \gamma < \infty$ 来控制搜索空间。当搜索空间上界达到时，算法直接停止此搜索。注意到当 $\gamma = 0$ ，公式 3.13 就会退化为推论 3.31。当 $\gamma > 0$ ，被加入到 $H$ 中的结点的个数，会小于这个确定的上界。当 $\gamma \rightarrow -\infty$ ， $s$ 不被此公式所限制。论文会先介绍算法的剩余步骤，之后再回到这些参数的估计问题上。

## 6.2. 候选结点生成

在第 2 步中，算法从第 1 步生成的 $H$ 产生候选结点集合。具体来说，算法使用两个方法来产生 $C$ （算法 4 的第 16 行），并分析了在生成质量和性能上的权衡取舍。假设 $C_{naive}(k)$ 表示算法 3 的结果。也就是说，从 $v_0$ 的邻居中不断移除度数小于 $k$ 的点，就形成了 $C_{naive}(k)$ 。

**解法 1:**  $C \leftarrow A$  在这种情况下，有如下结论：

**定理 3.32.** 给定图 $G$ 和查询点 $v_0$ ，当 $\gamma \rightarrow -\infty$ ，算法 4 一定可以找到 $CSM$  问题的最优解。



**证明.** 考虑算法4中的WHILE循环。假设  $k = \delta(G[H])$ 。很明显有  $m^*(G, v_0) \geq k$ ,  $C_{naive}(k) \subseteq A$ 。于是可以得到  $C_{naive}(m^*(G, v_0)) \subseteq C_{naive}(k)$ 。因此  $C_{naive}(m^*(G, v_0)) \subseteq A$ 。即  $maxcore(G[A], v_0)$  是一个最优解。  $\square$

当  $C \leftarrow A$ , 算法4通过调整  $\gamma$  的方式, 在答案质量和效率之间作取舍。更具体的, 当  $\gamma$  趋近于  $-\infty$  时, 算法具有高质量。当  $\gamma$  趋近于  $\infty$  时, 算法会选择更快的效率。

**解法2:**  $C \leftarrow C_{naive}(k)$ 。这里  $k = \delta(G[H])$ 。

**定理 3.33.** 给定图  $G$  和查询点  $v_0$ , 算法4总能找到一个最优解:  $maxcore(G[C_{naive}(k)], v_0)$ , 这里  $k = \delta(G[H])$ 。

**证明.** 对于图  $G$  和查询点  $v_0$ , 当  $k$  变小,  $C_{naive}(k)$  会变大。因此, 对于任意的  $k$  满足  $k \leq m^*(G, v_0)$ ,  $C_{naive}(k)$  包含了所有的有效结点。因此算法4将会返回一个子集  $H \subseteq V$ , 满足  $k = \delta(G[H])$  不大于  $m^*(G, v_0)$ 。  $maxcore(C_{naive}(k), v_0)$  即是  $CSM(G, v_0)$  的最优解。  $\square$

注意到, 选择不同的  $\gamma$  不会影响答案的质量, 但是会影响程序运行时间。一般来说, 当  $\gamma \rightarrow -\infty$ , 由于算法已经给了足够的次数来尝试找到好的  $H$ , 算法非常有可能在第2步之前找到一个完整解, 或者一个好的部分解。然而, 这会导致很大的运行时间消耗。如果一个完整解或者部分解已经在第1步被找到, 最大核算法过程就只会花费极少的时间。因此, 一个理想的  $\gamma$  会带来最小的运行时间。

最后, 需要强调的是, 在算法4中, 使用不同的策略实现, 其最差情况的时间复杂度都是  $O(|V| + |E|)$ 。然而, 一般来说就像CST问题一样, 在CSM问题中, 可以期望通过局部搜索, 使用很少的结点访问找到一个最优解。

## 第7节 实验

本节使用真实的语义网络来验证方法的有效性。所有的实验都是在一个配置为AMD Athlon<sup>TM</sup> X2 Dual-core QL-6 2GHz CPU, 2G内存的机器上完成的。

### 7.1. 数据集

实验使用了四个真实的图谱网络: DBLP, Berkeley, Youtube和LiveJournal。DBLP (<http://dblp.uni-trier.de/xml/>) 是一个作者协作网络, 其中每个结点代表一个作者, 每条边表示共同作者关系。Berkeley是一个web网络, 其中节点表示来自berkeley.edu和stanford.edu域的页面, 边表示它们之间的超链接。实验忽略链接的方向。YouTube[69]是一个用户到用户链接网络。LiveJournal是一个在线社交网络。对于每个数据集, 实验只考虑网络的最大连接组件。这些图的统计报告在表3.2中, 其中该图的最大核的最小结点度被标记为  $\delta^*(G)$ 。

网络	#Vertex	# Edge	$\delta^*(G)$	Opt.(ms)	k=20	40	60
DBLP	481K	1.72M	114	703	0	0	0
Berkeley	654K	6.58M	202	2328	9	0	0
Youtube	1.1M	3M	52	1359	0	0	0
LiveJournal	4.0M	34.7M	360	2381	0	0	0

表 3.2: 网络基本信息

## 7.2. 案例分析

本文用两个案例分析来验证基于最小度的社团搜索的有效性。第一个是在DBLP数据集。实验使用数据挖掘领域著名的科学家“Jiawei Han”作为查询结点。设置 $k=5$ 后，实验使用 $ls-li$ 获得了如图3.6(a)所示的社区。实验发现这6位作者都是数据挖掘社区的领先科学家，他们的合作非常频繁。例如，Jiawei Han和Jian Pei共同撰写了超过37篇论文，与Haixun Wang和Philip S. Yu分别合作撰写了46篇和15篇论文。

第二个案例分析来自WordNet。WordNet是一个语义网络，其中每个结点表示特定的概念，每条边表示概念之间存在某种语义关系。实验使用单词“pot”作为 $v_0$ 。使用 $LCS(3)$ 以及 $li$ 策略，并得到社团如图3.6(b)。实验发现该社区的点在语义上与查询点高度相关：它们都是关于容器。*pot*, *bowl*, *dish*是一些容器实体。*vessel*和*container*是这些实体的两个上位词，*containerful*是一个与容器相关的形容词。

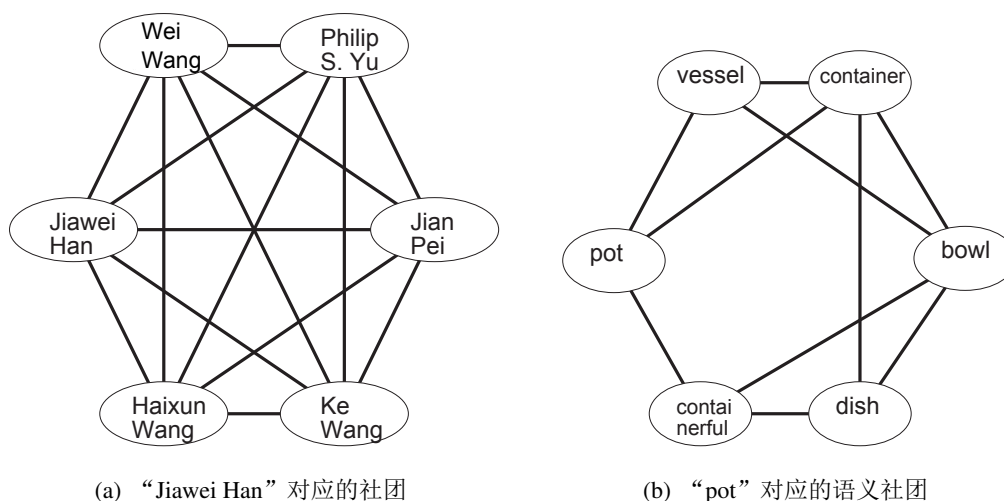


图 3.6: 局部搜索的有效性.

## 7.3. CST的结果

实验通过将CST的局部搜索结果与第4节中介绍的全局搜索方法进行比较，来评估局部搜索的效率。实验将全局搜索表示为 $global$ 。局部搜索的三个版本，即使用朴素候选生成、 $li$ 和 $lg$ 的策略，分别被表示为 $ls-naive$ ,  $ls-li$ 和 $ls-lg$ 。

CST对输入参数 $k$ 敏感。为了评估不同参数下的性能，实验使用 $k = s, 2s, 3s, \dots, 8s$ 测试每个方法，其中 $s = \delta^*(G)/10$ 。对于每个 $k$ ，实验从图的 $k$ 核中随机选择100个结点作为查询结点，并保证查询结点存在有意义的解。然后统计对这100个的查询效率。

**基准方法** 实验首先表明，基准方法在大图上开销巨大。对于每个真实网络，实验选择 $k = 20, 40, 60$ 。对于每个 $k$ ，随机选择100个度不小于 $k$ 的结点作为查询结点。在表3.2的最后三列中记录了在1分钟内可以返回查询结果的结点数量。结果表明，在大多数情况下，基准解决方案不能在给定时间内产生结果。因此，在以下评估中，将省略采用基准方法的评估对比。

**离线排序** 此实验用来验证离线的邻接表排序对于局部搜索的效率优化。本节只给出在DBLP数据上的结果，在其他网络上可以获得类似的效果。在图3.7，实验比较局部搜索加离线排序优化（opt）与不加优化（non-opt）的结果。可以看到，对于大多数 $k$ 下的ls-li和ls-lg方法，离线排序带来了明显的加速。如表3.2中所报告的，对于DBLP数据集，离线排序仅花费703ms。因此，在所有以下实验中，所有本地搜索解决方案都加入此优化方式。

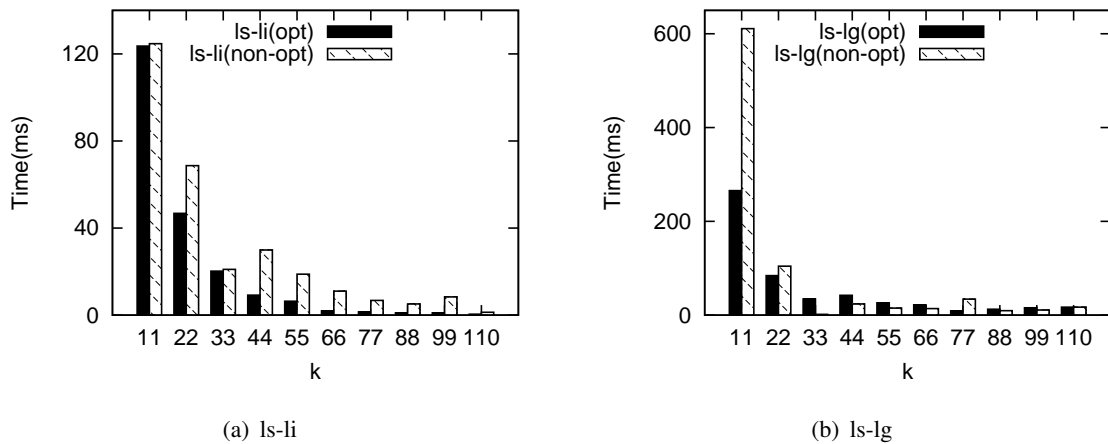


图 3.7: 离线排序的优化力度。

**CST的评估** 图3.8中显示了CST的局部搜索解决方案的性能（平均运行时间以及标准差）。可以看到，在大多数情况下，局部搜索的性能优于全局搜索。当 $k$ 增加时，局部搜索对全局搜索的优势变得更加明显。在最好的情况下，例如当 $k$ 在DBLP和Berkeley上足够大时，ls-li或ls-lg比全局搜索快两个数量级。只有当 $k$ 很小时，全局搜索与局部搜索性能近似。原因是当 $k$ 很小时，图中的大多数结点倾向于参与到答案社团中，这使得局部搜索无法发挥效力。还可以看到，在本地搜索的所有策略中，大多数情况下，ls-li具有最好的效果。

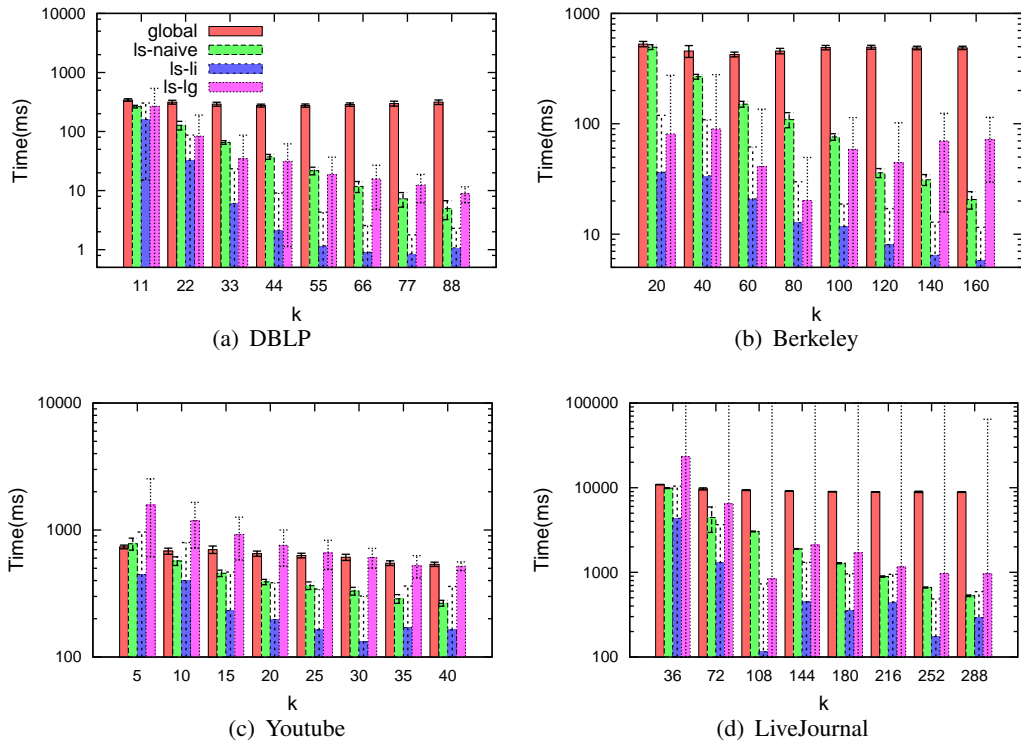


图 3.8: 不同CST方法的效率。

**局部搜索的合理性** 为了验证局部搜索的合理性，本文报告了答案大小和在局部搜索中访问的结点的数量。图3.9显示DBLP的结果（其它真实网络上的结果是相似的）。可以看到，局部搜索方法往往产生一个小的社团。在某些情况下，由ls-li或ls-lg发现的社区的数量级小于全局搜索和ls-naive找到社团的大小。局部搜索比全局搜索访问的节点数要小很多。在许多情况下，局部搜索性能甚至优于全局搜索两个数量级。较小的答案大小和较少额外数量的访问结点解释了局部搜索相对于全局搜索的优势。

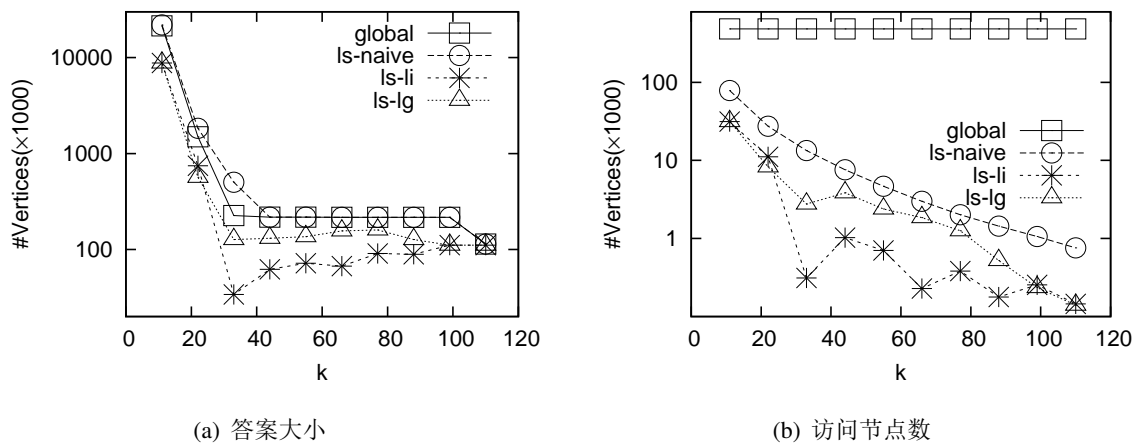


图 3.9: 局部搜索的答案大小和访问节点数。

#### 7.4. CSM的结果

前文设计了两个CSM的局部搜索解决方案。由 $C \leftarrow A$ 生成 $C$ 的方法被表示为CSM1，另一个方法被表示为CSM2。实验将它们与全局搜索global进行比较。在局部搜索中，以前的结果已经显示了li的效率 and 有效性，因此以下实验仅比较它的结果。

**CSM的评估** 对于CSM1，算法可以通过调整参数 $\gamma$ 来平衡性能的解决方案的取舍。为了公平起见，设置 $\gamma \rightarrow -\infty$ ，这样CSM1不受大小 $s$ 约束。结果如图3.10所示。可以看到CSM2表现最好。CSM1消耗了最多的时间，因为实验删除了 $s$ 的大小约束，使得局部搜索具有很大的搜索空间。在下一个实验中，实验将展示通过调整 $\gamma$ ，CSM1可以在不牺牲质量的前提下，实现比全局搜索更快的速度。

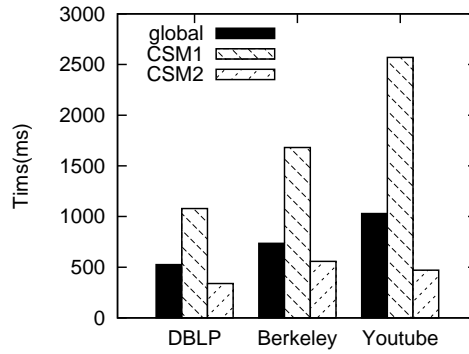


图 3.10: 局部搜索在CSM上的表现

**$\gamma$ 对CSM1的影响**  $\gamma$ 控制CSM1的质量和性能之间的平衡。算法设置 $\gamma$ 从1到15，观察质量和运行时间的变化情况。CSM1的质量通过以下公式计算：

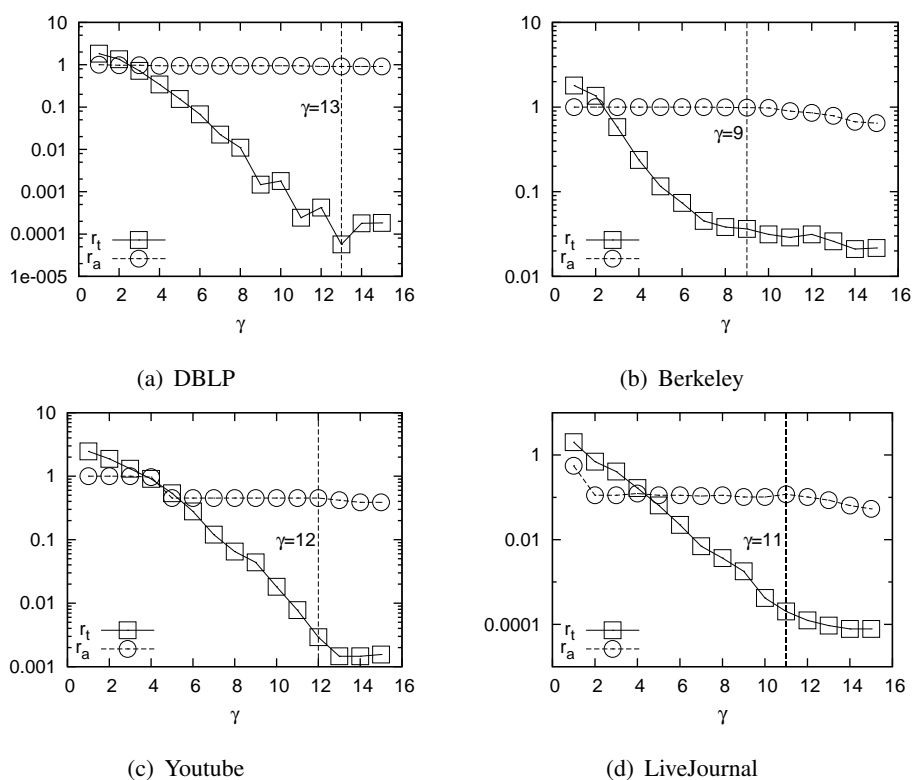
$$r_a = \Sigma_{v_0} \delta(H') / \Sigma_{v_0} \delta(H) \quad (3.14)$$

其中 $H'$ 是CSM1找到的答案， $H$ 是全局搜索找到的最佳答案。测评中还将CSM1对于全局搜索的时间比表示为

$$r_t = \Sigma_{v_0} t_1(v_0) / \Sigma_{v_0} t_2(v_0) \quad (3.15)$$

其中 $t_1(v_0)$ 是CSM1对于查询结点 $v_0$ 的运行时间， $t_2(v_0)$ 是全局搜索的运行时间。显然，时间比 $r_t$ 越低表示算法效率越高，而质量 $r_a$ 越高则表示算法越准确。

质量和时间比之间的变化如图3.11所示。它们都随着 $\gamma$ 的增长而下降。但是，时间比的降低比质量下降快得多。CSM1的质量仍然保持在一定高度。可以观察到 $r_t$ 变化的临界点（由虚线标识），在此之前，较小的质量牺牲使算法性能显著提高。上述结果清楚地表明，以CSM1对于质量和效率的平衡是有效的。在实际应用中，用户可以根据其对性能和质量的要求来指定 $\gamma$ 。

图 3.11:  $\gamma$  对于 CSM1 的影响

## 第 8 节 小结

本章研究了在语义网络或知识图谱中找到包含某一查询点的语义社团的问题，并提出了局部搜索的方法。局部搜索比全局搜索更高效，它避免了访问网络中的所有结点。本章研究并解决了局部搜索中的度量的非单调性问题。经过充分的理论支持，提出了用来解决 CST 问题和 CSM 问题的局部搜索算法。并使用不同的在真实网络上的实验来证明算法效率。

## 第四章 基于知识图谱的短文本动词理解

动词理解在自然语言的语义理解中至关重要。对于动词的理解，有助于问答系统在短文本层面理解问题语义，进而帮助实现上下文相关的实体概念化。传统的动词表示，如FrameNet、PropBank、VerbNet等，专注于动词的角色。但是动词角色过于粗粒度，无法表示和区分动词的语义。本章将介绍基于动词模板的动词语义表示模式。每个动词模板对应于动词的单个语义。首先本文分析了动词模板的原则：一般性和特殊性原则。然后提出一个非监督的最小描述长度模型来建模此问题。实验结果证明了模型的有效性。论文进一步将动词模板应用在上下文相关的实体概念化问题上。

### 第1节 引言

动词对句子的理解是至关重要的[30, 104]。动词理解的一个主要问题是歧义性[77]，即当同一动词和不同宾语相结合时，会表示不同的语义。本文只考虑和宾语相结合的动词，即及物动词。就像例4.1中展示的那样，大部分动词都是具有歧义的。因此需要一个好的动词语义表示方式，来表示其奇异性。

**例 4.1.** eat具有以下几种意思：

- 把食物在嘴里咀嚼并吞咽，例如eat apple, eat hot dog。
- 吃一顿饭，例如eat breakfast, eat lunch, eat dinner。
- 其他俗语。例如eat humble pie表示承认错误。

许多典型的动词表示方式，包括FrameNet [7]，PropBank [52]和VerbNet [82]，侧重描述动词的语义角色，（例如“eat”的摄食者和被进食物）。但是，语义角色是一种非常粗糙的表示，无法区分动词细粒度的语义。不同短语中的同一动词，可以在具有同样语义角色的同时表达不同的语义。在例4.1中，无论是“eat apple”还是“eat breakfast”，都具有摄食者等角色。但是这些eat具有有不同的语义。

由于无法表达动词的歧义性，传统动词表示形式方式无法完整的表示动词语义。在句子“I like eating pitaya”，人直接可以推断“pitaya”大概是一种食物，因为食物是最常见的“eat”的对应语义。这就可以让人利用上下文相关的概念化技术将“pitaya”理解为食物。而传统动词表示形式只理解“pitaya”是一个被进食物，而无法知道它是一顿饭还是一种食物。



**动词模版**本文认为，动词模版可以用来表示一个动词更细粒度的语义。动词模版基于语言学家对单词结合的两个原则设计[84]：俗语原则(idiom principle)和开放选择原则(open-choice principle)。基于这两个原则，本文设计了两种类型的动词模版：

- **概念化模版**：根据开放选择原则，一个动词可以和任何宾语搭配。这些宾语具有特定的概念，可以用于语义表示和消歧[103]。这促使本章使用宾语的概念来表示动词语义。在例4.1中，eat breakfast和eat lunch 具有相似的语义，因为这些宾语对应同样的概念meal。因此，本章将动词短语中的宾语替换为它的概念，从而组成概念化动词模板动词  $\$_C$  概念（例如 eat  $\$_C$ food）。根据宾语的概念，每个开放选择原则中的动词短语会被关联到一个概念化模板中。
- **俗语模版**：根据俗语原则，有些动词短语的具体语义是和宾语概念无关的。本章在宾语前增加 $\$_I$ 标记来表示此俗语模板（即动词  $\$_I$  宾语）。

根据上述定义，本文使用动词模板来表示动词的不同的语义。分配到相同的动词模板的动词短语，其动词具有相似的语义，而分配到不同动词模板的动词具有不同的语义。通过动词模版，可以将“eat pitaya”对应到模板“eat  $\$_C$ food”，从而得到“I like eating pitaya”中的“pitaya”是一种食品。另一方面，俗语模版表示哪些短语不应被概念化。表4.1中列出了例4.1中的短语的对应模板。

动词短语	动词短语	类别
eat apple	eat $\$_C$ food	概念化模板
eat hot dog	eat $\$_C$ food	概念化模板
eat breakfast	eat $\$_C$ meal	概念化模板
eat lunch	eat $\$_C$ meal	概念化模板
eat dinner	eat $\$_C$ meal	概念化模板
eat humble pie	eat $\$_I$ humble pie	俗语模板

表 4.1: 动词短语及其动词模板

这样，本文的核心问题在于如何对动词产生概念化模板或俗语模板。为了做到这一点，实验使用了两个公开数据集：Google Syntactic N-Grams和Probase[103]。Google Syntactic N-Grams包含了百万级别的动词短语，提供了动词短语的丰富语料。Probase包含了丰富的实体-概念信息，帮助算法将宾语映射到其概念。这样问题就变为给定动词 $v$ 以及其对应的动词短语集合，对每个动词短语产生一个动词模板（概念化模板或者俗语模板）。然而，这个模板生成的过程是具有挑战性的。一般来说，在此过程中的最严重问题在于一般性和特殊性之间的取舍。本文在下边具体说明。



## 第2节 相关工作

**传统的动词理解** 在此讨论比较动词模板与传统的动词理解[73]方法。FrameNet[7]建立在大多数词汇可以被很好的通过语义框架[32]理解的思想。语义框架是对于一类事件，关系，或者在其中实体与参与者的描述。每一个语义框架使用了框架元素(FEs)来进行简单的注解。PropBank [52]使用了人工标注的谓词和语义对象变量，来获取准确的谓词-变量结构。谓词这里指动词，变量是动词的其他对象。为了使 PropBank 更加形式化，变量通常包含施动者，受动者，工具，起始点和终止点等。VerbNet[82]将动词根据它们基于Levin classes[60]的句法模板来分类。所有的动词理解都关注于动词的不同角色而不是它的语义。然而动词的不同语义可能有相同的角色，现有的理解并不能完整的表述动词的语义。

**实体概念化** 本章工作的典型应用就是基于上下文的实体概念化理解。实体概念化决定了对于一个实体最适合的概念。传统的基于文本检索的方法利用NER[92]来进行实体概念化。但是NER经常只含有一些粗糙的预定义概念。Wu等人建立了一个有大量词汇信息的知识库Probase[103]，来提供丰富的IsA关系。对于IsA关系，基于上下文的实体概念化理解[50]会更有效。Song等人[87]提出了一种利用Naive Bayes的实体概念化方法。Wen等人[46]提出了主流的联合共现网络，IsA网络和概念聚类的模型。

**语义构成** 本章利用动词模板来表示动词短语。然而语义构成的工作致力于将任意的短语表示为向量或树结构。基于向量空间的模型被广泛地被用来表示单个单词的语义。因此，一个直接的表示词汇的语义的方法就是将这些出现在短语中的单词的向量取平均值[105]。但是这个方法却忽略了词之间的句法关系 [57]。Socher等人[85]将句法关系表示为一棵二叉树，并且将它与单词的向量利用recursive neural network同时训练。目前，word2vec[67]显示了它在单词表示方面的优势。Mikolov等人[68]进行了更深层次的研究并使word2vec可以表示短语向量。总之，这些工作并没有使用有关动词的俗语短语和动词的对象的概念来表示动词的语义。

### 2.1. 一般性和特殊性的取舍问题

本节试着问答问题：“什么样的动词模板可以很好的总结一个动词短语集合”。由于每个动词短语都有若干候选动词模板，这个问题的回答是很困难的。直觉上，一个好的动词模板需要兼备一般性和特殊性。

**一般性：**本章希望用较少的模板个数来覆盖一个动词的所有语义。否则，抽取出的动词模板会变得琐碎。考虑极端的情况：所有的动词短语都被考虑为俗语模板。这些俗语模板显然大部分都是没意义的，因为大部分动词短语需要被概念化。

**例 4.2.** 在图4.1中，模板(eat  $\$c$ meal)显然比三个模板(eat  $\$l$ breakfast + eat  $\$l$ lunch + eat  $\$l$ dinner)要好。前者提供了一个更一般的模板表示。

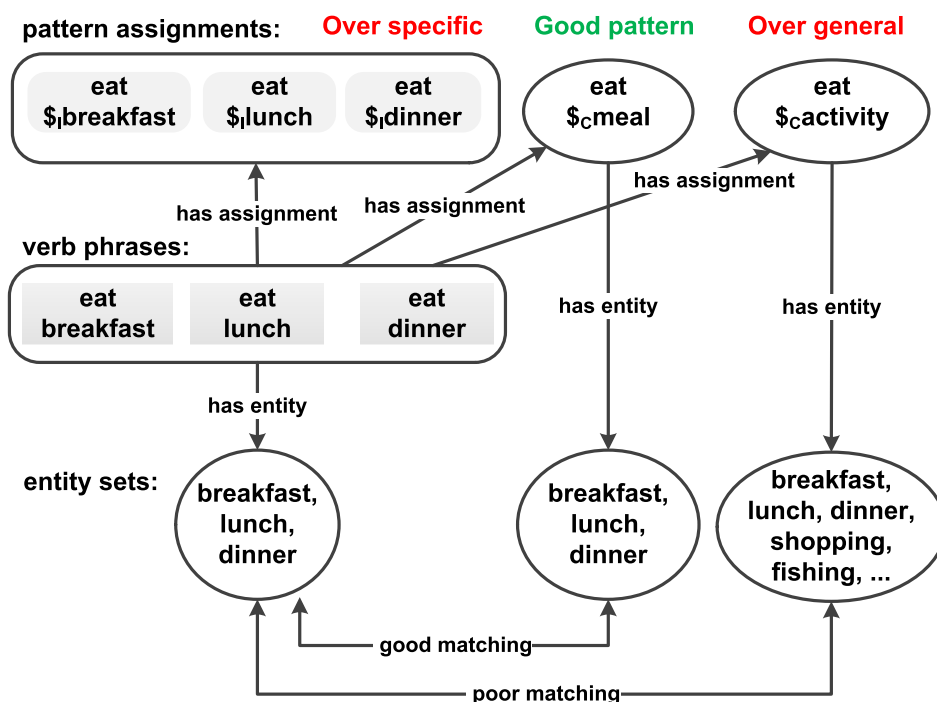


图 4.1: 模板分配的例子

**特殊性:** 另一方面, 本章期望产生的动词模板具备特殊性, 否则结果可能会变得非常模糊。就像例4.3展示的那样, 算法可以将任意宾语都概念化到某些非常高层的概念上, 例如activity, thing, item等。这样概念化的模板就会变得特别模糊而无法精确描述一个动词的语义。

**例 4.3.** 对于图4.1中的动词短语, eat \$c\_activity是比eat \$c\_meal更一般性的动词模板。这样, 一些错误的动词模板, 例如eat shopping或each fishing也会被识别为eat的有效例子或短语。相反, eat \$c\_meal具有更好的特殊性。因为breakfast、lunch、dinner是三个典型的meal的实例。而meal几乎不再具有其它典型实例。

**贡献** 一般性和特殊性显然是相互矛盾和制约的。因此如何在一般性和特殊性之间做取舍构成了本文的主要挑战。本文使用最小描述长度 (minimum description length, MDL) 作为调和这两个目标的基本框架。更具体的, 本章的贡献可以被总结如下:

- 本章提出了动词模板——一种新型的动词语义表现形式。本章提出了两类动词模板: 概念化模板和俗语模板。动词模板可以表示动词的歧义性, 因此可以用来识别动词的不同语义。
- 本章提出了关于动词模板抽取的两个原则: 一般性和特殊性原则。本章阐述了这两个原则间的互相制约, 并提出了一个基于最小描述长度的无监督模型来产生高质量动词模板。

- 本章进行了多样的实验。其结果证实了模型和算法的有效性。

### 第3节 问题模型

本节形式化定义从动词短语中提取动词模板的问题。此模板提取过程中要计算两个值：（1）每个动词短语的动词模板分配；（2）每个动词的动词模板分布。接下来，本文将首先给一些基本的定义。接着提出了一种基于最小描述长度的问题模型，并证明了该模型的合理性。请注意，不同动词的模板是独立的，在问题和算法描述中可以单独考虑每一个动词。因此在以下的说明中，将只讨论针对某一给定动词的解法。

#### 3.1. 初步定义

首先给出动词短语、动词模板、模板分配的标准定义。

**定义 4.4 (动词短语).** 一个动词短语 $p$ 在自然语言中是一个动词及其对应宾语。本文把短语 $p$ 中的宾语表示为 $o_p$ 。

**定义 4.5 (动词模板).** 动词模板是一个或若干个动词短语的总结。每个动词短语只有一个动词模板对应。对应于同一动词模板的动词短语，其动词语义是相似的。本文用 $a$ 表示动词模板。考虑两种动词模板：

- 俗语模板是“*verb \$object*”的形式。只有动词短语“*verb object*”可以对应模板“*verb \$object*”。
- 概念化模板是“*verb \$concept*”形式。动词短语“*verb object*”可以对应于“*verb \$concept*”，仅当 $object$ 和 $concept$ 具有 $isA$ 关系。将概念化模板 $a$ 中的概念表示为 $c_a$ 。

**定义 4.6 (模板分配).** 模板分配是一个函数 $f: P \rightarrow A$ ，它将任意一个动词短语映射到其对应动词模板模板。 $f(p) = a$ 表示 $p$ 的模板是 $a$ 。注意每个模板可以有任意数量的对应的动词短语。

表4.1中展示了一些动词短语、动词模板和模板分配的例子。

短语的分布是已知的（在本章的实验中，其动词短语分布是从Google Syntactic Ngram数据库中抽取的）。所以本文中模板抽取的目标倾在于找到 $f$ 函数。有了 $f$ 函数，就可以很容易的计算模板分布 $P(A)$ ：

$$P(a) = \sum_p P(a|p)P(p) = \sum_{p \text{ s.t. } f(p)=a} P(p) \quad (4.1)$$

这里 $P(p)$ 是给定动词的短语分布。注意这里的第二个等式是成立的，因为当 $f(p) = a$ 时， $P(a|p) = 1$ 。 $P(p)$ 可以直接由 $p$ 的频率得到，见公式4.14。

### 3.2. 模型

这一小节提出了一种基于最小描述长度的模型，它可以精确地建模模板分配中的一般性和特殊性原则。使用最小描述长度的出发点：最小描述长度（minimum description length, MDL）[9]是基于数据压缩程度的数据复杂度描述方法。而在动词模板分配问题中，一个动词模板可以被视为一组动词短语的压缩。对于概念化的模板，直觉上来说，如果一个模板分配是一个对于动词短语的简短描述，那么这个分配方案就抓住了底层的动词语义特征。

给定动词短语集合，寻找一个模板分配函数 $f$ ，使得这些动词短语的描述长度最短。假设 $L(f)$ 表示 $f$ 的描述长度，那么可以将动词短语模板分配问题形式化表示为：

**问题定义 4.7** (模板分配). 给定动词短语分布 $P(p)$ ，找到模板分配 $f$ ，使得 $L(f)$ 最小化：

$$\arg \min_f L(f) \quad (4.2)$$

对于每个短语 $p$ ，它的编码方式包含两部分：左侧部分编码它的对应模板 $f(p)$ （表示为 $l(p, f)$ ），右侧部分编码在给定模板时的动词短语（表示为 $r(p, f)$ ）。这样可以得到：

$$L(f) = \sum_p P(p) L(p) = \sum_p P(p) [l(p, f) + r(p, f)] \quad (4.3)$$

这里 $L(p)$ 表示 $p$ 的整体描述长度，包括左侧编码长度和右侧编码长度。

$l(p, f)$ ：模板编码长度 为了编码 $p$ 的模板 $f(p)$ ，需要的编码长度为：

$$l(p, f) = -\log P(f(p)) \quad (4.4)$$

这里 $P(f(p))$ 可以被公式4.1计算得到。

$r(p, f)$ ：给定模板的短语编码长度 在得到其模板 $f(p)$ 之后，使用从模板 $f(p)$ 到动词短语 $p$ 的转移概率 $P_{\mathcal{T}}(p|f(p))$ 来编码 $p$ 。 $P_{\mathcal{T}}(p|f(p))$ 是通过Probbase[103]计算得来的，并在本文计算中视为先验概率。因此，对 $p$ 的编码需要的编码长度是 $-\log P_{\mathcal{T}}(p|f(p))$ 。为了计算 $P_{\mathcal{T}}(p|f(p))$ ，考虑两种情况：

- 情况一：  $f(p)$ 是一个俗语模板。这样由于俗语模板只有一个对应的动词短语，有 $P_{\mathcal{T}}(p|f(p)) = 1$ 。
- 情况二：  $f(p)$ 是一个概念化模板。在这种情况下，只需要编码给定概念的动词宾语 $o_p$ 。使用从概念 $c_{f(p)}$ 到实体 $o_p$ 的转移概率 $P_{\mathcal{T}}(o_p|c_{f(p)})$ （通过Probbase得到）。实验部分会给出关于此概率的更明确的计算方法。

这样得到：

$$\begin{aligned} r(p, f) &= -\log P_{\mathcal{T}}(p|f(p)) \\ &= \begin{cases} -\log P(o_p|c_{f(p)}) & f(p) \text{ 是概念化模板} \\ 0 & f(p) \text{ 是俗语模板} \end{cases} \end{aligned} \quad (4.5)$$

**总长度** 通过将所有动词短语的描述长度相加，得到模板分配 $f$ 下的总描述长度 $L$ ：

$$\begin{aligned} L(f) &= \sum_p [P(p)l(p, f) + \theta P(p)r(p, f)] \\ &= -\sum_p [P(p)\log P(f(p)) + \theta P(p)\log P_{\mathcal{T}}(p|f(p))] \end{aligned} \quad (4.6)$$

请注意这里公式引入了超参数 $\theta$ 来控制 $l(p, f)$ 和 $r(p, f)$ 的相对重要程度。后文将会解释 $\theta$ 是如何具体影响动词模板在一般性和特殊性中的取舍。

**合理性分析** 接下来，本文会通过证明该模型对于动词模板的两个原则（即一般性和特殊性原则）的体现，来说明模型的合理性。为了简单起见，定义 $L_L(f)$ 和 $L_R(f)$ 分别用来表示对于动词模板部分的编码总长度，和给定模板编码具体动词短语的编码长度。具体计算如下

$$L_L(f) = -\sum_p P(p)\log P(f(p)) \quad (4.7)$$

$$L_R(f) = -\sum_p P(p)\log P_{\mathcal{T}}(p|f(p)) \quad (4.8)$$

**一般性** 通过最小化 $L_L(f)$ ，模型可以找到具有一般性的模板。假设 $A$ 表示所有在分配 $f$ 下的模板， $P_a$ 表示 $a \in A$ 对应的动词短语集合，即满足 $f(p) = a$ 的动词短语集合。根据公式 4.1 和公式 4.7，有：

$$L_L(f) = -\sum_{a \in A} \sum_{p \in P_a} P(p)\log P(a) = -\sum_a P(a)\log P(a) \quad (4.9)$$

所以 $L_L(f)$ 即为动词模板的熵（entropy）。最小化熵将使得模型选择并使用较少的动词模板。这体现了模板的一般性原则。

**特殊性** 通过最小化 $L_R(f)$ ，模型可以找到具有特殊性的模板。公式 4.10 的内部实际上是 $P(P|a)$ 和 $P_{\mathcal{T}}(P|a)$ 的交叉熵。因此最小化 $L_R(f)$ 会使得模型找到使 $P(P|a)$ 和 $P_{\mathcal{T}}(P|a)$ 尽量接近的分布。这体现了特殊性原则。

$$\begin{aligned} L_R(f) &= -\sum_{a \in A} \sum_{p \in P_a} P(p)\log P_{\mathcal{T}}(p|a) \\ &= -\sum_{a \in A} P(a) \sum_{p \in P_a} \frac{P(p)}{P(a)} \log P_{\mathcal{T}}(p|a) \\ &= -\sum_{a \in A} P(a) \sum_{p \in P_a} P(p|a) \log P_{\mathcal{T}}(p|a) \end{aligned} \quad (4.10)$$



### 3.3. 算法

本节提出了一种基于模拟退火的算法来解决问题4.7，并展示了如何利用外部知识来优化俗语模板。本章使用模拟退火算法来计算最好的动词模板分配 $f$ 。算法流程如下。首先选取随机的分配作为初始值（初始温度）。然后，算法生成并评估一个新的分配，如果它是一个更好的分配，用这个分配替换掉之前的分配；否则，算法以一定的概率接受这个分配（温度降低）。重复生成以及替换的步骤直到最后的 $\beta$ 轮结果没有出现变化（终止条件）。

**候选分配生成：**显然，候选分配的生成对于算法的效用和效率来说非常关键。接下来首先介绍一个直接生成候选分配的方法。然后对候选动词模板作典型性的改进。

**直接生成方法：**模板分配 $f$ 的最基本的单元是对一个动词短语的模板分配。直接生成方法会随机选择一个动词短语 $p$ 并将它分配给一个随机的模板 $a$ 。产生一个有效的模板。算法需要确保（1） $a$ 是一个 $p$ 的俗语模板，或者（2） $a$ 是一个实体概念化模板并且 $c_a$ 是 $o_p$ 的上位词。然而，因为很难达到最优状态，这个方法效率很低。对于一个动词，假设它有 $n$ 个动词短语并且每一个动词短语都有平均 $k$ 个候选模板。达到最优状态的最小轮数平均是 $\frac{n}{2}$ ，这在大语料库上是不可接受的。

**利用典型性的生成方法** 注意到对于每一个确定的动词词组，一些模板会因为它们更高的典型性而比其他模板更有效。例4.8介绍了这种情况。这使算法倾向于将动词短语分配给具有更高典型性的动词模板。

**例 4.8.** 对于 eat breakfast, eat lunch。eat  $\$_{Cmeal}$ 显然比eat  $\$_{Cactivity}$ 更好。因为对于一个真实的人来说，他更容易想到eat  $\$_{Cmeal}$ 而不是 eat  $\$_{Cactivity}$ 。换句话说eat  $\$_{Cmeal}$ 相比eat  $\$_{Cactivity}$ 具有更高的典型性。

更形式化的，对于一个确定的动词短语 $p$ ，定义 $t(p,a)$ 衡量模板 $a$ 相对于动词短语 $p$ 的典型性。如果 $a$ 是俗语模板， $t(p,a)$ 被设置成一个常数 $\gamma$ 。如果 $a$ 是实体概念化模板，使用 $o_p$ 相对于 $c_a$ 的典型性定义 $t(p,a)$ ，这里 $c_a$ 是模板 $a$ 的概念。特别的有：

$$t(p,a) = \begin{cases} \gamma & a \text{ 是俗语模板} \\ P_{\mathcal{T}}(o_p|c_a)P_{\mathcal{T}}(c_a|o_p) & a \text{ 是概念化模板} \end{cases} \quad (4.11)$$

这里 $P_{\mathcal{T}}(o_p|c_a)$ 和 $P_{\mathcal{T}}(c_a|o_p)$ 可以从Probase [103]通过式4.15推导出。在这个计算中，公式同时考虑从 $c_a$ 到 $o_p$ 的概率，和从 $o_p$ 到 $c_a$ 的概率，以获得它们相互之间的影响。

#### 3.3.1. 流程

现在将介绍解决方案的详细流程：

1. 通过将每一个 $p$ 分配给它的俗语模板来初始化 $f^{(0)}$ 。

2. 随机选择新的动词模板 $a$ 。对于每一个动词短语 $p$ ,

$$f^{(i+1)}(p) = \begin{cases} a & t(p, a) > t(p, f^{(i)}(p)) \\ f^{(i)}(p) & \text{v} \end{cases} \quad (4.12)$$

这里 $f^{(i)}$ 是第 $i$ 轮产生的分配。

3. 以如下概率接受 $f^{(i+1)}$ :

$$p = \begin{cases} 1 & L(f^{(i+1)}) < L(f^{(i)}) \\ e^{(L(f^{(i)}) - L(f^{(i+1)}))/S^A} & L(f^{(i+1)}) \geq L(f^{(i)}) \end{cases} \quad (4.13)$$

这里 $L(f^{(i+1)})$ 是 $f^{(i+1)}$ 的描述长度,  $S$ 是SA算法进行的轮数,  $A$ 是控制冷却速度的常数。

4. 重复步骤2和步骤3, 直到最后 $\beta$ 轮结果不发生变化。

步骤2和步骤3使算法不同于一般的基于SA的解决方法。在步骤2中, 对于每一个随机选择的动词模板 $a$ , 算法计算它的典型性。如果它的典型性大于当前分配的动词模板, 则将这个动词短语分配给动词模板 $a$ 。在步骤3中, 当一个新的分配的描述长度比上一轮的分配更小时, 算法接受这个新的分配。否则, 算法以 $(L(f^{(i)}) - L(f^{(i+1)}))/S^A$ 的概率接受原有的分配。它的合理性是显然的:  $L(f^{(i+1)})$ 相对于 $L(f^{(i)})$ 的偏离越大,  $f^{(i+1)}$ 被接受的概率越小。

**复杂性分析** 假设有 $n$ 动词短语。在每一轮循环中, 算法随机选择一个动词模板, 然后计算它对于所有 $n$ 个动词短语的典型性, 这需要 $O(n)$ 的时间来实现。然后, 算法通过加和所有 $n$ 个动词短语的编码长度来计算 $f^{(i+1)}$ 的描述长度。这个步骤同样需要 $O(n)$ 时间。假设算法在 $S$ 轮后终止, 则整体的复杂度将是 $O(Sn)$ 。

**利用俗语的先验知识** 可以从外部词典中直接找到许多动词的俗语。如果在字典中一个动词短语被认定为俗语, 它应该被直接分配到俗语模板。特别的, 本章工作中首先从线上字典中爬取了2868个动词短语。然后在步骤2中, 当 $p$ 是其中一个俗语短语时, 将它排除在分配更新的过程之外。

## 第4节 实验

### 4.1. 设置

**动词短语数据** 模板分配会使用动词短语的分布 $P(p)$ 。为了计算 $P(p)$ , 实验使用在Google Syntactic N-Grams的“English All”数据集。该数据集包含从Google Books英文

语料库中提取的统计句法ngrams的信息。它包含22,230个不同的动词, 和147,056个动词短语。对于一个固定的动词, 计算动词短语 $p$ 的概率为:

$$P(p) = \frac{n(p)}{\sum_{p_i} n(p_i)} \quad (4.14)$$

这里 $n(p)$ 是 $p$ 在语料库中的出现的次数, 分母部分是对所有动词短语的次数加和。

**IsA 关系** 本章使用Probase来计算给定概念的情况下实体出现的概率 $P_{\mathcal{T}}(e|c)$ , 同时也计算给定实体概念出现的概率 $P_{\mathcal{T}}(c|e)$ :

$$P_{\mathcal{T}}(e|c) = \frac{n(e, c)}{\sum_{e_i} n(e_i, c)} \quad P_{\mathcal{T}}(c|e) = \frac{n(e, c)}{\sum_{c_i} n(e, c_i)} \quad (4.15)$$

这里 $n(e, c)$ 是 $c$ 和 $e$ 同时出现在Probase的频数。

**测试数据** 实验使用两个数据集来验证方法在长文本和短文本上的有效性。短文本数据集包含来自于Twitter [38]的160万个tweets数据。长文本数据集包含来自于Reuters [5]的21,578个新闻文章。

## 4.2. 动词模板的统计信息

现在简要介绍本文提取的动词模板。对于所有的22,230个动词, 实验列举最频繁的100个动词的统计信息。在过滤掉出现次数 $n(p) < 5$ 的噪声动词短语后, 每一个动词平均有171个不同的动词短语和97.2个不同的动词模板。53%的动词短语有实体概念化模板。47%的动词短语有俗语模板。表格4.2列举了5个有代表性的动词与它们出现最频繁的模板。这个案例分析表明 (1) 有关动词模板的定义反映了动词的一词多义性; (2) 大多数算法得到的动词模板是有意义的。

## 4.3. 有效性

为了评估动词模板的效果, 实验使用了两个评测指标: (1) *coverage*, 表示方法可以找到多少对应于自然语言中的动词短语的模板; (2) *precision*, 表示有多少动词短语和它对应的模板正确匹配。实验通过以下的公式来计算这两个指标:

$$coverage = \frac{n\_cover}{n\_all} \quad precision = \frac{n\_correct}{n\_cover} \quad (4.16)$$

这里 $n\_cover$ 是在测试数据中找到的对应有模板的动词短语的数量,  $n\_all$  是动词短语的总数,  $n\_correct$ 是对应的动词模板正确的动词短语的数量。为了评估*precision*, 实验从测试数据中随机选择了100个动词短语并让志愿者去标注被分配模板的正确性。当一个模板太过具体或者太过一般, 实验认为它是一个不正确的动词短语—模板匹配 (见图4.1中的例子)。为了比较算法好坏, 实验同样列出了模板总结的两种基准方法的结果。



动词: <b>feel</b>	#短语: 1355
feel \$ <sub>C</sub> symptom	feel pain (27103), feel chill (4571), ...
feel \$ <sub>C</sub> emotion	feel love (5885), feel fear (5844), ...
动词: <b>eat</b>	#短语: 1258
eat \$ <sub>C</sub> meal	eat dinner (37660), eat lunch (22695), ...
eat \$ <sub>C</sub> food	eat bread (29633), eat meat (29297), ...
动词: <b>beat</b>	#短语: 681
beat \$ <sub>I</sub> retreat	beat a retreat (11003)
beat \$ <sub>C</sub> instrument	beat drum (4480), beat gong (223), ...
动词: <b>ride</b>	#短语: 585
ride \$ <sub>C</sub> vehicle	ride bicycle (4593), ride bike (3862), ...
ride \$ <sub>C</sub> animal	ride horse (18993), ride pony (1238), ...
动词: <b>kick</b>	#短语: 470
kick \$ <sub>I</sub> ass	kick ass (10861)
kick \$ <sub>C</sub> body part	kick leg (703), kick feet (336), ...

表 4.2: 一些提取的动词模板。在括号中的数字是动词短语在 **Google Syntactic N-Gram** 数据中出现的频数。#*phrase* 表示这个动词的不同动词短语的个数。

- **Idiomatic Baseline (IB)** 每一个动词短语是一个俗语。
- **Conceptualized Baseline (CB)** 对于每一个动词短语，它被分配给一个实体概念化模板。对于宾语 $o_p$ ，基准算法选择最高出现概率的概念，即 $\arg \max_c P(c|o_p)$ ，来构建这个模板。

在Tweets和News数据集上，动词模板分别覆盖了64.3%和70%的动词短语。考虑到在Google N-Gram数据中的拼写错误以及解析错误，这样的覆盖率是可以接受的。图4.2展示了本章方法以及基准方法提取的动词模板(VP)的查准率。结果显示本章方法相比于基准方法在精确度方面有很大的提升。结果同时显示了对于动词的语义理解来说实体概念化模板与俗语模板都是必要的。

## 第5节 应用：基于上下文的实体概念化

如同在引言中所提及的，动词模板可以用来优化基于上下文的实体概念化任务（通过考虑实体的上下文来提取一个实体的概念）。本节将动词模板与主流的基于实体的方法 [87]相结合来优化这一问题。

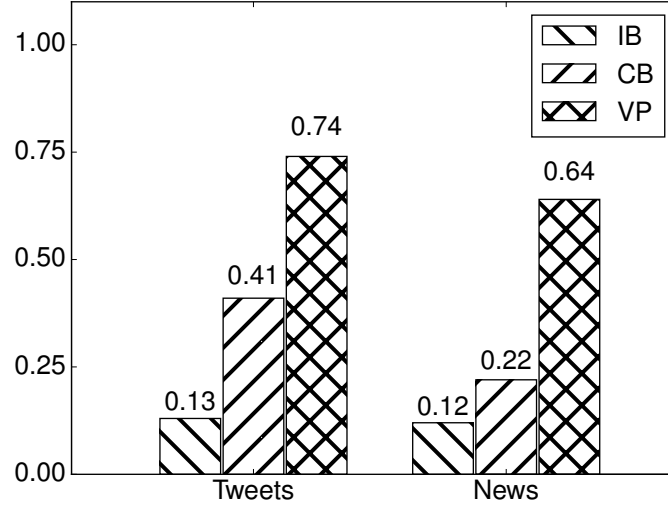


图 4.2: 准确率

**基于实体的方法** 此方法利用在上下文中出现过的实体来概念化一个实体 $e$ 。令 $E$ 是上下文中的所有实体的集合。定义在给定上下文 $E$ 下 $c$ 是实体 $e$ 的概念的概率为 $P(c|e, E)$ 。通过假设所有的实体在给定概念下是独立的，可以得到以下公式来计算 $P(c|e, E)$ ：

$$P(c|e, E) \propto P(e, c) \prod_{e_i \in E} P(e_i | c) \quad (4.17)$$

**本节的方法** 本节在上下文中增加了动词作为附加信息来实体概念化 $e$ 。当 $e$ 是一个动词的对象的时候，利用动词模板可以推导出 $P(c|v)$ ，即在给定动词 $v$ 的动词短语下观察到有关概念 $c$ 的实体概念化模板的概率。因此，在给定上下文 $E$ 和实体 $e$ 还有动词 $v$ 的情况下，概念 $c$ 出现的概率是 $P(c|e, v, E)$ 。类似于等式 4.17， $P(c|e, v, E)$ 可以通过以下公式来计算：

$$\begin{aligned}
 P(c|e, v, E) &= \frac{P(e, v, E | c) P(c)}{P(e, v, E)} \propto P(e, v, E | c) P(c) \\
 &= P(e | c) P(v | c) P(E | c) P(c) \\
 &= P(e | c) P(c | v) P(v) \prod_{e_i \in E} P(e_i | c) \\
 &\propto P(e | c) P(c | v) \prod_{e_i \in E} P(e_i | c)
 \end{aligned} \quad (4.18)$$

注意到如果 $v + e$ 在Google Syntactic N-Grams数据中被观察到，这意味着算法已经学习到了这个模板，可以使用这些模板来进行实体概念化。也就是说，如果 $v + e$ 被映射到了一个实体概念化模板，则使用模板的概念作为实体概念化的结果。如果 $v + e$ 是一个俗语模板，则停止实体概念化。

**设置与结果** 对于在实验部分使用的两个数据集，本实验同时利用上述两个方法来概念化在动词短语中的宾语。然后，选择概率最大的概念作为对象的概念标签。本实验随机选取了两种方法所对应标签不同的100个短语。对于每一个不同，使用人工标注其结果是否好于(*better*)，等于(*equal*)或差于(*worse*)不适用动词模板的方法的结果。结果显示在图片4.3 中。在这两个数据集上，利用了动词模板后，精确度都显著的被提高了。这表明了动词模板对于语义理解任务是有意义的。

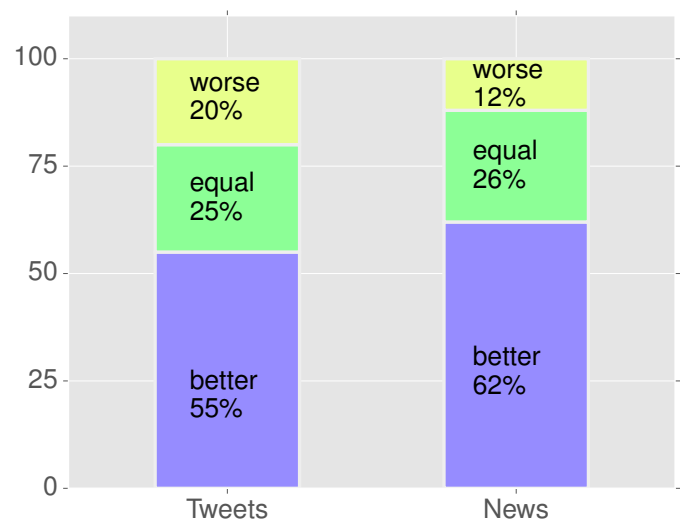


图 4.3: 实体概念化结果

第 6 节 小结

动词的语义对于文本理解来说非常的重要。本章提出了动词模板，用来区分动词的不同语义。论文建立了基于最小描述长度的模型，来平衡动词模板的一般性与特殊性。同时本章提出了一个基于模拟退火的算法来获得动词模板。算法使用了模板的典型性来使候选模板产生的过程收敛速度加快。实验验证了模板的高精确度与覆盖率。本章还展示了动词模板在基于上下文的实体概念化中的成功应用。



## 第五章 从问答语料库和知识图谱学习问答

问答系统 (QA) 已经成为人类访问十亿级知识图谱的流行方式。与网络搜索不同, 在自然语言问题能够被精确地理解和映射到知识图谱上的结构化查询的前提下, 基于知识图谱的问答系统将给出准确且简洁的结果。这其中的挑战是人类可以以许多不同的方式提出同一询问。现有的解决方案由于它们的模型表示而有着天然的缺陷: 基于规则的实现只能理解一小部分的问题, 而基于关键词或同义词的实现不能完全地理解问题。在十亿规模的知识图谱和百万规模的问答语料库的基础上, 本章设计了一种新的问题表现形式: **问题模板**。例如, 对于一个关于某个城市人口数目的问题, 可以学习到诸如 `what is the total number of people in $city?` 或 `how many people are there in $city?` 这样的问题模板。本章共为2782种关系学习了约两千七百万种模板。基于这些模板, 本章设计的问答系统 **KBQA** 能够有效地支持二元事实型问题, 以及由一系列二元事实型问题组合而成的复杂问题。此外, 通过将 **RDF** 知识图谱进行属性扩展, 知识图谱的覆盖范围提高了57倍。在 **QALD** 标准测试集上, **KBQA** 系统在有效性和效率上击败了其他所有竞争对手。

### 第1节 绪论

问答系统 (QA) 已吸引了大量的研究。一个QA系统是被设计用于回答某种特定类型的问题[12]。这其中最重要的一种问题类型是事实型问题 (factoid question, FQ), 这些问题询问有关某个实体的客观事实情况。一种特定的事实型问题是二元事实型问题 (binary factoid question, BFQ) [1], 这些问题询问某个实体的一种属性。例如, `How many people live in Honolulu?` 是一个二元事实型问题。如果系统能回答BFQ, 那么它就有能力去回答其他种类的问题, 比如 1) 排序问题: `Which city has the third largest population?`; 2) 比较问题: `Which city has more population, Honolulu or New Jersey?`; 3) 列举问题: `List the cities ordered by their populations`等。除了BFQ及其变种之外, 系统还能回答像 `When was Barack Obama's wife born?` 这样的复杂的事实型问题。这一问题的回答可以通过合并两个BFQ的回答来实现: `Who is the wife of Barack Obama (Michelle Obama)` 和 `When was Michelle Obama born? (1964)`。系统将复杂事实型问题定义为那些可以分解成一系列BFQ的问题。本章将重点讨论BFQ和如前所述的复杂事实型问题。

基于知识图谱的QA已经有了较长的历史。最近, 大规模知识图谱, 如Google

Knowledge Graph, Freebase[10], YAGO2[45]等, 不断涌现, 极大地增加了问答系统的重要性和商业价值。大部分这样的知识图谱采用了RDF作为数据格式, 并且它们包含数以百万或是十亿的SPO三元组 ( $S, P, O$ 分别表示主体, 属性, 宾语)。

### 1.1. 方法概览

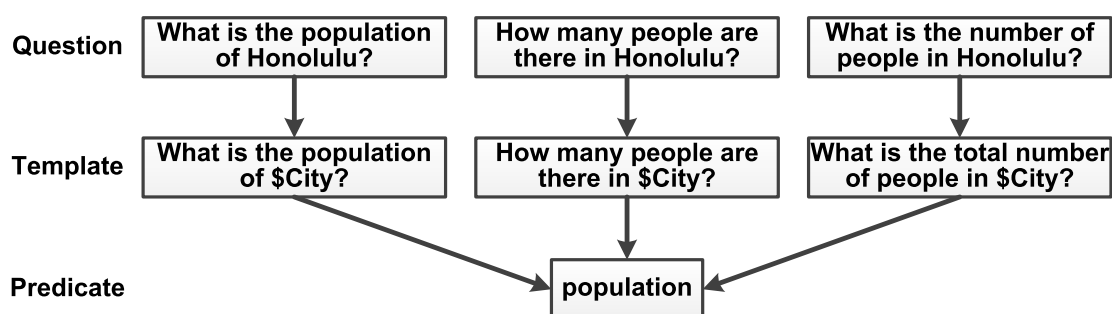


图 5.1: 基于模板的方法

为了回答一个问题, 系统需要首先表示这个问题。所谓表示一个问题, 指的是将问题从自然语言转换为一种能够捕获问题语义和意图的计算机内部表示。然后, 对于每种内部表示, 学习将其映射到知识图谱上的RDF查询。因此, 本章工作的核心之一就是这一内部表示设计, 记为“问题模板”。

**通过模板表现问题** 基于同义词的方法在问题①上的失败, 启发系统通过模板来理解问题。例如, `how many people are there in $city`是问题①的模板。无论\$city指的是檀香山市还是其他城市, 这一模板永远询问人口数的问题。

这样, 问题表示的任务转化为了将问题映射到现有模板的任务。为了完成这一点, 系统将问题中的实体替换为它的概念。如图5.1, Honolulu会被\$city所替代。这一过程并不是直接的。它通过一种称为概念化[87, 50]的机制完成目的。这一机制会自动对输入进行歧义消除 (因此苹果的总部是什么中的苹果会被概念化为\$company而非\$fruit)。概念化机制本身基于一个考虑数百万种概念的语义网络 (Probase [103]), 其拥有足够的粒度来模板化所有类型的问题。

模板的思想对于复杂问题同样起效。通过使用模板, 可以将复杂问题简单地分解为一系列仅对应一个属性的简单问题。以表1.1中的问题①为例, 系统将①分解为Barack Obama’s wife和when was Michelle Obama born。这两个子问题分别对应“marriage→person→name”和“date of birth”。由于第一个问题嵌套于第二个问题, 可知“date of birth”修饰了“marriage→person→name”, 而“marriage→person→name”修饰了Michelle Obama。

**将模板映射到属性** 系统从雅虎问答（Yahoo! Answers）中学习模板以及如何将模板映射到知识图谱中的属性。这一问题与语义解析[13, 14]类似。从模板到属性的映射是多对一的，换言之，每个属性都对应于多个问题模板。系统一共学习了2782个属性的27,126,355种不同的模板。这一巨大的数目保证了基于模板的问答系统的高覆盖率。

学习模板的属性的过程如下所述。首先，对于每个雅虎问答中的问答对，系统提取问题中的实体及其对应值。之后，寻找连接实体和值的“直接”属性。其基本想法是，如果某个模板的绝大多数实例对应于共同的属性，就可以将这一模板映射到这一属性上。例如，假设从模板how many people are there in\$city中得出的问题总是可以映射到属性“population”上，无论\$city特指哪个城市，系统都可以认为这一模板必然会映射到属性“population”上。从模板到知识图谱中复杂结构的学习也采用类似的过程。唯一的区别在于系统寻找对应于一条由多条边组成的，从某个实体导向某个特定值的路径的“扩展属性”。（例如marriage→person→name）。

**本章组织** 本章余下部分的组织形式如下。在第2节中，将会给出KBQA的概览。本章的主要贡献是从QA语料库中学习模板以及通过模板回答自然语言问题。全部技术部分都与这一核心贡献紧密相关。第三节展示了系统如何在线上问答中使用模板。第四节详述了如何从模板中推断属性。这也是基于模板的问答系统的关键步骤。第五节扩展了解决方案，用于回答可以分解为一系列BFQ的复杂问题。第六节扩展了模板的能力来推断复杂的属性结构。实验结果呈现在第7节，第8节讨论了更多的相关工作。第9节做出了小结。

## 第2节 系统概览

本节将要介绍KBQA的一些背景知识及其概览。表5.1中列举了本章使用的符号。

符号	描述	符号	描述
$q$	问题	$s$	主语
$a$	回答	$p$	属性
$\mathcal{QA}$	QA语料库	$o$	宾语
$e$	实体	$\mathcal{K}$	知识图谱
$v$	值	$c$	类别
$t$	模板	$p^+$	扩展属性
$V(e, p)$	$\{v   (e, p, v) \in \mathcal{K}\}$	$s_2 \subset s_1$	$s_2$ 是 $s_1$ 的子串
$t(q, e, c)$	$q$ 的模板 通过概念化 $e$ 为 $c$	$\theta^{(s)}$	$\theta$ 的估计值 在第 $s$ 次迭代

表 5.1: 符号表



**二元事实型QA** 本章主要关注二元事实型问题 (BFQ)，亦即询问某个实体的某种属性的问题。例如，表1.1中除①外的所有问题均为BFQ。

**RDF知识图谱** 给定一个问题，系统在一个RDF知识图谱中寻找其回答。一个RDF知识图谱 $\mathcal{K}$ 是一个 $(s, p, o)$ 格式三元组的集合，这里 $s, p, o$ 分别表示主语，属性和宾语。图1.1通过一个边带标注的有向图展示了一个示例的RDF知识图谱。每个 $(s, p, o)$ 都由一条从 $s$ 指向 $o$ ，标注有属性 $p$ 的边表示。例如，从 $a$ 指向1961的标注有 $dob$ 的边表示RDF三元组 $(a, dob, 1961)$ ，意味着Barack Obama出生于1961年。

**QA语料库** 系统从雅虎问答学习问题模板，其包含有约四千一百万对问答对。这一QA语料库被记为 $\mathcal{QA} = \{(q_1, a_1), (q_2, a_2), \dots, (q_n, a_n)\}$ ，其中 $q_i$ 是某个问题而 $a_i$ 是其回复。每个回复 $a_i$ 含有一个或多个句子，并且确切的事实回答也被包含在回复中。表5.2展示了QA语料库中的一些例子。

Id	问题	回答
$(q_1, a_1)$	When was Barack Obama born?	The politician was born in 1961.
$(q_2, a_2)$	When was Barack Obama born?	He was born in 1961.
$(q_3, a_3)$	How many people are there in Honolulu?	It's 390K.

表 5.2: QA语料库中的QA对示例。

**模板** 通过用实体 $e$ 的一个概念 $c$ 替换 $e$ ，可以从问题 $q$ 中得到模板 $t$ 。这一模板记为 $t = t(q, e, c)$ 。一个问题可能含有多个实体，并且一个实体可能属于多个概念。系统通过上下文相关的概念化过程[103]获得 $e$ 的概念分布。例如，问题when was Barack Obama born?中含有图1.1中的实体 $a$ 。由于 $a$ 属于两个概念： $\$person$  和  $\$politician$ ，系统可以从这一问题中获得两个模板：When was  $\$person$  born?和When was  $\$politician$  born?。

**系统结构** 图8.1展示了问答系统的流水线。它含有两个主要过程：在线QA部分和离线预处理部分。

- **在线过程：** 当一个问题到来，系统首先将其解析和分解为一系列二元事实型问题。这一分解过程将在第5节详述。对于每个二元事实型问题，系统使用概



率推断来寻找它的值，如第3节所示。这一推断基于给定模板的属性分布，亦即 $P(p|t)$ 。这一分布是离线习得的。

- **离线过程：** 离线过程的目标是学习从模板到属性的映射，由 $P(p|t)$ 表示。这一过程将在第4节详述。在第6节中，系统在知识图谱中扩展了属性，以学习更复杂的属性形式（例如图1.1中的marriage→person→name）。

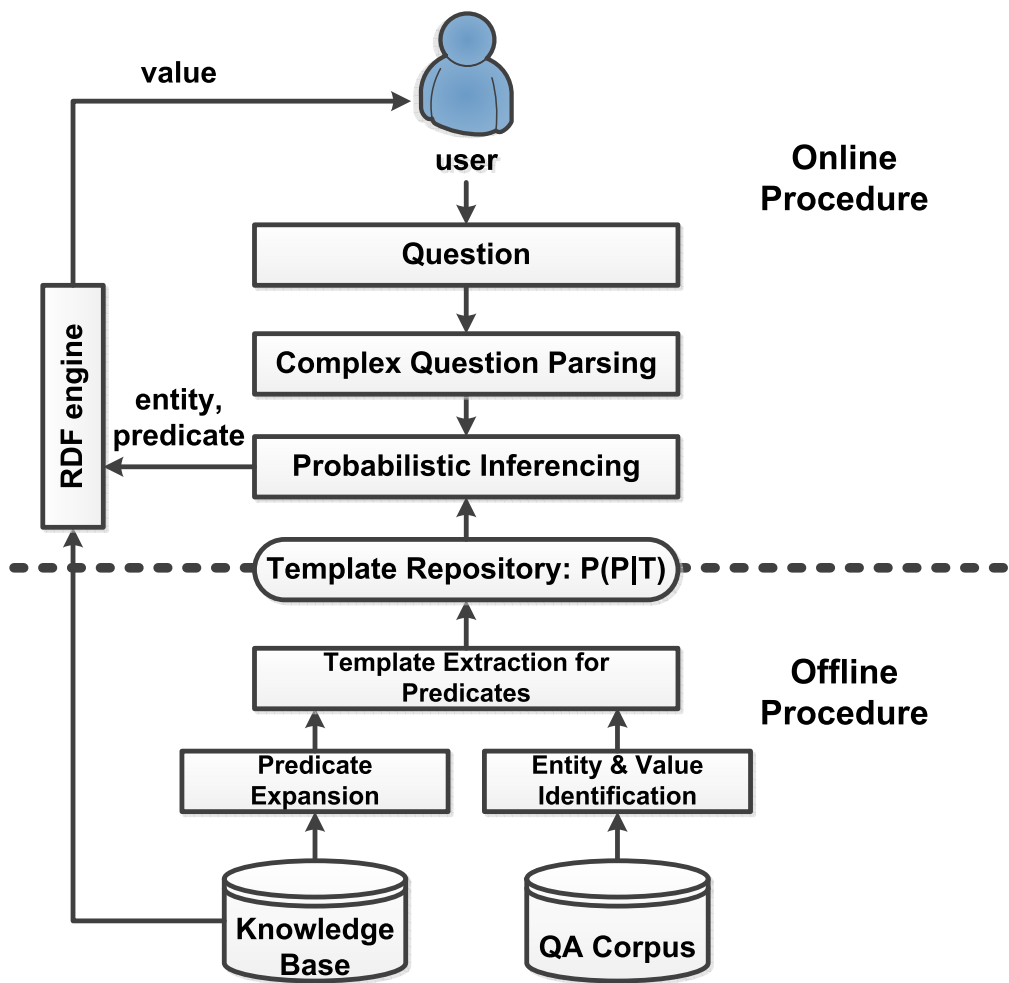


图 5.2: 系统概览

### 第 3 节 本文的方法：KBQA

第3.1.节在概率框架下将问题形式化。这一问题被化简为两个主要部分：离散概率计算和在线推断。第3.2.节中展示有关概率计算的大部分细节，但将 $P(p|t)$ 的计算留在4节。第3.3.节将详述在线问答过程。

### 3.1. 问题模型

KBQA通过QA语料库和知识图谱进行学习。由于问答过程的不确定性（一些问题的意图是模糊的）、不完整性（知识图谱几乎总是不完备的）和噪音（QA语料库中的问答可能是错误的）等问题，本章为知识图谱上的问答系统构建了一个概率模型。需要强调的是从问题意图到知识图谱属性的不确定性。例如，问题Barack Obama来自哪里？至少与Freebase中的两个属性连接：“place of birth”、“place lived location”。在DBpedia中，谁创建了\$organization？与属性“founder”、“father”均相关。

**问题定义 5.1.** 给定问题 $q$ ，问答系统的目标是寻找具有最大概率的回答 $v$ （ $v$ 是一个简单值）：

$$\arg \max_v P(V = v | Q = q) \quad (5.1)$$

为了说明给定问题时如何寻找目标值，系统使用了一个生成模型。从用户问题 $q$ 开始，系统首先通过其分布 $P(e|q)$ 生成/识别它的实体 $e$ 。在得知了问题和实体之后，系统根据分布 $P(t|q, e)$ 产生模板 $t$ 。由于属性 $p$ 仅依赖于 $t$ ，系统可以通过 $P(p|t)$ 来推断 $p$ 。最终，给定实体 $e$ 和属性 $p$ ，系统通过 $v$ 产生回答值 $P(v|e, p)$ 。 $v$ 可以被直接返回，或是嵌入一个自然语言句子作为回答。例5.2阐明了生成过程，并且显示了图5.3中随机变量的依赖关系。基于这个生成模型，可以如下计算 $P(q, e, t, p, v)$

$$P(q, e, t, p, v) = P(q)P(e|q)P(t|e, q)P(p|t)p(v|e, p) \quad (5.2)$$

现在问题5.1被化简为：

$$\arg \max_v \sum_{e, t, p} P(v|q, e, t, p) \quad (5.3)$$

**例 5.2.** 考虑表5.2中 $(q_3, a_3)$ 的生成过程。由于 $q_3$ 中的唯一实体为“Honolulu”，系统通过 $P(e = d | q = q_3) = 1$ 生成实体结点 $d$ （见图1.1）。通过概念化“Honolulu”为\$city，系统生成模板how many people are there in \$city。注意到无论特指哪个城市，这一模板对应的属性总是“population”，系统通过分布 $P(p|t)$ 生成属性“population”。在生成实体“Honolulu”和属性“population”后，目标值“390k”能够轻易地从知识图谱中获取，如图1.1所示。最终系统使用自然语言句子 $a_3$ 作为回答。

**以下小节的概要** 给定了上述的目标函数，问题化简为对式5.2中各个概率项的计算。其中 $P(p|t)$ 在离线过程中计算（见第4节），其他全部概率项可以通过现成的解决方案（例如概念化、NER）计算。第3.2.将详述这些概率的计算过程。第3.3.节将基于这些概率结果详述在线过程。

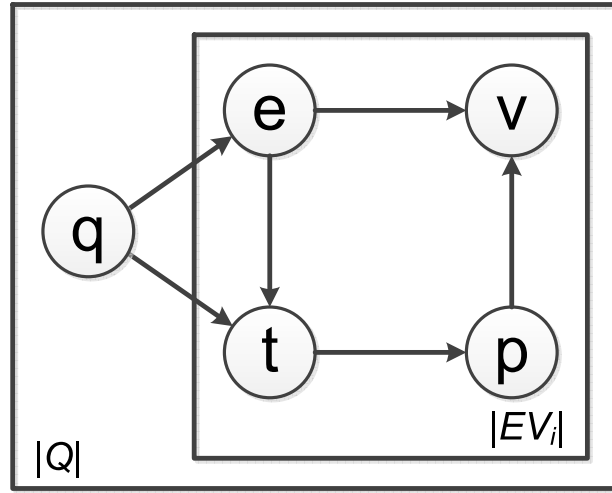


图 5.3: 概率图模型

### 3.2. 概率计算

在本小节中，将计算式5.2中除 $P(p|t)$ 外的各概率项。

**实体分布** $P(e|q)$  这一分布代表从问题中辨识实体。当满足以下两个条件时，将其辨识为实体：(a)它是问题中的一个实体；(b)它在知识图谱中。对于(a)，系统使用Stanford Named Entity Recognizer [33]。对于(b)，系统检验实体的名字是否在知识图谱中。如果存在多个候选实体，简单地给予他们一样的概率。

系统通过 $q$ 的回答优化离线过程中 $P(e|q)$ 的计算。由第4.1.节知，系统已经从问题 $q_i$ 和回答 $a_i$ 中提取了实体-值对 $EV_i$ 。假定 $EV_i$ 中的实体有相等的概率来被生成：

$$P(e|q_i) = \frac{[\exists v, (e, v) \in EV_i]}{|\{e' | \exists v, (e', v) \in EV_i\}|} \quad (5.4)$$

其中 $[\cdot]$ 是Iverson bracket。由第7.5.节知，这一途径比直接使用NER 更为精确。

**模板分布** $P(t|q, e)$  模板有类似\$person何时出生？这样的形式。换言之，它是将一个问中的某个实体（如“Barack Obama”）替换为实体的概念（\$person）的结果。

令 $t = t(q, e, c)$ 表示模板 $t$ 是通过将 $q$ 中实体 $e$ 替换为 $e$ 的概念 $c$ 得到的。由此可得：

$$P(t|q, e) = P(c|q, e) \quad (5.5)$$

其中 $P(c|q, e)$ 是 $e$ 在上下文 $q$ 中的概念分布。本章的工作直接应用了[87]中的概念化方法来计算 $P(c|q, e)$ 。

**值（回答）分布** $P(v|e, p)$  对于实体 $e$ 和一个关于 $e$ 的属性 $p$ ，在知识图谱中寻找属性指向的值 $v$ 是容易的。例如，在图1.1所示的知识图谱中，让实体 $e = \text{Barack Obama}$ ，属性 $p = \text{dob}$ ，很容易就很可能从知识图谱中得到得到Obama出生年份1961。在这一例子中， $P(1961|\text{Barack Obama}, \text{dob}) = 1$ ，因为Obama只有一个生日。有一些属性可能有多个指

向的值（例如Obama的孩子）。在这样的例子中，模型假定所有可能的值有相同的概率。更形式化地，可以通过如下公式计算 $P(v|e, p)$ ：

$$P(v|e, p) = \frac{[(e, p, v) \in \mathcal{K}]}{|\{(e, p, v') | (e, p, v') \in \mathcal{K}\}|} \quad (5.6)$$

### 3.3. 在线过程

在这一过程中，给定用户问题 $q_0$ ，系统可以通过式5.7计算 $p(v|q_0)$ ，并且返回 $\arg \max_v P(v|q_0)$ 作为回答。

$$P(v|q_0) = \sum_{e, p, t} P(q_0) P(v|e, p) P(p|t) P(t|e, q_0) P(e|q_0) \quad (5.7)$$

其中 $P(p|t)$ 由第4节所述的离线学习得到，其他概率项由第3.2节所述的计算方法得到。

**在线计算的复杂度：**在在线计算过程中，系统依次枚举 $q_0$ 的实体、模板、属性和对应值。系统将每个问题的实体数，每个实体的概念数，每个实体-属性对的对应值数视为常量。因此在线计算过程的复杂度是 $O(|P|)$ ，由对属性的枚举而产生。这里 $|P|$ 指知识图谱中的属性数。

## 第4节 属性推断

本节介绍如何从模板中推断属性，也就是 $P(p|t)$ 的估计值。其基本思路是将分布 $P(P|T)$ 视作参数，然后使用极大似然（ML）估计法来估计 $P(P|T)$ 。第4.1节介绍了基于参数估计的第一步，制定观测数据（亦即语料库中的QA对）的似然度。第4.2和4.3节分别阐述参数估计的细节以及其算法实现。

### 4.1. 似然度

算法的推导并不直接公式化似然概率来观察QA语料库（ $\mathcal{QA}$ ），而是先公式化一个更简单的情形——从QA对中提取的一个问题-实体-答案值三元组集合的似然概率。接着构造两个似然概率之间的关系。这种间接公式构造更为直接。 $\mathcal{QA}$ 的一个回答通常是一句包括精确值和其他许多符号的复杂的自然语言。这些符号中很大一部分对于推断属性是无意义的，并且为观察带入噪音。另一方面，在生成模型中直接建立完整答案的模型比较困难，但在其中建立答案值的模型则相对简单。

接下来，第4.1.1节首先从给定的QA对中提取实体-答案值对，从而实现对问题-实体-答案值三元组（ $X$ ）的似然概率的公式化。然后，第5.13节和第4.1.2节建立了QA语料库和 $X$ 的似然概率之间的关系。

#### 4.1.1. 实体-答案值提取

从答案中提取候选值的原则是一个有效实体-答案值对通常在知识图谱中存在一些一致关系。根据这个原则，可以从 $(q_i, a_i)$ 中鉴别出候选实体-答案值对如下：

$$EV_i = \{(e, v) | e \subset q_i, v \subset a_i, \exists p, (e, p, v) \in \mathcal{K}\} \quad (5.8)$$

其中 $\subset$ 表示“是……的子串”。系统支持近似匹配（比如“390K”与“395,327”匹配），从而能增加召回值。如例5.3所示。

**例 5.3.** 考虑表5.2中的 $(q_1, a_1)$ 。许多单词（例如the, was, in）在答案中是无用的。注意到图1.1中， $q_1$ 中的实体Barack Obama与1961由属性“dob”连接，从而提取有效值1961。同时要注意这步中系统也提取了噪音值politician。下面的精炼步骤将展示如何过滤它。

$EV_i$ 的精炼在 $EV(q, a)$ 中系统过滤了噪音对。例如例5.3中的（Barack Obama, politician）。直觉表明：正确值和问题应该属于同一类别。这里问题的类别表示问题的预期答案的类别。问题分类[66]已经有了相关研究。KBQA系统使用UIUC分类框架[61]。并使用[66]中提出的具体分类方法。对于答案值分类，系统参考其属性的分类。属性分类是通过人工标记实现的。因为属性总共只有几千个，因此人工标记是可行的。

#### 4.1.2. 似然函数

在实体-答案值提取后，每个QA对 $(q_i, a_i)$ 被转移到一个问题和一个实体-答案值对集合也就是 $EV_i$ 中。假设实体-答案值对之间是独立的，观察这样的一个QA对的概率为：

$$P(q_i, a_i) = P(q_i, EV_i) = P(q_i) \prod_{(e, v) \in EV_i} P(e, v | q_i) \quad (5.9)$$

因此，整个QA语料库的似然概率为：

$$L_{\mathcal{QD}} = \prod_{i=1}^n [P(q_i) \prod_{(e, v) \in EV_i} P(e, v | q_i)] \quad (5.10)$$

假设每个问题都会生成一个相等的概率，也就是说 $P(q_i) = \alpha$ ，可以得到：

$$\begin{aligned} L_{\mathcal{QD}} &= \prod_{i=1}^n [P(q_i)^{1-|EV_i|} \prod_{(e, v) \in EV_i} P(e, v | q_i) P(q_i)] \\ &= \beta \prod_{i=1}^n \left[ \prod_{(e, v) \in EV_i} P(e, v, q_i) \right] \end{aligned} \quad (5.11)$$

其中 $\beta = \alpha^{n - \sum_{i=1}^n |EV_i|}$ 被视作一个常量。式5.11意味着 $L_{\mathcal{Q}\mathcal{A}}$ 与这些问题-实体-答案值三元组的似然概率成比例。令 $X$ 为从QA语料库中提取的这类三元组集合：

$$X = \{(q_i, e, v) | (q_i, a_i) \in \mathcal{Q}\mathcal{A}, (e, v) \in EV_i\} \quad (5.12)$$

令 $x_i = (q_i, e_i, v_i)$ 来表示 $X$ 中的一项。因而 $X = \{x_1, \dots, x_m\}$ 。本节建立了 $\mathcal{Q}\mathcal{A}$ 的似然概率与 $X$ 的似然概率之间的线性关系。

$$L_{\mathcal{Q}\mathcal{A}} = \beta L_X = \beta \prod_{i=1}^m P(x_i) = \beta \prod_{i=1}^m P(q_i, e_i, v_i) \quad (5.13)$$

现在，最大化 $\mathcal{Q}\mathcal{A}$ 的似然概率等同于最大化 $X$ 的似然概率。用式5.2中的生成模型，通过排除所有模板 $t$ 和属性 $p$ 的联合概率 $P(q, e, t, p, v)$ ，模型能够计算 $P(q_i, e_i, v_i)$ 。式5.14表示了这种似然概率。

$$L_X = \prod_{i=1}^m \sum_{p \in P, t \in T} P(q_i) P(e_i | q_i) P(t | e_i, q_i) P(p | t) p(v_i | e_i, p) \quad (5.14)$$

## 4.2. 参数估计

**目标：**此节通过最大化式5.14来估计 $P(p|t)$ 。模型用参数 $\theta$ 和它对应的对数-似然概率来表示分布 $P(P|T)$ 。同时模型用 $\theta_{pt}$ 来表示概率 $P(p|t)$ 。所以下式被用来估计 $\theta$ ：

$$\hat{\theta} = \arg \max_{\theta} L(\theta) \quad (5.15)$$

其中

$$\begin{aligned} L(\theta) &= \sum_{i=1}^m \log P(x_i) = \sum_{i=1}^m \log P(q_i, e_i, v_i) \\ &= \sum_{i=1}^m \log \left[ \sum_{p \in P, t \in T} P(q_i) P(e_i | q_i) P(t | e_i, q_i) \theta_{pt} P(v_i | e_i, p) \right] \end{aligned} \quad (5.16)$$

**EM估计的直觉：**注意到一些随机变量（例如属性和模板）在概率模型中是隐藏的。这促使本章在参数估计中使用最大化期望算法来估计参数。最终目的是最大化完整数据的似然概率 $L(\theta)$ 。然而，由于它包含对数求和，其计算有一定难度。因此推导转化为最大化其似然概率的下界，即**Q-函数** $\mathcal{Q}(\theta; \theta^{(s)})$ 。Q-函数的定义使用了完整数据的似然概率 $L_c(\theta)$ 。EM算法通过迭代来最大化下界 $\mathcal{Q}(\theta; \theta^{(s)})$ 从而最大化 $L(\theta)$ 。在第 $s$ 轮迭代中，**E-步骤**对每一个给定参数 $\theta^{(s)}$ 计算 $\mathcal{Q}(\theta; \theta^{(s)})$ ；**M-步骤**估计能够最大化下界的参数 $\theta^{(s+1)}$ （下一轮迭代的参数）。

**完整数据的似然概率：**这个函数包括对数求和，因此直接最大化 $L(\theta)$ 在计算上是很困难的。直观上来说，如果参数估计过程知道每个被观察三元组的完整数据，也就是它们是由哪个模板和属性生成的，那么估计的过程会更容易。因此对每个被观察的

三元组 $x_i$ ，估计过程引入一个隐藏变量 $z_i$ 。 $z_i$ 的值是一对属性和模板即 $z_i = (p, t)$ ，用于指示 $x_i$ 是由属性 $p$ 和模板 $t$ 生成的。注意需要同时考虑属性和模板，因为它们在生成时不是独立的。 $P(z_i = (p, t))$ 是 $x_i$ 由属性 $p$ 与模板 $t$ 生成的概率。

记 $Z = \{z_1, \dots, z_m\}$ 。 $Z$ 和 $X$ 一起形成完整数据。这个完整数据的对数-似然概率是：

$$L_c(\theta) = \log P(X, Z | \theta) = \sum_{i=1}^m \log P(x_i, z_i | \theta) \quad (5.17)$$

其中

$$\begin{aligned} P(x_i, z_i = (p, t) | \theta) &= P(q_i, e_i, v_i, p, t | \theta) \\ &= P(q_i)P(e_i | q_i)P(t | e_i, q_i)\theta_{pt}P(v_i | e_i, p) \\ &= f(x_i, z_i)\theta_{pt} \end{aligned} \quad (5.18)$$

$$f(x = (q, e, v), z = (p, t)) = P(q)P(e | q)P(t | e, q)P(v | e, p) \quad (5.19)$$

正如第3.2节所讨论的， $f()$ 可以在估计 $P(p | t)$ 之前被独立计算。所以它被视作一个已知的因子。

**Q-函数：**相比于直接优化 $L(\theta)$ ，式5.20中定义“Q-函数”作为观察完整数据似然概率的期望。这里 $\theta^{(s)}$ 是 $\theta$ 在迭代 $s$ 下的估计值。根据定理5.4，当把 $h(\theta^{(s)})$ 视为常量时， $\mathcal{Q}(\theta; \theta^{(s)})$ 为 $L(\theta)$ 提供了一个下界。因此，算法尝试去优化 $\mathcal{Q}(\theta; \theta^{(s)})$ ，而不是直接优化 $L(\theta)$

$$\begin{aligned} \mathcal{Q}(\theta; \theta^{(s)}) &= E_{P(Z|X, \theta^{(s)})}[L_c(\theta)] \\ &= \sum_{i=1}^m \sum_{p \in P, t \in T} P(z_i = (p, t) | X, \theta^{(s)}) \log P(x_i, z_i = (p, t) | \theta) \end{aligned} \quad (5.20)$$

**定理 5.4 (下界 [24]).**  $L(\theta) \geq \mathcal{Q}(\theta; \theta^{(s)}) + h(\theta^{(s)})$  其中 $h(\theta^{(s)})$ 只随 $\theta^{(s)}$ 改变，对于 $L(\theta)$ 来说可以视作常量。

**E-步骤**中计算 $\mathcal{Q}(\theta; \theta^{(s)})$ 。对于式5.20中的每个 $P(z_i | X, \theta^{(s)})$ ，有：

$$P(z_i = (p, t) | X, \theta^{(s)}) = f(x_i, z_i)\theta_{pt}^{(s)} \quad (5.21)$$

**M-步骤**最大化Q-函数。通过使用拉格朗日乘子，式5.22计算得到 $\theta_{pt}^{(s+1)}$ 。

$$\theta_{pt}^{(s+1)} = \frac{\sum_{i=1}^m P(z_i = (p, t) | X, \theta^{(s)})}{\sum_{p' \in P} \sum_{i=1}^m P(z_i = (p', t) | X, \theta^{(s)})} \quad (5.22)$$



### 4.3. 实现

本节讨论算法5中EM算法的实现。这一算法包含三步：初始化，E步骤和M步骤。

**初始化：**为了避免式5.21中 $P(z_i = (p, t) | X, \theta^{(s)})$ 为全零的情况，模型要求 $\theta^{(0)}$ 在所有满足 $f(x_i, z_i) > 0$ 的 $(x_i, z_i)$ 上均匀分布。从而得到：

$$\theta_{pt}^{(0)} = \frac{[\exists i, f(x_i, z_i = (p, t)) > 0]}{|\{p' | \exists i, f(x_i, z_i = (p', t)) > 0\}|} \quad (5.23)$$

**E步骤：**这一步骤中，算法枚举所有的 $z_i$ ，通过式5.21计算 $P(z_i | X, \theta^{(s)})$ 。这一步骤的复杂度为 $O(m)$ 。

**M步骤：**这一步骤中，对每一个 $\theta_{pt}^{(s+1)}$ ，算法计算 $\sum_{i=1}^m P(z_i = (p, t) | X, \theta^{(s)})$ 。直接计算需要消耗 $O(m|P||T|)$ 的时间，因为算法需要枚举全部可能的模板和属性。接下来，通过对每个 $i$ 只枚举常量的模板和属性，算法的复杂度可以被减少为 $O(m)$ 。

注意到只有 $P(z_i = (p, t) | X, \theta^{(s)}) > 0$ 的 $z_i$ 需要考虑。由式5.19和5.21可知：

$$f(x_i, z_i = (p, t)) > 0 \Rightarrow P(t | e_i, q_i) > 0, P(v_i | e_i, p) > 0 \quad (5.24)$$

由于 $P(t | e_i, q_i) > 0$ ，算法可以减少枚举的模板数。 $P(t | e_i, q_i) > 0$ 意味着算法只枚举从 $q_i$ 中的 $e_i$ 概念化过程中得到的模板。 $e$ 的概念数显然是有上界的，并且可以被看作常量。因此，第7行中枚举的模板 $t$ 的总数是 $O(m)$ 。由于 $P(v_i | e_i, p) > 0$ ，算法可以减少枚举的属性数。 $P(v_i | e_i, p) > 0$ 意味着只有在知识图谱中连接 $e_i$ 和 $v_i$ 的属性需要被枚举。这样的属性数也可以被视作常量。因此M步骤的复杂度是 $O(m)$ 的。

**EM算法的总体复杂度：**假定整个过程重复EM算法 $k$ 次，则总体复杂度为 $O(km)$ 。

## 第5节 复杂问题回答

这一节详细阐述如何回答复杂问题。首先第5.1节将问题形式化为一个最优化问题。第5.2节和第5.3节分别阐述优化量度和算法。

### 5.1. 问题陈述

本节着重关注由一系列BFQ组成的复杂问题，例如表1.1中的问题①可以被分解为两个BFQ：（1）Barack Obama's wife (Michelle Obama)；（2）When was Michelle Obama born? (1964年)。显然，第二个问题的答案依赖于第一个问题的答案。

在解答复杂问题时，分而治之框架可以自然而然地被利用：（1）系统首先把问题分解为一系列BFQ，（2）然后系统依次回答每个问题。既然在第3节已经给出了如何回答BFQ，那么这一节中的关键步骤就是问题分解。

需要强调的是，在一个问题分解的序列中，除了第一个问题之外的每个问题都是一个具有实体变量的问题。只有当变量被指派到一个特定实体之后，问题序列中的问

**Algorithm 5:** 属性推断的EM算法

**Data:**  $X$ ;  
**Result:**  $P(p|t)$ ;

- 1 初始化迭代计数器  $s \leftarrow 0$ ;
- 2 初始化参数  $\theta^{(0)}$ ;
- 3 **while**  $\theta$  未收敛 **do**
  - 4   //E-step ;
  - 5   **for**  $i = 1 \dots m$  **do**
    - 6     通过式5.21估计  $P(z_i|X, \theta^{(s)})$  ;
  - 7   //M-step ;
  - 8   **for**  $i = 1 \dots m$  **do**
    - 9     **for all**  $t \in T$  for  $q_i, e_i$  with  $P(t|q_i, e_i) > 0$  **do**
      - 10       **for all**  $p \in P$  with  $P(v_i|e_i, p) > 0$  **do**
        - 11           $\theta_{pt}^{(s+1)} += P(z_i = (p, t)|X, \theta^{(s)})$  ;
  - 12   通过式5.22标准化  $\theta_{pt}^{(s+1)}$ ;
  - 13    $s += 1$  ;
- 14 **return**  $P(p|t)$

题才能被具体化，而这个特定实体也就是前一个问题的答案。回到之前的例子中去，第二个问题When was Michelle Obama born?在问题序列中是When was \$e born?。在这里，\$e作为一个变量来代表第一个问题Barack Obama's wife答案。从而当给定一个复杂问题 $q$ 后，系统需要将其分解为由 $k$ 个问题形成的序列 $\mathcal{A} = (\check{q}_i)_{i=0}^k$ ，使得：

- 每个 $\check{q}_i (i > 0)$ 都是一个有实体变量 $e_i$ 的BFQ，其值为 $\check{q}_{i-1}$ 的答案。
- $\check{q}_0$ 是一个BFQ，其实体等于 $q$ 的实体。

**例 5.5 (问题序列).** 考虑表1.1中的问题①。一个自然问题序列是 $\check{q}_0 = \text{Barack Obama's wife}$ 和 $\check{q}_1 = \text{when was } \$e_1 \text{ born?}$ 系统也可以替换任意一个子串来构造问题序列，诸如 $\check{q}'_0 = \text{Barack Obama's wife born}$ 和 $\check{q}'_1 = \text{When was } \$e?$ 。但因为 $\check{q}'_0$ 既不是一个可回答的问题也不是一个BFQ，所以后者是无效的。

给定一个复杂问题，系统用递归的方式构造一个问题序列。系统首先用一个实体变量来替换一个子串。如果这个子串是可以被直接回答的BFQ，使它为 $q_0$ 。否则对子串重复以上步骤直到得到一个BFQ 或者子串是一个单独的词汇。然而，正如例5.5所示，许多问题分解是不可行的（或不可回答的）。因此，系统需要度量一个分解的序列有多大可能被回答。更形式化地，使 $\mathbb{A}(q)$ 成为 $q$ 所有分解可能的集合。对于一个分解 $\mathcal{A} \in \mathbb{A}(q)$ ，规定 $P(\mathcal{A})$ 为 $\mathcal{A}$ 是有效（可回答）问题序列的概率。从而问题被简化为：

$$\arg \max_{\mathcal{A} \in \mathbb{A}(q)} P(\mathcal{A}) \quad (5.25)$$

接下来的第5.2.节和第5.3.节将分别阐述对 $P(\mathcal{A})$ 的估计以及如何有效求解最优化问题。

## 5.2. 度量标准

根据直觉，如果问题序列 $\mathcal{A} = (\check{q}_i)_{i=0}^k$ 中的每个问题 $\check{q}_i$ 都是有效的，那么该序列是有效的。因此，需要首先估计 $P(\check{q}_i)$  ( $q_i$ 是有效的概率)，然后将每个 $P(\check{q}_i)$ 合起来来计算 $P(\mathcal{A})$ 。

算法用QA语料库来估计 $P(\check{q}_i)$ 。 $\check{q}$ 是一个BFQ。如果可以通过将 $q$ 的一个子串替换为\$e得到 $\check{q}$ ，那么认为问题 $q$ 与 $\check{q}$ 是匹配的。本节称匹配是有效的，当被替换的子串是 $q$ 中的实体时。例如When was Michelle Obama born?匹配when was \$e born?和when was \$e?。但是，只有前者是有效的因为只有Michelle Obama 是一个实体。本节用 $f_o(\check{q})$ 来表示QA 语料库中匹配 $\check{q}$ 的所有问题的数量，用 $f_v(\check{q})$ 来表示有效匹配 $\check{q}$ 的问题数量。

$f_v(\check{q}_i)$ 和 $f_o(\check{q}_i)$ 都从QA语料库得到计数。这样算法估计 $P(\check{q}_i)$ 为：

$$P(\check{q}_i) = \frac{f_v(\check{q}_i)}{f_o(\check{q}_i)} \quad (5.26)$$

这个式子明显是合理的：匹配数越多， $\check{q}_i$ 可回答的可能性越大。 $f_o(\check{q}_i)$  被用来惩罚过于笼统的问题样式。下面给出一个 $P(\check{q}_i)$ 的例子。

**例 5.6.** 令 $\check{q}_1 = \text{When was } \$e \text{ born?}$ ,  $\check{q}_2 = \text{When was } \$e?$ , QA语料库如表5.2所示。显然， $q_1$ 满足 $\check{q}_1$ 和 $\check{q}_2$ 的样式。但是，因为只有当 $q_1$ 匹配 $\check{q}_1$ 时，被替换的子串才对应一个有效实体“Barack Obama”，因此只有 $q_1$ 是 $\check{q}_1$ 的有效样式。从而得到 $f_v(\check{q}_1) = f_o(\check{q}_1) = f_o(\check{q}_2) = 2$ 。且有 $\check{q}_0 \equiv 0$ 。由式5.26,  $P(\check{q}_1) = 1$ ,  $P(\check{q}_2) = 0$ 。

对于每个给定的 $P(\check{q}_i)$ ，定义 $P(\mathcal{A})$ 。假设 $\mathcal{A}$ 中的每个 $\check{q}_i$ 有效是独立事件。则当且仅当问题序列 $\mathcal{A}$ 中所有 $\check{q}_i$ 有效时，该序列有效。所以 $P(\mathcal{A})$ 可以计算如下：

$$P(\mathcal{A}) = \prod_{\check{q} \in \mathcal{A}} P(\check{q}) \quad (5.27)$$

### 5.3. 算法

给定 $P(\mathcal{A})$ ，算法的目标是找到使 $P(\mathcal{A})$ 最大的问题序列。因为搜索空间巨大，因此这步不能忽略。考虑一个长度也就是字数为 $|q|$ 的复杂问题 $q$ 。 $q$ 中共有 $O(|q|^2)$ 个子串。如果 $q$ 最终被分解为 $k$ 个子问题，那么总搜索空间为 $O(|q|^{2k})$ ，这是不能被接受的。本节提出一个基于动态规划的方法来求解最优化问题。该方法复杂度为 $O(|q|^4)$ 。方法利用了最优化问题的局部最优解性质。定理5.7证明了这个性质。

**定理 5.7 (局部最优解).** 对于复杂问题  $q$ ，令 $\mathcal{A}^*(q) = (\check{q}_0^*, \dots, \check{q}_k^*)$  是 $q$ 的最优分解，则 $\forall 1 \leq i \leq k, \exists q_i \subset q, \mathcal{A}^*(q_i) = (\check{q}_0^*, \dots, \check{q}_i^*)$ 也是 $q_i$ 的最优分解。

基于定理5.7，可以得到一个动态规划（DP）算法。考虑 $q$ 中的一个子问题 $q_i$ 是（1）一个初始BFQ（不可分解）或（2）一个可被进一步分解的问题串中的其中一个。对于情形（1）， $\mathcal{A}^*(q_i)$ 包括一个元素也就是 $q_i$ 本身。对于情形（2）， $\mathcal{A}^*(q_i) = \mathcal{A}^*(q_j) \oplus r(q_i, q_j)$ ，其中 $q_j \subset q_i$ 有最大 $P(r(q_i, q_j))P(\mathcal{A}^*(q_j))$ ， $r(q_i, q_j)$ 是通过将 $q_i$ 中的 $q_j$ 用一个占位符“\$e”替换而生成的问题。从而得到动态规划方程：

$$P(\mathcal{A}^*(q_i)) = \max\{\delta(q_i), \max_{q_j \subset q_i} \{P(r(q_i, q_j))P(\mathcal{A}^*(q_j))\}\} \quad (5.28)$$

其中 $\delta(q_i)$ 是决定 $q_i$ 是否为初始BFQ的指示函数。也就是说，当 $q_i$ 是初始BFQ或 $\delta(q_i) = 0$ 时， $\delta(q_i) = 1$ 。

算法2描述了动态规划算法。算法在外层循环（第1行）中枚举 $q$ 的所有子串。在每个循环中，算法首先初始化 $\mathcal{A}^*(q_i)$ 和 $P(\mathcal{A}^*(q_i))$ （第2-4行）。在内层循环中，算法枚举 $q_i$ 的所有子串 $q_j$ （第5行），然后更新 $\mathcal{A}^*(q_i)$ 与 $P(\mathcal{A}^*(q_i))$ （第7-9行）。注意到算法按照长度升序枚举所有 $q_i$ ，这确保了通过每个被枚举的 $q_j$ ，可以知道它们的 $P(\mathcal{A}^*(q_j))$ 和 $\mathcal{A}^*(q_j)$ 。

因为每个循环枚举 $O(|q|^2)$ 个子串，从而算法2的复杂度为 $O(|q|^4)$ 。在实验的QA语料库中，超过99%的问题字数少于23个（ $|q| < 23$ ），因此这样的复杂度是可以接受的。

**Algorithm 2:** 复杂问题分解

**Data:**  $q$ ;  
**Result:**  $\mathcal{A}^*(q)$ ;  
1 **for**  $q$ 的子串 $q_i$ 从长度1到 $|q|$  **do**  
2      $P(\mathcal{A}^*(q_i)) \leftarrow \delta(q_i)$ ;  
3     **if**  $\delta(q_i) = 1$  **then**  
4          $\mathcal{A}^*(q_i) \leftarrow \{q_i\}$ ;  
5     **for**  $q_i$ 的子串 $q_j$  **do**  
6          $r(q_i, q_j) \leftarrow$  替换 $q_i$ 中的 $q_j$ 为“\$e”;  
7         **if**  $P(\mathcal{A}^*(q_i)) < P(r(q_i, q_j))P(\mathcal{A}^*(q_j))$  **then**  
8              $\mathcal{A}^*(q_i) \leftarrow \mathcal{A}^*(q_j) \oplus r(q_i, q_j)$ ;  
9              $P(\mathcal{A}^*(q_i)) \leftarrow P(r(q_i, q_j))P(\mathcal{A}^*(q_j))$ ;  
10 **return**  $\mathcal{A}^*(q)$

## 第6节 属性扩展

在知识图谱中，许多关系不是由一个直接属性表达的，而是由一条由许多属性组成的路径表示的。正如图1.1所示，在RDF数据库中，“spouse of”关系是由三个属性  $marriage \rightarrow person \rightarrow name$  表达的。本章称这些多属性的路径为扩展属性。利用扩展属性来回答问题可以高效提升KBQA的覆盖率。

**定义 5.8 (扩展属性).** 一个扩展属性  $p^+$  是一个属性序列  $p^+ = (p_1, \dots, p_k)$ 。本章把  $k$  称为  $p^+$  的长度。如果存在一个宾语序列  $s = (s_1, s_2, \dots, s_k)$  使得  $\forall 1 \leq i < k, (s_i, p_i, s_{i+1}) \in \mathcal{K}$  且  $(s_k, p_k, o) \in \mathcal{K}$ ，则说  $p^+$  连接了主语  $s$  和宾语  $o$ 。正如  $(s, p, o) \in \mathcal{K}$  表示了  $p$  连接了  $s$  和  $o$ ，这里将  $p^+$  连接  $s$  和  $o$  记作  $(s, p^+, o) \in \mathcal{K}$ 。

第3节中提出的KBQA模型可以充分适应属性拓的问题。系统只需要一些轻微的调整就可以使得KBQA对扩展属性有效。第6.1节展示了这种调整。第6.2节展示了如何使得它对十亿级别的数据库有效。最后，第6.3节中展示了如何选择一个合理的属性长度来保证最高的效率。

### 6.1. 对扩展属性的KBQA

上文曾提到，对单一属性的KBQA由两大部分组成。在离线部分，系统计算对给定模板的属性分布  $P(p|t)$ ；在线上部分，系统抽取问题的模板  $t$ ，然后通过  $P(p|t)$  计算它的属性。当把  $p$  替换成  $p^+$  之后，系统做了如下调整：

在离线部分，系统学习了对扩展属性的问题模板。例如计算  $P(p^+|t)$ 。  $P(p^+|t)$  的计算仅仅只要知道  $(e, p^+, v)$  是否在  $\mathcal{K}$  中。如果系统生成了所有的  $(e, p^+, v) \in \mathcal{K}$ ，就可以计算这一存在性。第6.2节展示了这一生成过程。

在线上部分，系统用扩展属性来回答问题。系统可以通过RDF数据库中的  $e$  到  $p^+$  来计算  $P(v|e, p^+)$ 。例如，让  $p^+ = marriage \rightarrow person \rightarrow name$ ，为了从图1.1中的数据库来计算  $P(v|Barack\ Obama, p^+)$ ，系统从节点  $a$  开始遍历，然后经过节点  $b$  和  $c$ ，最后得到了  $P(Michelle\ Obama|Barack\ Obama, p^+) = 1$ 。

### 6.2. 扩展属性的生成

一个简单的生成所有的扩展属性的方式是对数据库中的每一个节点进行广度优先搜索（BFS）。然而，扩展属性的数量随着属性的长度指数级增长。所以当数据量达到十亿级别的时候，BFS的开销是无法承受的。

为了实现扩展属性的生成，系统首先对属性的长度  $k$  设置了限制来提升延展性，也就是说，它只搜索长度小于等于  $k$  的扩展属性。下一个小节会展示如何得到一个合适的  $k$ 。本节通过另外两个方面来提升延展性：（1） $s$  的约减；（2）内存高效的BFS。



**$s$ 的约减:** 离线处理的过程只对QA语料库中出现过至少一次的 $s$ 有兴趣。因此, 系统只用那些在QA语料库中的问题中出现过的宾语作为BFS的起始节点。这一策略很大程度上减少了生成的 $(s, p^+, o)$ 的数量, 因为这些实体的数量比起在十亿级别数据库中的要少得多。在系统使用的数据库(15亿实体)和QA语料库(79万不同实体)中, 这一过滤策略理论上可以减少 $(s, p^+, o)$ 的数量 $1500/0.79 = 1899$ 倍。

**内存高效的BFS:** 为了在1.1TB大小的数据库中使用BFS, 本节使用了基于磁盘的多源BFS算法。在一开始, 系统将在QA语料库(记作 $S_0$ )中出现过的所有的实体读入内存, 并在 $S_0$ 创建了一个散列索引。第一轮中, 系统通过扫描磁盘上的所有RDF三元组一次, 并将三元组的主语和 $S_0$ 结合, 我们就得到了所有长度为1的 $(s, p^+, o)$ 。本节建立的对 $S_0$ 的散列索引, 允许算法在线性时间内完成这一操作。第二轮中, 系统将所有的三元组读入进内存中, 然后建立对所有的宾语 $o$ 建立散列索引(记作 $S_1$ )。然后再次扫描RDF, 并将RDF中三元组的主语和 $s \in S_1$ 结合。现在系统得到所有的长度为2的 $(s, p^+, o)$ , 并将它们读入进内存中。系统重复上述的“索引+扫描+结合”操作 $k$ 次来得到所有的长度为 $p^+.length \leq k$ 的 $(s, p^+, o)$ 。

这个算法非常高效, 其时间消耗主要用在了 $k$ 次扫描数据库上。散列索引的建立和结合的操作在内存中执行, 时间消耗对于磁盘上的I/O来说是可以忽略不计的。注意到从 $S_0$ 开始的扩展属性的数量总是比数据库的大小要小得多, 因此可以被容纳在内存中。对于实验使用的数据库(KBA, 更多细节请参阅实验章节)和QA语料库, 只需要存储21M的 $(s, p^+, o)$ 三元组。所以很容易将他们读入内存。假设 $\mathcal{K}$ 的大小是 $|\mathcal{K}|$ , 算法找到的 $(s, p^+, o)$ 三元组的数量是 $\#spo$ , 它消耗了 $O(\#spo)$ 的内存, 算法的时间复杂度是 $O(|\mathcal{K}| + \#spo)$ 。

### 6.3. $k$ 的选择

扩展属性的长度限制 $k$ 影响了属性扩展的效率。 $k$ 越大,  $(s, p^+, o)$ 越多, 导致更高的答案覆盖率。然而, 这也产生了更多的无意义的 $(s, p^+, o)$ 三元组。例如, 图1.1中, 扩展属性 $marriage \rightarrow person \rightarrow dob$ 连接了“Barack Obama”和“1964”, 但是他们明显没有关系, 对于KBQA也没有用。

属性扩展需要选择一个能够得到最多的有意义的关系, 并且排除最多无意义的关系的 $k$ 的值。本文使用Wikipedia的Infobox估计最佳的 $k$ 。Infobox存储了实体的一些知识, 并且大部分条目都是以“主语-属性-宾语”的三元组的形式存储的。Infobox中的条目可以被视作有意义的关系。因此,  $k$ 的选择中首先列举一些长度为 $k$ 的 $(s, p^+, o)$ 三元组, 然后测试它们中有多少在Infobox中出现。选择过程希望看到 $k$ 值的减少。

特别地, 实验按照它们出现的频率的顺序, 从RDF数据库中选择了前17000个实体。实体 $e$ 出现的频率被定义为在 $\mathcal{K}$ 中存在的使得 $e = s$ 的 $(s, p, o)$ 三元组的数量。选取这些实体是因为他们有更多的知识, 因此更值得信任。对于这些实体, 使用第6.2节中提



出的BFS生成了他们的长度为 $k$ 的 $(s, p^+, o)$ 三元组。然后，对于每一个 $k$ ，计算这些可以在Wikipedia的Infobox中找到对应的 $(s, p^+, o)$ 三元组的数量。更形式化地，假设 $E$ 是作为例子的条目的集合， $SPO_k$ 是长度为 $k$ 的 $(s, p^+, o) \in \mathcal{K}$ 。定义 $valid(k)$ 来度量 $k$ 对于有意义的关系的数量，方法如下：

$$valid(k) = \sum_{s \in E} |\{(s, p^+, o) | (s, p^+, o) \in SPO_k, \exists p, (s, p, o) \in Infobox\}| \quad (5.29)$$

在KBA和DBpedia上得出的 $valid(k)$ 的值在表5.3中展示。当 $k = 3$ 时，有效的扩展属性的数量显著减少。这说明了大部分有意义的因素在这个长度内可以被表示出来，所以系统选择了 $k = 3$ 。

k	1	2	3
KBA	14005	16028	2438
DBpedia	352811	496964	2364

表 5.3:  $valid(k)$

## 第7节 实验

第7.1.节中阐明实验设置；第7.2.节验证了概率模型的合理性；第7.3.节和第7.4.节中分别评估了系统的有效性和效率；第7.5.节验证了KBQA的三个组成部分的有效性。

### 7.1. 实验设置

KBQA系统运行在在一台装配了Intel Xeon CPU, 2.67 GHz, 2 processors, 24 cores, 96 GB 内存, 64 bit windows server 2008 R2的服务器上。它使用Trinity.RDF [110]作为RDF引擎，这一引擎被部署在了6台服务器上，并且一共使用了284.1GB的内存和1.5TB的磁盘资源。

**数据库** 实验部分使用了三个开放领域的RDF数据库。由于商用保密协议本文无法公开第一个数据库的名称，在这里称它为KBA。KBA有15亿实体和115亿SPO三元组，共占1.1TB空间。SPO三元组包含了2658个不同的属性和1003种不同的种类。为了实验的再现性，实验也在其他两个知名的数据库Freebase和DBpedia上测试了KBQA系统。Freebase包含1.16亿个条目和29亿SPO三元组，占了380GB存储空间。DBpedia包含了560万条目，1.11亿三元组，占了14.2G存储空间。

**QA语料库** QA语料库包含了从Yahoo! Answer上得到的4100万QA二元组。如果对于一个问题由多个回答，则只考虑“最佳答案”。

**测试数据** 实验在QALD-5 [99], QALD-3 [96]和QALD-1 [95]上分别测试了KBQA, 它们是测评基于知识图谱的问答系统设计的。这些测试数据的基本信息展示在了表5.4中。由于KBQA关注问答的BFQ, 所以也展示了对于这些数据库中BFQ问题的数量( $\#BFQ$ )。

	$\#total$	$\#BFQ$
QALD-5	50	12
QALD-3 [96]	99	41
QALD-1 [95]	50	27

表 5.4: 评估标准

**对比方法** 实验把KBQA和13个QA系统进行比较, 表5.5列举了这些系统。

系统	来源	系统	来源
squall2sparql	Q3[96]	Xser [106]	Q5
SWIP	Q3	APEQ	Q5
CASIA	Q3	QAnswer	Q5
RTV	Q3	SemGraphQA	Q5
Intui2	Q3	YodaQA	Q5
Scalewelis	Q3	DEANNA	[107]
gAnswer	[116]		

表 5.5: 对比方法。Q5表示QALD-5; Q3表示QALD-3。

## 7.2. 概率模型的合理性

接下来实验解释为什么一个概率模型是必须的。在问题理解的每一个步骤中, 有些选择会给系统的决策带来不确定性, 在表5.6中展示了每一个决策的候选答案数量。这种不确定性需要系统使用一个好的概率模型来表示。

举例来说,  $P(t|e, q)$ 表示将一个问题和它的实体转化成模型的时候的不确定性。比如对问题How long it Mississippi river?来说, 系统很难从一些候选项中直接决定这个实体的概念是river或是location。

概率	解释	平均个数
$P(e q)$	一个问题对应的平均实体数	18.7
$P(t e, q)$	一个实体-问题对对应的模板数	2.3
$P(p t)$	一个模板对应的属性个数	119.0
$P(v e, p)$	一个问题-属性对对应的值个数	3.69

表 5.6: 概率图模型中每个随机变量的不同取值个数。

### 7.3. 有效性

为了评估KBQA的有效性，本节进行了如下实验。对于线上部分，实验评估了回答问题的准确性和召回率。在线下部分，实验评估了属性推断的覆盖率和准确性。

#### 7.3.1. 问题回答的有效性

**指标** 当一个QA系统发现当前问题没有答案时，它可能会返回null，所以实验对一个QA系统返回的非空（不一定是正确答案）( $\#pro$ )的答案的数量和正确答案( $\#ri$ )的数量做了记录。然而，事实上，一个系统只能部分正确地回答一个问题（例如，仅仅找到正确答案的一部分）。因此实验评测也需要那些部分正确的答案( $\#par$ )的数量。当KBQA找到一个属性时，问题的答案便可以从RDF数据库中被找到。因此对于KBQA来说 $\#pro$ 是KBQA找到的属性的数量。 $\#ri$ 是KBQA找到正确的属性的数量。 $\#par$ 是KBQA找到部分正确的属性的数量。例如，对于问题Which city was \$person born?来说，“place of birth”是一个部分正确属性。因为它可能返回一个国家或者一个村庄，而不是问题所要找的一个城市。

现在实验部分已经定义了评估指标：准确性  $P$ ，部分准确性  $P^*$ ，召回率  $R$ 和 部分召回率  $R^*$ ：

$$P = \frac{\#ri}{\#pro}; P^* = \frac{\#ri + \#par}{\#pro}; R = \frac{\#ri}{\#total}; R^* = \frac{\#ri + \#par}{\#total}$$

实验也对关于BFQ的召回率和部分召回率有兴趣，分别记作 $R_{BFQ}$ 和 $R_{BFQ}^*$ ：

$$R_{BFQ} = \frac{\#ri}{\#BFQ}; R_{BFQ}^* = \frac{\#ri + \#par}{\#BFQ}$$

**QALD-5 和 QALD-3的结果** 表5.7 和表5.8中展示了结果。对于所有的竞争者，表格直接使用了它们论文中的结果，可以发现在所有的数据库上，除了在准确性上略逊

于squal2sparql, KBQA战胜了其他所有的竞争者。这是因为squal2sparql对于所有的问题都使用了真人来标注识别实体和属性。另外KBQA在DBpedia上表现的最好, 这是因为QALD主要是为了DBpedia设计的。对于大多数QALD中的问题, KBQA可以直接从DBpedia中找到正确的答案。

	#pro	#ri	#par	R		R*		P	P*
Xser	42	26	7	0.52		0.66		0.62	0.79
APEQ	26	8	5	0.16		0.26		0.31	0.50
QAnswer	37	9	4	0.18		0.26		0.24	0.35
SemGraphQA	31	7	3	0.14		0.20		0.23	0.32
YodaQA	33	8	2	0.16		0.20		0.24	0.30
				R	R <sub>BFQ</sub>	R*	R <sub>BFQ</sub> *		
KBQA+KBA	7	5	1	0.10	0.42	0.12	0.50	<b>0.71</b>	<b>0.86</b>
KBQA+Freebase	6	5	1	0.10	0.42	0.12	0.50	<b>0.83</b>	<b>1.00</b>
KBQA+DBpedia	8	8	0	0.16	0.67	0.16	0.67	<b>1.00</b>	<b>1.00</b>

表 5.7: QALD-5的结果。

	#pro	#ri	#par	R	R <sub>BFQ</sub>	R*	R <sub>BFQ</sub> *	P	P*
squal2sparql	96	77	13	0.78	0.95	0.91	0.95	0.80	0.94
SWIP	21	14	2	0.14	0.24	0.16	0.24	0.67	0.76
CASIA	52	29	8	0.29	0.56	0.37	<b>0.61</b>	0.56	0.71
RTV	55	30	4	0.30	0.56	0.34	0.56	0.55	0.62
gAnswer	76	32	11	0.32	-	0.43	-	0.42	0.57
Intui2	99	28	4	0.28	0.54	0.32	0.56	0.28	0.32
Scalewelis	70	1	38	0.01	0.41	0.39	0.41	0.01	0.56
KBQA+KBA	25	17	2	0.17	0.42	0.19	0.46	<b>0.68</b>	<b>0.76</b>
KBQA+FB	21	15	3	0.15	0.37	0.18	0.44	<b>0.71</b>	<b>0.86</b>
KBQA+DBp	26	25	0	0.25	<b>0.61</b>	0.25	<b>0.61</b>	<b>0.96</b>	<b>0.96</b>

表 5.8: QALD-3的结果。

召回率分析 表5.7和表5.8中的结果表明了KBQA有一个相对低的召回率。主要原因是KBQA只回答BFQ（二元事实性问题），然而QALD包含了很多非BFQ问题。当只考

虑BFQ时，召回率分别上升至0.67和0.61。实验对于KBQA在QALD-3上没有回答的问题进行了研究，结果发现原因很大程度上是因为KBQA对模型匹配用了相对严苛的标准。无法回答的情况通常发生在一个稀少的属性和一个稀少的问题进行了匹配时。15个无法回答的情况中有12个是因为这个原因。例如，对于问题In which military conflicts did Lawrence of Arabia participate?，在DBpedia 中这个问题的属性是battle。KBQA对于这部分属性没有充分进行训练。如果将KBQA和一个同义词QA系统结合起来，可能就会有效增加召回率。当KBQA中发生了误匹配时，系统可以用基于同义词的QA系统的答案作为替代。这超出了本章主要讨论的内容，因此不在这里进行阐释。

**QALD-1的结果** 实验将KBQA和DEANNA进行了比较，结果列在了表5.9中。DEANNA是基于同义词的BFQ问答系统。对于DEANNA来说，*#pro*是被转化成SPARQL的问题数量。结果表明KBQA的准确性比DEANNA高得多。由于DEANNA是一个典型的基于同义词的QA系统，这一结果表明了基于模板的问答系统在准确性方面比基于同义词的要好。

	#pro	#ri	#par	$R_{BFQ}$	$R_{BFQ}^*$	P	P*
DEANNA	20	10	0	0.37	0.37	0.5	0.5
KBQA + KBA	13	<b>12</b>	0	<b>0.48</b>	<b>0.48</b>	<b>0.92</b>	<b>0.92</b>
KBQA + Freebase	14	<b>13</b>	0	<b>0.52</b>	<b>0.52</b>	<b>0.93</b>	<b>0.92</b>
KBQA + DBpedia	20	<b>18</b>	1	<b>0.67</b>	<b>0.70</b>	<b>0.90</b>	<b>0.95</b>

表 5.9: QALD-1的结果。

### 7.3.2. 属性推断的有效性

接着本节阐释KBQA属性推断的有效性：（1）KBQA学习了大量的自然语言的问题的模板和属性（覆盖率），（2）对大多数的模板，KBQA可以推断出正确的属性（准确率）。

**覆盖率** 表5.10中展示了KBQA学习的模板和属性的数量，并与最新的基于同义词的Bootstrapping [107, 97]进行了对比。Bootstrapping从数据库和网络文本中对属性学习了同义词（BOA式样，主要是网络文本中主语和宾语之间的部分）。BOA可以被看做是一种模板，它们之间的关系可以被看做是属性。

结果表明，即使Bootstrapping用了更大的语料库，KBQA依然比它找到了明显更多的模板和属性。这意味着KBQA在属性推断方面更有效：（1）模板的数量确保

	KBQA +KBA	KBQA +Freebase	KBQA +DBpedia	Bootstrapping
语料	41M QA	41M QA	41M QA	256M sentences
模板数	<b>27126355</b>	1171303	862758	471920
属性数	<b>2782</b>	4690	1434	283
一个属性的平均模板数	<b>9751</b>	250	602	4639

表 5.10: 属性推断的覆盖率

了KBQA对不同问题的理解；（2）属性的数量确保了KBQA对于不同关系的理解。因为KBA是实验使用的最大的数据库，基于KBA的KBQA生成了数量最多的模板，所以在接下来的实验中主要关注对KBA的测试。

**准确性** 此评测的目标是评估对于一个给定的模板，KBQA是否能生成正确的属性。为了这个目的，实验按照出现频率选择了最高的100个模板。实验也随机选择了100个频率大于1（只出现一次的模板可能意义十分模糊）的模板。对于每一个模板 $t$ ，使用人工核对它的属性 $p$ （最大值化 $P(p|t)$ ）是否正确。和QALD-3上的评估相似，在某些情况下属性是部分正确的。结果被展示在了表5.11中。对于两个模板集，KBQA都有更高的准确率。对于频率最高的100个模板，KBQA的准确率甚至达到了100%。这表明了基于模板的属性推断的质量。

模板来源	<i>#right</i>	<i>#partially</i>	P	P*
随机100	67	19	67%	86%
最高100	100	0	100%	100%

表 5.11: 属性推断的准确率

## 7.4. 效率

本节首先给出和其他问答系统的运行时间比较，然后给出KBQA的时间复杂度分析。

**运行时间实验** 运行时间由两部分组成：线下与线上。线下的处理过程，主要是学习模板，用了1438分钟。时间的消耗主要是由庞大的数据量造成的：十亿级别的数据库和上百万的QA对。鉴于线下部分只用运行一次，这个时间的消

耗是可以承受的。线上部分主要负责回答问题，实验把线上部分的时间消耗在表5.12中和gAnswer和DEANNA进行了比较。KBQA用了79ms，比gAnswer快了13倍，比DEANNA快了98倍，这意味着KBQA可以有效地支持实时的QA。

	时间消耗	时间复杂度	
		问题理解	问题评估
DEANNA	7738ms	NP-hard	NP-hard
gAnswer	990ms	$O( V ^3)$	NP-hard
		问题解析	概率推导
KBQA	<b>79ms</b>	$O( q ^4)$	$O( P )$

表 5.12: 时间消耗

**复杂度分析** 表5.12中展示了它们的时间复杂度， $|q|$ 表示问题的长度， $|V|$ 表示RDF图中向量的个数。所有的KBQA的步骤都可以在多项式时间内完成，然而gAnswer和DEANNA都有NP-hard的步骤。gAnswer的问题理解的时间复杂度是 $O(|V|^3)$ ，这种复杂度对于十亿级别的数据库来说是不能接受的。相比之下，KBQA的时间复杂度是 $O(|q|^4)$ 和 $O(|P|)$ （ $|P|$ 是不同属性的数量），和数据库的大小无关。正如第5.3节中提到的，超过99%的问题的长度都是小于23的。因此，在时间复杂度方面，KBQA比其他QA系统有着更好的表现。

### 7.5. KBQA的具体模块评估

实验评估了KBQA的三个关键模块：实体-值的识别（第4.1节），复杂问题的回答（第5节），属性扩展（第6节）。

**实体和值的识别的准确性** 大多数过去的研究都主要关注实体的抽取，这种技术并不能被用在同时抽取实体和值上。所以，实验只能和最新的实体识别的研究对比[33]。实验随机从问题语料中选择了50个答案在知识图谱中的问答对。通过人工判断抽取的结果是否正确。本文的方法正确识别了36个问答对（72%）。相比之下，斯坦福NER只识别的15个问答对（30%）。结果表明对于实体的共同抽取要比单独抽取要好。

**回答复杂问题的有效性** 因为没有对复杂问题回答的有效基准测试集，实验构造了如表5.13中的8个问题。这里列举的所有问题都是真实的用户提出的典型复杂问题。实验比较了KBQA和其他两个最新的QA系统：Wolfram Alpha 和gAnswer。表5.13中展示了



复杂问题	KBQA	WA	gA
How many people live in the capital of Japan?	Y	Y	N
When was Barack Obama's wife born?	Y	Y	N
What are books written by author of Harry Potter?	Y	N	N
What is the area of the capital of Britain?	Y	N	N
How large is the capital of Germany?	Y	N	N
What instrument do members of Coldplay play?	Y	N	N
What is the birthday of the CEO of Google?	Y	N	N
In which country is the headquarter of Google located?	Y	N	N

表 5.13: 复杂问题回答. WA 表示 Wolfram Alpha, gA 表示 gAnswer.

结果, 实验发现KBQA在回答复杂问题方面战胜了它的对手, 这表明KBQA对回答复杂问题是有效的。

**属性扩展的有效性** 接下来实验将会测评系统在属性扩展在两方面的有效性。第一, 属性扩展可以识别更多的属性。第二, 扩展属性使KBQA学习更多的模板。表5.14中展示了评估结果。可以发现 (1) 相较于直接属性 (长度为1), 扩展属性 (长度为2 到 $k$ ) 生成了十倍于前者的属性数量; (2) 归因于扩展属性, 模板的数量增加了57倍。

实验进一步使用了两个案例来阐明: (1) 扩展的属性是有意义的, (2) 扩展属性是正确的。表5.16中列举了学习出的5个扩展属性。可以发现KBQA识别出的这些属性都是有意义的。实验进一步选择了一个扩展属性 $marriage \rightarrow person \rightarrow name$ , 来验证从这一属性中学习出的模板是否正确并有意义。表5.15中列举了5个模板, 这些模板都是合理的。

长度	模板数	属性数
1	467,393	246
2 to $k$	26,658,962	2536
Ratio	57.0	10.3

表 5.14: 属性扩展的效果

Who is \$person marry to?
Who is \$person's husband?
What is \$person's wife's name?
Who is the husband of \$person?
Who is marry to \$person?

表 5.15:  $marriage \rightarrow person \rightarrow name$ 的对应属性

扩展的属性	语义
marriage → person → name	配偶
organization_members → member → alias	机构成员
nutrition_fact → nutrient → alias	营养值
group_member → member → name	团队成员
songs → musical_game_song → name	游戏音乐

表 5.16: 属性扩展的例子

第 8 节 相关工作

在计算机领域，问答系统是一个经典的研究问题。在信息检索（IR），数据挖掘和自然语言处理（NLP）领域它都被广泛研究。本节首先根据数据来源调研了一些相关工作。然后本节调研基于知识图谱的的问答系统。最后调研了RDF数据管理的最新进展。

**自然语言文本vs知识库** 问答系统对于语料库的质量有着很强的依赖性。传统的问答系统使用web文本或是Wikipedia作为它们的语料库来回答问题。在这一分类中的最新的方法[78, 56, 22, 44]通常将网络文档或是Wiki中的句子作为问题的答案，并根据它们和问题的相关性来进行打分。他们也使用一些去噪音的方法，比如说问题分类[66, 111]，来增加答案的质量。最近几年，许多大规模知识库的诞生，例如Google Knowledge Graph，Freebase[10]和YAGO2[45]，为建立新的QA系统提供了机会[72, 98, 97, 36, 107, 31]。这些知识库比起依赖于网络文本的QA系统，有着更系统的架构，并且有着更清晰和可靠的回答。

**基于知识图谱的问答系统** 基于知识图谱的问答系统的核心处理是对问题的属性识别。例如，对于问题How many people are there in Honolulu，如果系统能找到属性“population”，这个问题就能被回答。根据属性识别的方式分类，这些知识库的发展经历了三个主要的阶段：基于规则, 基于关键词, 和基于同义词 。基于规则的方法用人为创造的规则将问题映射到属性。例如，Ou et al. [72]认为形如What is the xxx of entity?的问题应该被映射到属性xxx。人为构建的规则通常有高的准确率，但是召回率很低。基于关键词的方法[98]用问题中关键词或词组作为特征来找到问题和属性之间的映射。但是通常，很难用关键词来找到问题和复杂属性之间的映射。系统很难基于关键词，例如“how many”，“people”，“are there”等，来映射问题 how many people are there in ...?到属性“population”。基于同义词的方法[97, 107]通过考虑属性的同

义词，扩展了基于关键词方法。这使得它可以回答更多的问题。这个方法的主要影响因素是同义词的质量。Unger et al. [97]用bootstrapping [36] 来生成同义词。Yahya et al. [107]则用Wikipedia来生成同义词。然而，由于和基于关键词的方法相同的原因，基于同义词的方法仍然不能回答复杂问题。True knowledge [94]用关键词和词组来表示一个模板。True knowledge 应该被分类到基于同义词的方法。相比之下，本章的问题模板将实体概念化来表示问题。

总而言之，相比于本章使用问题模板的理解方式，之前所有的基于知识库的QA系统仍然在准确率和召回率方面有着他们的弱点。

## 第9节 小结

基于知识库的QA系统现在已经成为了一项重要且可行的工作。本章在一个大型开放领域RDF知识库的基础上建立了一个问答系统。系统和之前的工作有以下四点不同：（1）它用模板理解问题，（2）它用模板抽取来学习从模板到属性的映射，（3）用RDF中的扩展属性来提升知识库的覆盖率，（4）理解复杂问题来提高对于问题的覆盖率。实验表明KBQA是有效且高效的，尤其是在准确性方面，比其他的QA系统都要优秀。

## 第六章 针对序列-序列的自然语言处理任务的迁移学习框架

为了将开放领域问答系统迁移到具体领域中，需要对系统中的若干具体自然语言处理模型进行领域适配，包括词性标注、实体识别等。针对开放领域NLP任务的模型往往在特定领域效果不好。迁移学习是一种为特定领域建模的有效方法。之前迁移学习的方法要么不能解决序列-序列标注任务，要么还非常初步。本章提出了一个针对序列-序列NLP任务的迁移学习框架。这个框架包含了两个重叠的神经网络，一个是开放领域的神经网络，一个是特定领域的神经网络。参数从开放领域的神经网络迁移到特定领域。这个框架有两个优点：(1) 可以简单地用CNN和RNN层来处理序列-序列标注任务；(2) 这个框架能够在多层神经网络中迁移多层次信息（比如：词层次、短语层次）。本章在不同NLP任务上进行了广泛的实验，包括词性标注、分块和语句情感分析。和最先进的工作相比，本章提出的方法在这些任务中表现最好。

### 第1节 引言

问答系统的复杂流程中，涉及到很多自然语言处理（NLP）任务。许多重要的NLP任务可以被建模成序列-序列问题，比如词性标注（POS）[20]，分块[19]，以及依存句法解析[16]。序列-序列模型在语言理解中起着非常重要的作用，因为自然语言可以被建模为词序列，并且许多NLP任务的输出是序列。

大多数以前的NLP研究专注于开放领域自然语言理解。但是由于自然语言的多样性和模糊性[37, 87]，开放（源）领域的模型在处理特定（目标）领域时通常会招致更多错误。这问题对于神经网络来说尤为严重，因为基于词向量的神经网络模型通常会过拟合[75]。现有的NLP模型通常使用开放领域的标准数据集，因此当在特定领域的语料上运行时，它们效果严重退化。这促使本章为特定领域训练特定模型。

然而，特定领域模型训练的主要挑战是训练数据的不足[48]。这个问题也存在于基于深度学习的方法中。现有研究[23, 48]表明（1）在开放领域和特定领域中的NLP应用仍然共享许多共同特征（比如通用的词汇、相似的句法）。（2）开放领域语料库通常比特定领域的更丰富。所以一个有效的方法是从开放领域到特定领域的迁移学习。

当前的迁移学习方法，包括应用深度学习的迁移学习方法，通常对序列-序列任务不适用或是效果受限。根据文献调研[74]，大多数以前的有监督的迁移学习方法可以分为三类：实例迁移，特征迁移和参数迁移。

- **实例迁移方法**[48] 从开放领域和目标领域中重新给训练实例加权。它使用概率模型来指导加权过程。但是，该模型仅适用于分类问题。
- **特征迁移方法** 为源领域和目标领域学习出统一的特征空间。许多最近的工作使用深层神经网络来做到这一点。[113]使用一个编码-解码的框架，其中不同领域的数使用相同的编码函数以达到统一表示。作者之后又使用了更复杂的自动编码器来优化他们的工作[114]。DANN框架[2]利用领域指示标识来学习统一的特征表示。DANN通过使领域指示标识无法区分数据的来源，来学习统一的特征表示。需要注意的是，在特征迁移方法中，统一特征表示的学习与具体任务的标签无关。因此，学习过程不受标签的监督。在学习过程中缺少序列信息使得特征迁移方法不能解决序列标注问题。
- **参数迁移方法** 发现源领域和目标领域之间的共享参数。它们通常用于多任务学习，但可以被修改而用于迁移学习。最先进的多任务学习方法[19]使用卷积神经网络进行序列标注。该方法仍然是初步的，因为它仅使用共享词向量层来进行参数迁移。

现在说明本章模型的出发点和合理性。对特定领域的模型来说，主要的挑战是数据缺失，这使得特定领域中的一些参数训练不足。在预测特定域中的样本时，最佳情况是相应的参数已经被充分训练。否则，对于具有未充分训练的参数的样本，存在两种情况：(i) 样本的标签与开放领域的标签相同；(ii) 样本的标签与开放领域的标签不同。情况(i)肯定比情况(ii)容易处理得多。因此，本章的主要挑战是情况(ii)。

为了解决情况(ii)，模型需要利用来自开放领域的知识，但不是直接使用它的开放领域输出标签。因此，模型在特定领域中使用另一个更高的层 $S_2$ ，如图6.1(a)所示。开放领域和特定领域分别具有层 $O_1$ 和 $S_1$ 。 $S_2$ 接受来自 $O_1$ 和 $S_1$ 的输入。在这个框架中，来自开放领域的知识通过侧连接（红边）传递。 $S_2$ 有两个作用：(1) 区分来自特定领域的参数是否被充分训练；(2) 如果不是，学习如何利用来自开放领域的知识辅助决策。因此对于情况(ii)， $S_2$ 将认识到来自特定领域的参数训练不足。它将主要使用 $O_1$ 的输出（红边）进行预测。由于 $O_1$ 的输出被更充分地训练，所以 $S_2$ 能够基于充分训练的输出来预测标签。这样，模型解决了情况(ii)。例子6.1和图6.1(a)中展示这个过程的一个基本示例。

**例 6.1.** 图6.1中展示了情况(ii)的示例。模型要预测输入字的情感标签（1为正，0为负）。假设模型在开放领域中给出了“cat”和“kitt”的标签（都是1）。模型想要预测特定领域中的“kitt”的标签，这是未知的。但是模型知道特定领域中的“cat”的标签是0。额外的层 $S_2$ 接受来自 $O_1$ 和 $S_1$ 的输入。对于“kitt”， $S_1$ 生成一个模糊的输出到 $S_2$ 。因此， $S_2$ 将主要根据 $O_1$ 的输出预测标签。由于“cat”和“kitt”在开放领域中经过充分训练，并且它们的语义相似， $O_1$ 为它们生成类似的输出。通过从 $O_1$ 接受类似“cat”的输入， $S_2$ 会预测“kitt”为同一个标签，也就是0。

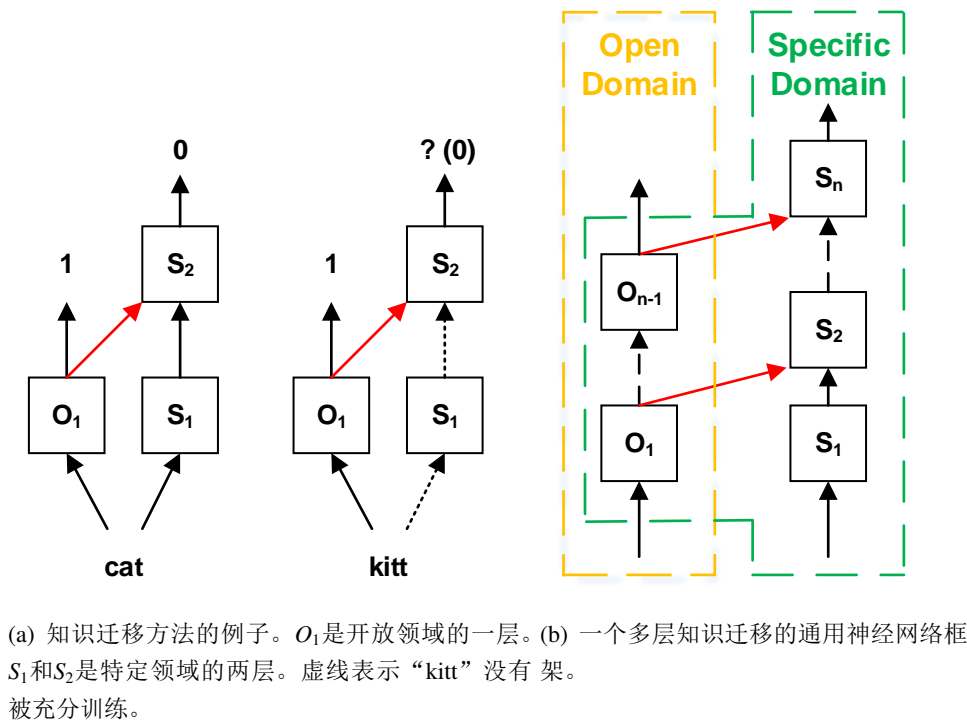


图 6.1: Model Illustration

需要注意的是，图6.1中的层都是可变的。因此模型可以使用CNN，RNN，以及LSTM来对序列-序列标签任务进行处理。

此外，上面的迁移方法实际上是通过层到层侧连接（从 $O_1$ 到 $S_2$ ）实现的。由于深层神经网络的主要优点是多层知识表示，模型的框架也允许利用层-层迁移单元进行多层知识迁移。相比之下，传统的多任务学习方法[19]只允许词向量层参数迁移。图6.1(b)中展示了多层知识迁移的通用框架。这样，模型可以在深层神经网络中迁移词层次/短语层次/句子层次的知识。

**贡献** 本章的贡献如下：

- 本章提出了一个针对序列-序列NLP任务的迁移学习框架。本章的框架可以很方便地实现对多层次知识的转移。
- 本章给出了拥有词向量层和LSTM层的具体实现，称为TransferLSTM。
- 本章在词性标注、分块和情感分析任务上进行了广泛的实验。结果表明，本章的方法确实迁移了知识，并且表现比最先进的解决方案和基准方法好。

第 2 节 相关工作及其不足

迁移学习是机器学习的一个重要任务[74]，并且在NLP领域中吸引了很多研究兴趣。一些以前的工作也为研究NLP任务上的基于深度神经网络的迁移学习。在



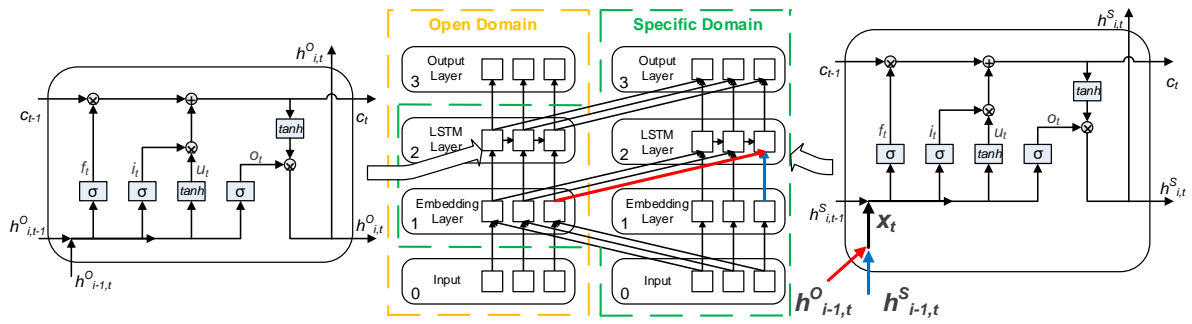


图 6.2: 由词向量层和LSTM层组成的序列-序列标记模型。每层的下标 $i$ 表示它的层数 $i$ 。黄色框表示开放领域的LSTM网络。绿色框表示由LSTM网络（右侧）和两个与开放领域的重叠层组成的特定领域网络。两个重叠层存储开放领域的知识（参数矩阵）。他们的参数矩阵在特定领域训练期间被冻结。这里要着重指出的是，侧连接被用于将开放领域知识迁移到特定领域。LSTM层和输出层（在右侧）分别从词向量层和LSTM层（在左侧）接收迁移的知识。

[37, 113]中，作者使用深层神经网络（DNN）将数据从源领域迁移到目标领域。DNN由每个领域的一个统一编码器和两个解码器组成。因此，如果使用目标领域的解码器用于源领域数据，模型期望获得适合于目标领域的数据。该模型有三个缺点。首先，它仅使用词袋模型作为输入，并且无法实现短语/句子层次的知识迁移。第二，在迁移期间完全无视标签，即，迁移学习框架不能理解标签。第三，它不能用于序列-序列标注。另一个工作[19]使用用于领域适配的多任务学习框架。这项工作解决了数据的领域适配问题。他们还每次从两个域中选择一个数据样本来学习参数。在这种方法中的迁移学习将神经网络视为黑盒。因此，它无法利用DNN的多层表示。

在DNN之前，NLP应用中的迁移学习研究主要关注于实例迁移[48]，特征迁移[113]以及参数转移[19]。在[48]中，作者从概率分布的角度解决实例加权问题。他们假设目标领域和源领域具有相似的概率分布。它们基于该假设实现几个适应启发算法。[17]提出了一种基于规则的NER领域适应的方法。他们设计了一种称为NERL的高级语言。NERL可用来实现特定领域的NER标注。

本章决定使用DNN的迁移学习，是由于其最近在NLP应用中令人印象深刻的性能。[20]使用卷积神经网络对句子进行建模。该工作在许多NLP任务中表现良好，包括词性标注，分块，命名实体识别和语义角色标注。许多DNN也用于不同的应用，并表现出很大的价值。例如，[62]使用耦合LSTM来匹配问题与答案。[64]使用递归神经网络进行词向量学习。所有这些优点促使本章在NLP应用程序中使用DNN进行迁移学习。

### 第3节 框架

本节详细阐述这个通用框架。图6.1(b)展示了这个框架。黄色框中的网络是开放



领域的深度神经网络。绿色框中的是特定领域的神经网络，它包含 $n-1$ 个和开放领域（左侧）重叠的层。此外，它还包含 $n$ 个自己的层（右侧）。侧连接从开放领域将知识（在重叠层中）迁移到特定领域。

下面形式化对网络进行定义。用 $h_i^O, h_i^S \in \mathbb{R}^m$ 分别表示开放领域和特定领域的第 $i$ 层的输出。这里 $n_i$ 是第 $i$ 层中的单元数。这里将开放领域中第 $i$ 层的参数表示为 $\Theta_i^O$ ，将特定领域中第 $i$ 层参数表示为 $\Theta_i^S$ 。

对于特定领域的神经网络，重叠层 $h_i^O$ 仅接受来自开放领域 $h_{i-1}^O$ 的输出。而非重叠层 $h_i^S$ 接受来自 $h_{i-1}^S$ 和 $h_{i-1}^O$ 的输出：

$$h_i^O = f(h_{i-1}^O) \quad (6.1)$$

$$h_i^S = f(h_{i-1}^S + W_{i-1}h_{i-1}^O) \quad (6.2)$$

其中， $W_{i-1} \in \mathbb{R}^{n_{i-1} \times n_{i-1}}$ 是侧连接的权重矩阵。 $f$ 是激活函数。

**训练过程** 包含两个阶段：（1）首先使用开放领域数据来训练开放领域神经网络（即 $\Theta_i^O$ ）。（2）然后使用特定领域的的数据来训练特定领域的神经网络中的其余参数（即 $\Theta_i^S$ 和 $W_i$ ）。在反向传播过程中，随着误差值向后传播，每个神经元计算其梯度。如果神经元在重叠层中，则停止该神经元的反向传播。在权重更新期间，只有非重叠层的神经元更新它们的权重。以这种方式， $n-1$ 个重叠层在第二阶段期间保持它们的参数不变。

对于**预测过程**，特定领域的神经网络如下进行工作：对于特定领域中的每个输入样本，它既通过重叠层，又通过非重叠层。因此，开放领域中的知识能够在预测中被使用。

## 第4节 TransferLSTM: 一个具体实现

本节给出具有词向量层和LSTM层的上述框架的具体实现。这一实现被称为TransferLSTM。

### 4.1. 开放领域的神经网络

本节详细阐述开放领域的神经网络。这个网络接受自然语言形式的输入。它使用词向量层（层1）作为词层次的表示，用LSTM层（层2）作为短语、句子层次的表示。输出层（层3）为特定应用生成输出。

**词向量层**对于词典大小为 $|V|$ 的输入，将每个单词映射为 $d$ 维向量：

$$h_{1,j}^O = M_v r(w_j) \quad (6.3)$$

$M_v \in \mathbb{R}^{d \times |V|}$ 是词向量矩阵， $r(w_j) \in \{0, 1\}^{|V|}$ 是输入句第 $j$ 个词 $w_j$ 的one-hot表示。

**LSTM层** 词向量被送入LSTM层。在该层中，每个单元是存储单元。该单元从下层和前一单元获得输入。因此，它能获得当前和过去的信息。具体地，存储单元由四个主要部分组成：输入门，忘记门，存储单元状态和输出门。首先，忘记门从词向量层和先前单元接收输入，并且决定丢弃哪些值。然后输入门决定更新哪些值。存储单元状态存储更新的值。最后，输出门决定要输出什么。

这里将第 $t$ 个单元的激活状态定义为 $i_t$ （输入门）， $f_t$ （忘记门）， $o_t$ （输出门）和 $c_t$ （存储单元状态）。它们由以下复合函数计算：

$$\begin{aligned}
 i_t &= \sigma(W^{(i)}h_{1,t}^O + U^{(i)}h_{2,t-1}^O + b^{(i)}), \\
 f_t &= \sigma(W^{(f)}h_{1,t}^O + U^{(f)}h_{2,t-1}^O + b^{(f)}), \\
 o_t &= \sigma(W^{(o)}h_{1,t}^O + U^{(o)}h_{2,t-1}^O + b^{(o)}), \\
 u_t &= \tanh(W^{(u)}h_{1,t}^O + U^{(u)}h_{2,t-1}^O + b^{(u)}), \\
 c_t &= i_t \odot u_t + f_t \odot c_{t-1}, \\
 h_{2,t}^O &= o_t \odot \tanh(c_t)
 \end{aligned} \tag{6.4}$$

其中 $h_{1,t}^O$ 是词向量层的输出， $\sigma$ 是logistic sigmoid函数， $\odot$ 表示向量内积， $h_{2,t}^O$ 是该单元的输出， $W^{(i)}, U^{(i)}, b^{(i)}$ 是参数。

**输出层** 输出层接在LSTM层之后：

$$h_3^O = f(h_2^O) \tag{6.5}$$

其中， $f$ 是激活函数。

对于不同的应用， $f$ 可以变化（比如sigmoid函数或softmax函数）。本章会在实验部分展示如何在不同应用的时候变化输出层。

#### 4.2. 知识迁移的特定领域模型

如图6.2所示，特定领域的神经网络（在绿色框中）包含开放领域中的词向量层和LSTM层。此外，它自身也包含四个新层（右侧）：输入层，词向量层，LSTM层和输出层。这里LSTM层和输出层可以从开放领域利用侧连接进行知识迁移。

**LSTM层** 从开放领域的词向量层（通过迁移）和特定领域的词向量层接收输入。对于LSTM网络，它首先连接 $h_{1,t}^S$ 和 $h_{1,t}^O$ 来生成 $x_t$ 作为输入，而不是直接从 $h_{1,t}^S$ 接收输入。根据公式6.2，则有：

$$x_t = h_{1,t}^S + W_1 h_{1,t}^O \tag{6.6}$$

其中， $W_1$ 是侧连接的权重矩阵。因此，完整的函数如下：

$$\begin{aligned}
 i_t &= \sigma(W^{(i)}x_t + U^{(i)}h_{2,t-1}^S + b^{(i)}), \\
 f_t &= \sigma(W^{(f)}x_t + U^{(f)}h_{2,t-1}^S + b^{(f)}), \\
 o_t &= \sigma(W^{(o)}x_t + U^{(o)}h_{2,t-1}^S + b^{(o)}), \\
 u_t &= \tanh(W^{(u)}x_t + U^{(u)}h_{2,t-1}^S + b^{(u)}), \\
 c_t &= i_t \odot u_t + f_t \odot c_{t-1}, \\
 h_{2,t}^S &= o_t \odot \tanh(c_t)
 \end{aligned} \tag{6.7}$$

输出层利用从开放领域LSTM层迁移的知识：

$$h_3^S = f(h_2^S + W_2 h_2^O) \tag{6.8}$$

**实现细节** 开放领域和特定领域都用随机参数进行初始化。实验使用mini-batched AdaGrad进行非凸优化。Dropout rate设置为0.2。

## 第5节 实验

实验使用三个典型的NLP任务（词性标注，分块，情感分析）来评估本章提出的方法。

结果表明：（1）本章的方法成功地使用从开放领域迁移的知识来提高效果；（2）本章的方法在各种应用中表现最好，特别是对于序列-序列标注任务。

### 5.1. 设置

所有的实验在配置如下的电脑上实现：Intel酷睿i7 4.0GHz CPU，32GB内存，GeForce GTX 980 GPU。

**基准方法：**实验使用两个没有使用迁移学习的基准方法。两个基准方法都直接使用图6.2中的开放领域DNN。基准方法一由开放领域数据训练。实验使用它来验证本章的方法是否比开放领域NLP模型效果好。基准方法二由特定领域数据训练。实验使用它来验证本章的方法是否从开放领域迁移有效的知识以改善结果。

**NLP应用：**实验在不同的NLP任务上评估本章的方法，包括句子分类和序列-序列标注。具体来说，实验在三种典型的NLP任务上实现了本章的方法：词性标注（序列-序列学习，多分类），词分块（序列-序列学习，多分类），句子情感分析（句子分类，二分类）。

### 5.2. 词性标注

词性标注的目的得到句子的每个词的词性标签。实验在人民日报标注语料上测试本章的方法，人民日报标注语料有着294240句话以及44种不同的标签。

**输出层：**为了将输出层和优化函数适应此问题，实验使用softmax函数对标签进行预测。

$$\begin{aligned}\hat{p}_{\theta}^S(y) &= \sigma(W_s^S(h_{2,t}^S + W_2 h_{2,t}^O) + b_s^S), \\ \hat{y}_t^S &= \arg \max_y \hat{p}_{\theta}^S(y)\end{aligned}\quad (6.9)$$

其中， $\hat{y}_t^S$ 是特定领域的输出， $\sigma$ 是softmax函数， $h_{2,t}^S$ 是LSTM层的第 $t$ 个单元的输出， $W_s^S, b_s^S$ 是各层的参数。实验用categorical cross entropy作为优化函数。

**特定领域数据生成：**现在介绍如何为实验生成特定领域的语料库。人民日报是新闻文章的集合。为了提取特定领域的语料，实验将包含领域单词的所有文章视为特定领域的文章。例如，为了生成“经济”领域的语料，实验提取包含“经济”一词的所有文章。然后使用那些文章中的所有句子作为特定领域的预料。实验在四个领域进行这样的操作：经济，教育，科学，军事。对于每个特定领域，使用人民日报标注语料中的其余句子作为开放领域语料。实验使用85%的特定领域的语料库作为训练数据，其余作为测试数据。表6.1中列出了统计信息。

领域	训练数据	测试数据	单词准确率			
			基准方法一	基准方法二	Stanford	TransferLSTM
开放	209970	N/A	N/A	N/A	N/A	N/A
经济	39740	5961	94.6	93.8	95.0	<b>95.2</b>
教育	12172	1825	93.5	91.9	93.0	<b>94.8</b>
科学	11231	1684	94.1	91.8	93.2	<b>94.5</b>
军事	3206	480	95.0	89.0	92.4	<b>95.5</b>

表 6.1: 词性标注的数据和结果。

实验使用单词准确率作为度量。结果显示在表6.1中。实验同时记录了最先进的模型Stanford词性标注[93]的性能。实验用特定领域语料来训练它。本章的方法在所有领域中效果都好过其他对比实验。由于基准方法一是利用开放领域数据进行训练的，因此结果验证了本章的方法优于开放领域的词性标注算法。由于基准方法二是通过特定领域语料库训练的，因此这个结果验证了本章的方法能成功地从开放领域迁移有用知识以改善效果。在所有这些领域中，本章的方法比基准方法好。这验证了本章的方法的有效性。

实验还发现精度提高和特定领域的语料大小之间存在相关性。基准方法二、Stanford词性标注和TransferLSTM之间的差距在军事领域最大，而军事领域也是样本最少的。这是因为较少的数据使直接训练更难。因此，本章的方法对语料更少的领域更有帮助。

### 5.3. 词分块

词分块标记出一句话中的连续部分，并且把它们连接成有语义的高阶单元。本小节测试了词分块任务上的效果。实验使用IOB标签对每个词进行标注来实现分块。

**输出层** 实验使用一个和词性标注类似的输出层（softmax函数）和优化函数（categorical cross entropy）。

**数据集** 实验使用Brown语料库[54]，它有65857句子。这个语料库有15个文本类别（例如，社论，宗教，科幻小说）。因此，实验将每一个类别分别视为特定领域，并评估本章方法在这些领域上的实验效果。对于每个特定领域，实验使用其余14个领域的语料作为开放领域语料。实验使用特定领域语料的90%作为训练数据，其余作为测试数据。表6.2中列出了这些领域语料的统计信息。

领域	训练数据	测试数据	领域	训练数据	测试数据
Press 1	4501	500	Press 2	3072	341
Press 3	1899	211	Religion	1815	201
Skill	3999	444	Popular lore	5172	574
Belles-lettres	8179	908	Miscellaneous	2698	299
Learned	7727	858	Fiction 1	4519	502
Fiction 2	4031	447	Fiction 3	976	108
Fiction 4	4865	540	Fiction 5	4649	516
Humor	1174	130			

表 6.2: 分块的数据统计。

表6.3中展示了实验的准确度 $P$ （检测到的短语中正确短语的百分比），召回率 $R$ （由词分块找到的短语的百分比）和F1分数。实验使用Stanford的解析器[16]作为对比。本章的方法在大多数情况下优于基准方法。结果表明，本章的方法适用于不同的文本类别的词分块。

### 5.4. 情感分析

实验还在分类问题上评估本章方法的有效性。具体来说，实验在两个领域：电影评论和Twitter帖子上测试句子情感分析问题的领域迁移效果。

**输出层：**为了使输出层和优化函数适应这个问题，实验使用了sigmoid函数对给定句子 $x$ 进行打分，判断它是否是积极情感。特定领域输出层计算如下分数：

$$s^S(x) = \sigma(W_s^S(W_2 h_{2,n_2}^O + h_{2,n_2}^S) + b_s^S) \quad (6.10)$$

领域	基准方法一			基准方法二			Stanford			TransferLSTM		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Press 1	56	94	70	55	92	69	88	95	91	93	96	<b>94</b>
Press 2	49	93	64	49	89	63	86	94	90	90	94	<b>92</b>
Press 3	44	92	60	44	86	58	85	94	<b>89</b>	79	92	85
Religion	42	92	57	40	87	55	80	94	89	90	92	<b>91</b>
Skill	46	90	61	46	85	60	84	94	89	87	92	<b>89</b>
Popular.	52	91	66	51	90	65	85	94	89	89	92	<b>90</b>
Belles.	47	94	63	46	92	61	85	95	90	91	95	<b>93</b>
Miscell.	61	94	74	60	94	73	85	95	<b>90</b>	82	94	88
Learned	48	92	63	48	92	63	86	95	90	89	93	<b>91</b>
Fiction 1	47	95	63	46	92	61	88	95	91	92	95	<b>93</b>
Fiction 2	40	96	56	40	93	56	90	95	92	89	96	<b>92</b>
Fiction 3	37	96	54	36	85	51	81	92	86	87	95	<b>91</b>
Fiction 4	52	96	67	52	94	67	93	96	<b>94</b>	90	96	93
Fiction 5	42	95	59	42	93	58	87	95	91	87	96	<b>91</b>
Humor	44	92	60	40	85	54	88	95	<b>91</b>	86	89	88

表 6.3: 分块的结果

其中,  $s^S$ 是特定领域的输出。 $\sigma$ 是sigmoid函数,  $W_s^S$ ,  $b_s^S$ 是输出层的参数,  $h_{2,n_2}^S$ 是LSTM层最后一个单元的输出。实验使用binary cross entropy作为优化函数。

**数据集:** 实验使用Stanford情感树库 (SSTb) [86]作为电影评论语料, 以及Stanford的Twitter情感语料库 (STS) [38]作为Twitter语料。SSTb包含五个情感标签: 非常负, 负, 中性, 正和非常正。STS包含三个情感标签: 正, 负和中性。在本实验中, 只考虑二元情绪分类问题。对于SSTb, 实验删除中性句子, 并将两个负类和两个正类分别合并到负类和正类。在STS中, 实验删除中性句子并保留其余两个类。数据集的统计信息显示在表6.4中。

	训练数据	测试数据
STS	1599848	359
SSTb	7785	1821

表 6.4: STS和SSTb的数据。

从Tweet领域迁移到电影评论领域由于STS语料库的大小远大于SSTb，实验使用STS作为开放领域语料，SSTb作为特定域语料。除了使用基准方法一和基准方法二，实验还与STS基准的最先进的方法进行了比较。结果显示在表6.5中。

从结果来看，本章的方法击败了基准方法和最先进的对比实验。本章的方法和基准方法之间的比较验证了：（1）当与基准方法一相比时，本章的方法比仅仅在开放领域语料上训练的模型更好。（2）与基准线实验二相比，本章的方法能够使用开放领域知识迁移来提高效果。

模型	准确率
SCNN[26]	85.5
RAE[86]	82.4
RNTN[86]	85.4
Paragragh-Vec[58]	87.8
CNN-non-static[51]	87.2
CNN-multichannel[51]	<b>88.1</b>
DRNN[47]	86.6
Constituency Tree-LSTM[91]	88.0
基准方法一	77.5
基准方法二	76.9
TransferLSTM	<b>88.1</b>
基准方法一 + CNN-multichannel	72.5
基准方法二 + CNN-multichannel	81.0
Framework + CNN-multichannel	82.7

表 6.5: 情感分析的结果，从tweet领域迁移到电影评论领域。

实验还评估其他模型能否从本章的框架中得到提升。实验在本章的框架上应用了CNN-multichannel方法。比较结果在表6.5中显示。基准方法一和基准方法二仍然是分别通过开放领域数据和特定领域数据训练的模型。准确率的增加验证了不同的模型可以从本章的框架中获得提升。需要注意的是，实验直接使用他们的CNN-multichannel开源代码用于基准方法二+CNN-multichannel方法。但是CNN-multichannel的准确率比其论文中的报告差[51]。

从电影评论领域迁移到Tweet领域 实验也对一个小的领域的知识是否可以提升更大领域的结果感兴趣。因此，实验还使用SSTb作为开放领域和STS作为特定领域。本章的方法和SSTb的最先进的方法的结果显示在表6.6。



模型	准确率
CharSCNN (random init.)	81.9
SCNN (random init.)	82.2
LProp[89]	<b>84.7</b>
MaxEnt[38]	83.0
基准方法一	77.5
基准方法二	82.2
TransferLSTM	82.8

表 6.6: 情感分析的结果，从电影评论领域迁移到tweet领域。

从表6.6中可以看出，本章的方法与最先进的对比方法相近。这是可以接受的，因为这里的源领域小得多，不适合迁移。当实验与基准方法比较时，准确率仍然有提高。这验证了本章的方法也能将知识从较小的领域迁移到较大的领域，以提高效果。

第 6 节 小结

为了解决领域问答系统构建中的领域模型迁移问题，本章研究了NLP应用中的迁移学习的问题。本章提出了一个利用神经网络的通用框架。该框架有两个优点：（1）它对序列-序列标注任务很友好；（2）它能够实现多层知识迁移。该框架使用两个神经网络分别用于开放领域建模和特定领域建模。特定领域的神经网络与开放领域的神经网络共享部分层。知识通过来自共享层的侧连接进行迁移。本章还给出了具有一个词向量层和一个LSTM 层的框架的详细实现。实验部分对典型的NLP任务进行了实验，包括词性标注，词分块和情感分析。所有这些实验的结果都表明本章的方法能够迁移开放领域知识以改善结果，并可以适应多种NLP任务。

## 第七章 领域知识挖掘

领域问答的基础在于领域知识图谱。对于特定领域，其高质量、结构化的知识往往是不存在，或者是极少的。本章希望从一般文本描述中抽取富含知识的句子，并将其结构化，作为问答系统的知识源。特别的，对于不同的领域，其“知识”的含义是不一样的。有些数据对于某一领域是关键知识，而对于另一领域则可能毫无意义。传统的知识提取方法没有考虑具体领域特征。

本章提出了领域相关的富含知识的句子提取方法，*DAKSE*。*DAKSE*从领域问答语料库和特定领域的纯文本文档中学习富含知识的句子表示。本章在真实数据上的实验验证了*DAKSE*可以以很高的准确率和召回率提取出富含知识的句子。本章还进一步将*DAKSE*的结果应用于领域信息提取，以自动提取结构化的领域知识。

### 第1节 概述

当我们在阅读文档搜索目标信息时，人类并不会以稳定的速度来浏览所有的词语[28]。相反，人的眼睛会四处移动，定位文本的有意义部分，并建立一个整体的感知[65]。扫视的能力能帮助人类跳过大量无用的信息，并专注于富含知识的句子。这引发了一个问题：机器如何像人类一样提取富含知识的句子？这个问题在本文中被称为富含知识句子的抽取问题。

例如，当阅读示例7.1中的斯坦福大学的语料库时，AI研究者会认为句子s1富含更多的信息，给予更多的关注。相比之下，大学生可能跳过s1，但会关注s2。因此，一个句子是否富含知识对于不同的用户是不同的。富含知识的句子提取系统需要为AI研究人员提取s1，为大学生提取s2。

**例 7.1** (斯坦福大学语料库). ... (s1) *Feifei Li is currently the director of the Stanford Artificial Intelligence Lab.* ... (s2) *Full-time undergraduate tuition was \$42,690 for 2013-2014 in Stanford.* ...

从纯文本抽取知识的问题已经作为开放信息抽取（Open IE）[4, 29, 15]，关系抽取[36, 109]和句子抽取[41]被进行了研究。开放信息抽取从纯文本中提取所有的结构化关系。关系抽取只提取指定的关系（例如来自知识库的谓词）。句子抽取提取有意义的句子，通常用于文档概括问题。富含知识的句子提取与这三个问题的专注点不同。

与开放信息抽取相比，富含知识的句子抽取着重于抽取句子。如示例7.1所示，一个句子是否富含知识取决于不同用户的需求。一个统一的开放信息抽取框架是不知道不同用户的需求的。在示例7.1中，开放信息抽取系统不能为不同的用户分别提取 $s_1$ 和 $s_2$ 。因此，开放信息抽取不能直接用于富含知识的句子抽取。

与关系抽取相比，富含知识的句子提取是无监督的。关系提取仅识别指定的关系，而不同的用户关注不同的关系。在例子7.1，通过知道“学费”对于在校大学生是富含知识的关系，关系抽取系统可以从 $s_2$ 中为大学生抽取相应的知识。然而，对于不同的不同用户，他们有多样的需求，并且对于特定需求的一个定义良好的模式标注数据是不存在的。在大多数情况下，一个领域往往没有预定义模式和大量的标签数据。这使得对于来自不同用户的多种需求，关系提取是不适用的。

与句子提取相比，富含信息的句子提取可以满足用户不同的兴趣。但是一个句子在句子抽取中是否“有意义”是和用户无关的。

富含知识的句子抽取的关键是连接句子的知识和用户的需求。然而，这种需求对于不同的用户是不同的。而通常没有足够的数据来完全描绘一个人的特定需求。如示例7.1中所建议的，领域是自定义需求的关键特征。只有当抽取系统确定 $s_1$ 属于AI领域，它才会将其识别对于AI研究者的富含知识的句子。本文的目的是识别领域相关的富含知识的句子（DKS）。

为了使抽取方法能够处理领域需求，本章使用了领域问答语料库。在给定领域的QA语料库中，答案句子对于该领域肯定是富含知识的。系统学习答案句子的表示方法。如果某个句子具有类似的表示，它将被识别为DKS。

在学习一个领域的句子表示的时候，传统的模型包括主题模型和语言模型[90]。但是它们不能直接用于富含知识句子的提取。富含知识句子的抽取问题和它们主要有两个区别：（1）QA语料库的答案和句子在纯文本格式的表示不一样。一些元素通常在答案中会被省略。比如在示例7.1中，答案中省略了实验室的名称。因此，直接学习答案的表示通常会导致纯文本中的句子识别的更多错误。（2）当从纯文本学习句子表示时，其上下文是重要的特征。传统模型的着重于表示句子本身，而没有考虑它的上下文。

本文提出了一种数据驱动方法DAKSE，它包含无监督的种子DKS标记和有监督的DKS分类。DAKSE首先通过匹配纯文本句子和回答语句来标记种子DKS。这样就弥补了它们之间的差距。然后系统进一步使用神经网络来学习DKS分类器。该网络包含用于对候选语句和其上下文句子建模的三个并行的LSTM。

**应用：**富含知识的句子抽取的结果不止可以判定一个句子对于用户是否是富含知识的，对以下几个NLP任务也是有益的：

- **领域信息抽取** 开放信息抽取从给定语料库中提取所有结构化三元组。因此，如果开放信息抽取使用富含知识句子抽取系统抽取的句子，那么它就可以提取特定领

域的三元组。

- **问答系统** QA系统依赖大量的问答语料对进行训练 [21]。但现有的问答语料对是有限的。因此，从纯文本自动生成问答语料对可以丰富QA语料库。而富含知识的句子实际上就是问题的答案。自动问题生成技术[42]可以被进一步用来生成完整的QA对。
- **自动摘要** 文档摘要可以使用富含知识的句子提取作为预处理步骤，来识别有意义的句子。对领域知识的识别可以使得自动摘要更加定制化。

**贡献** 下面总结本章工作的贡献：

- 本章提出了富含知识的句子抽取的问题，并且分析了它的领域相关性。
- 本章提出了一种数据驱动方法DAKSE，用于富含知识的句子抽取问题。本章使用QA语料库中的答案为给定领域生成种子DKS。
- 本章在真实数据集上评估了DAKSE提取出的句子的效果，包括客户服务领域（中国移动客户服务）和百科全书领域（百度百科）。本章还将结果应用于领域信息的提取。

**本章结构** 本章的其余部分组织如下：首先概述了DAKSE的系统架构。接着，本章描述了DAKSE如何在预处理步骤中使用领域QA语料库来标记训练数据。然后详细阐述用于DKS分类的神经网络模型。最后展示了两个领域的实验结果，并将结果应用于领域信息提取。

## 第2节 相关工作及其不足

本章中的工作涉及几个相关主题，包括开放信息抽取，知识库中的关系抽取和句子抽取。

**开放信息抽取** 开放信息抽取系统使用自由关系[8]而不是预定义的模式从自然语言文本中抽取结构化信息。结果以(*Beijing, is the capital of, China*)的三元组形式展示。现在已经出现了许多经典的开放信息抽取系统。TextRunner[8]首先引入了开放信息抽取问题。Reverb[29]揭示了TextRunner的两个典型问题：不相关信息抽取和非信息抽取。Reverb通过添加句法约束来解决TextRunner的这两个典型错误。Stanford open IE[4]进一步利用句子的语言结构来解决了长程依赖问题。这些开放信息抽取系统在很少监督的情况下提取知识。但用户往往只关注一些特定的主题或领域。这些系统会给具有特定需求的用户带来很多无用的元组。

**关系抽取** 关系抽取问题是指从自然语言文本中学习实体关系。它们通常以有监督的方式学习，需要很多带标记的样本用于训练模型。[36]使用关系的自然语言模式从文本中提取新的关系。该方法的学习过程是迭代式的，在每次迭代中学习新的模式和新的关系。[70]使用强化学习来生成新查询，同时更新提取的值。在关系提取中，这些关系是预定义的（比如来自知识库）。但在本章的问题中，对每个用户的有意义的关系是未知的。

**句子抽取**[41]专注于从文档中提取“有意义的”句子。该方法主要用于文档摘要任务。[55]首先使用一个简单的贝叶斯分类器来提取句子和汇总文档。他们使用许多统计特征，如固定短语特征，大写字母特征来表示句子。自从那以后，很多方法被提出来解决这个问题，包括TF-IDF方法[35]，基于图的方法[53]，基于神经网络的方法[49]。这些句子提取方法从静态的角度考虑一个句子是否具有意义，“有意义”意味着一个句子可以总结文档。相反，本章认为某个句子是否有意义是动态的，它的意义取决于特定用户。也就是说，他们的方法没有考虑“领域”作为句子提取的特征。

### 第3节 系统概览

图7.1展示了DAKSE的系统架构。在种子DKS标注模块中，系统利用QA语料库标记种子DKS来进行进一步训练。通过使用这些种子DKS作为训练数据，系统构建了一个深层神经网络（DKS分类器）来学习这些种子DKS的表示。通过使用DKS分类器，DAKSE在纯文本语料库中提取更多的DKS。本节接下来会给出更多细节，同时给出一个示例（例7.2），展示DAKSE是如何工作的。

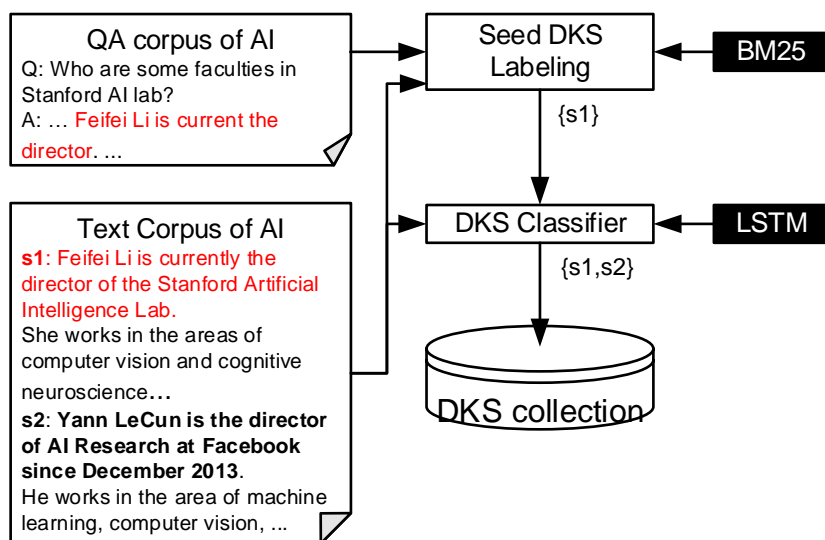


图 7.1: DAKSE 的系统架构

种子DKS标注是DAKSE识别种子DKS的预处理步骤。需要注意的是，在特定领域



的QA语料库中的答案，对于该领域用户是富含知识的。但是由于答案和纯文本之间的差距，DAKSE不直接使用这些答案作为种子DKS。因为答案通常会省略元素，这使得它们的表示不同于纯文本的表示。这里系统认为如果纯文本语料库中的某个句子与至少一个QA语料库中的答案具有高相似性，那么它就是DKS。通过这种方式，DAKSE通过QA语料库来标记种子DKS。然后在DKS分类器中学习这些种子DKS的表示。

**DKS分类器**通过使用种子DKS作为训练数据，DAKSE从纯文本中抽取的DKS学习一个深层神经网络。该模型将句子及其上下文句子视为特征。

**例 7.2.** 在图7.1, DAKSE试图从文本语料库中提取AI领域的DKS。首先，在种子DKS标记模块中，系统产生种子DKS。需要注意的是，直接使用答案作为种子DKS在这种情况下效果不好。因为 $s_2$ 与答案完全不同，很难根据答案将 $s_2$ 分类为DKS。DAKSE首先计算纯文本语句和答案之间的相似性。它通过识别出 $s_1$ 与答案具有高相似性，将 $s_1$ 标记为种子DKS。然后DAKSE学习 $s_1$ 的分类器，并使用分类器在文本语料库中提取新的DKS。由于 $s_1$ 和 $s_2$ 非常相似， $s_2$ 也被归类为DKS。注意这里上下文特征也有助于分类。

## 第4节 种子DKS标注

本节详细介绍种子DKS标注模块。该模块将纯文本语料库中的一些句子标记为种子DKS。这些种子DKS会进一步用于训练DKS分类器。

为了确定一个句子是否是DKS，DAKSE利用领域QA语料库。如前文所述，领域QA语料库中的答案是富含知识的。如果一个纯文本句子与至少一个领域QA语料库中的答案具有高相似性，DAKSE就认为它是种子DKS。这里相似度使用BM25[79]计算。

**用 BM25计算相似度** 给定一个纯文本语句  $s_1$ ，一个回答语句  $s_2$ ，和单词集合  $w_1, \dots, w_n$ ，它们的 BM25分数为：

$$\begin{aligned} &bm25(s_1, s_2) \\ &= \sum_{i=1}^n IDF(w_i) \cdot \frac{f(w_i, s_1) \cdot (k_1 + 1)}{f(w_i, s_1) + k_1 \cdot (1 - b + b \cdot \frac{|s_1|}{avgs_l})} \end{aligned} \quad (7.1)$$

这里  $IDF(w_i)$  是 $w_i$ 的逆文档频率权重(与该单词在QA语料中出现的文档个数有关),  $f(w_i, s_1)$  是  $w_i$ 's 在 $s_1$ 中的词频,  $|s_1|$  是 $s_1$ 的长度,  $avgs_l$ 是纯文本语料库的平均句子长度。 $k_1$  和  $b$  是超参数。它们分别被设置为常用值：1.5 和 0.75。

如果存在一个答案语句 $s_2$ ，使得 $bm25(s_1, s_2) > \delta$ ，那么系统标记这个纯文本语句 $s_1$ 为一个种子DKS。根据前人经验，这篇论文设置 $\delta = 0.4$ 。

## 第5节 DKS分类器

本节详细说明DKS分类器。DKS分类器判断一个纯文本语料中的句子是否是一个DKS。它将种子DKS视为正确标准并从中学习。分类器考虑目标语句本身和其两个上下文语句作为分类的特征。系统为三个句子构建三个具有类似结构的三个并行网络（一个嵌入层和一个LSTM层）。然后系统在输出层中聚合它们的输出来生成目标句子的总得分。

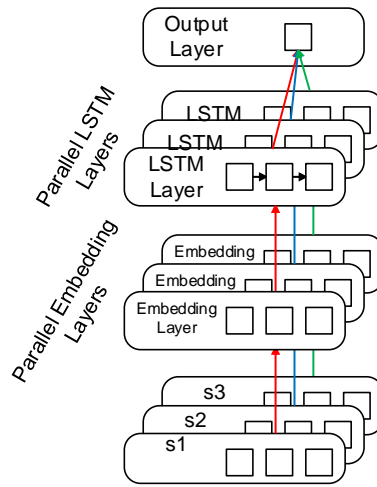


图 7.2: 为DKS分类器构建的神经网络

词向量层为分别为每个句子使用单独的嵌入矩阵。更正式的说，对于一个有前驱句子 $s_2$ 和后继句子 $s_3$ 的目标句子 $s_1$ ，句子 $s_i$ 中的单词 $w$ 使用词向量矩阵 $M_i$ 来做向量化：

$$e_w = M_i r(w), i = 1..3 \quad (7.2)$$

这里 $r(w) \in \{0, 1\}^{|V|}$ 是 $w$ 的one-hot表示,  $|V|$ 是词汇表大小。

**LSTM层** 词向量层的结果被传递到LSTM层，LSTM层用于对词序列的长依赖性进行建模。LSTM层由存储器单元序列组成，每个单元从嵌入层和前驱单元获得输入。存储器单元具有四个基本元件：输入门，忘记门，状态存储单元和输出门。首先，忘记门接收来自嵌入层和前驱单元的输入，并且决定丢弃哪个值。然后输入门决定更新哪个值。状态存储单元存储更新的值。最后，输出门决定要输出什么。

这里使用的LSTM版本来自于论文[108]。该模型记忆单元有如下复合函数：



$$\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\
j_t &= \sigma(W_{xj}x_t + W_{hj}h_{t-1} + b_j) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\
o_t &= \tanh(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\
c_t &= c_{t-1} \odot f_t + i_t \odot j_t \\
h_t &= \tanh(c_t) \odot o_t
\end{aligned} \tag{7.3}$$

这里 $\sigma$ 函数是一个sigmoid函数,  $i, f, o, c$  分别是输入门, 忘记门, 输出门, 单元激活向量,  $j$ 用于计算新的 $c$ 值,  $W, b$ 是LSTM层的参数。

**输出层:** 该层连接LSTM的结果, 并使用sigmoid函数来确定目标语句的得分, 来判断目标语句是否为DKS。

$$score(x_i; \theta) = \sigma(W_s[h_p, h_i, h_a] + b_s) \tag{7.4}$$

这里 $h_p, h_i, h_a$ 是LSTM层的三个输出,  $\sigma$ 是sigmoid函数,  $W_s, b_s$ 是该层的参数。

**模型训练** 训练过程使用种子DKS标记模块标记出的种子DKS作为正样本训练数据。模型将无意义的句子作为负样本训练数据。这些句子随机采样自中文小说。训练数据的更多细节可以在实验部分找到。模型使用二元交叉熵作为损失函数。令 $\mathcal{X} = \{x_1, \dots, x_n\}$ 为训练数据, 这样每个 $x_i$ 包括目标语句和它的上一句语句和下一句语句,  $\mathcal{Y} = \{y_1, y_2, \dots, y_n\}$ 是它们对应的标签(0 或者 1)。对于参数 $\theta$ ,  $x_i$ 在输出层对应的输出分数为 $score(x_i; \theta)$ 。如果 $score(x_i; \theta) \geq 0.5$ , 模型预测 $x_i$ 的标签 $\hat{f}(x_i; \theta) = 1$ , 否则为0。对于参数 $\theta$ , 对应的目标函数为:

$$L(\theta) = \sum_{i=1}^n -(y_i \log \hat{f}(x_i; \theta) + (1 - y_i) \log (1 - \hat{f}(x_i; \theta))) \tag{7.5}$$

至于详细的实现, 在训练过程使用随机参数初始化, 利用反向传播算法[80]来训练参数, 使用mini-batched AdaGrad[27]算法进行非凸优化, 相应的学习率为0.001, 词向量的维度设置为128, 每个LSTM层包含128个单元。

## 第6节 实验

### 6.1. 实验设置

所有的实验在一个双路Intel CPU E5-2620 v2 @ 2.10GHz, 128GB内存, GeForce GTX 980 GPU的机器上运行。

**数据集:** 实验在两个领域应用DAKSE: 中国移动客户服务和百度百科。对于每个领域, 实验首先通过种子DKS标记模块来标记种子DKS。这些DKS被认为是正样本。然

ID	DKS	类别
<i>dks<sub>1</sub></i>	You can guide customers to the corresponding service center in Shanghai to upgrade their softwares, and then the phone will not receive IMEI messages any more.	操作指导
<i>dks<sub>2</sub></i>	The XXT platform will inform trial users that the service will take fees soon, with the following message: “Dear Parents, the XXT service, which reports students’ academic performance, will start taking fees from next month. The fee is 10 yuan/month/person. Please reply QXJX to unsubscribe.”	短信提醒
<i>dks<sub>3</sub></i>	Three days before the appointed date, 10086 will send the first reminder message: “Shanghai Mobile: Your leasing business will be expired in XXXX (the date), the ID of rented phone is XXXXXXXX.”	事件描述
<i>dks<sub>4</sub></i>	For users who rent phones, we will send them two messages: the reminder to inform them the appoined date for return, and the reminder for renewing the service.	短信提醒
<i>dks<sub>5</sub></i>	Receiving text messages or multimedia messages normally do not take GPRS flow fees. But receiving multimedia messages abroad will take international roaming GPRS network flow fees.	服务介绍

表 7.1: 中国移动客服的前五5DKSs

后实验添加相等数量的非DKS作为负样本。这些非DKS是从中文小说中随机选择的句子，小说中的句子通常来说不包含知识。

- **中国移动客户服务** 包含中国移动公司的服务文档。纯文本语料库包含363354个句子。QA语料库包含9570个QA对，它们是来自中国移动呼叫中心的常见问题及其答案。
- **百度百科** 是最大的中文百科。本实验从百度百科上抓取了2074116个句子。实验使用百度知道<sup>1</sup>作为QA语料库，从中随机选取了 500000个中的QA对。很多百度知道中的问题是和百度百科中的知识相关的。

**基准实验：**实验使用两个基础的方法来和DAKSE做对比。

- **主题模型+SVM** 实验使用LDA来计算每个句子的主题分布。这些主题分布之后会被当作该句子的特征，用于SVM分类。实验将种子DKS标注模块得到的句子作为正样本，加入小说中的句子作为负样本。
- **语言模型** 实验在种子DKS上训练出一个语言模型[90]，对于一个新的句子，如果它的困惑度(Perplexity)小于一个给定的阈值，该模型认为它是一个DKS。

<sup>1</sup>zhidao.baidu.com

## 6.2. 中国移动客户服务

本小节评估DAKSE在中国移动客户服务语料上的效果。首先，实验给出提取的DKS的直观感受，然后评估抽取出来的DKS的有效性。

**DKS概览:** DAKSE从纯文本语料库提取总共63543个DKS。为了给出DKS的直观感受，表格7.1中列出了前5的DKS（由DKS输出层中的分数排序，该分数表现了分类的置信度）。实验发现了一些有趣的结果如下：

- DAKSE提取出了领域富含知识的句子。所有这些DKS都是客户服务领域的信息。
- DAKSE提取了一些非常复杂的知识。例如 $dk_{s_2}$ 包含消息提醒关系中的不同角色：发送者（XXT平台）、信息（服务将很快收费）、接受者等等。
- 抽取出来的DKS覆盖多种类型。

**实验结果的有效性:** 实验通过两种方式评估DAKSE是否可以很好地区分DKS和非DKS。首先，通过假设种子DKS是正确的，实验评估DKS分类器是否很好地提取出了DKS。实验选择80%的样本进行训练，其余的用于测试。训练过程从小说中添加相等数量的负样本。结果表示在表7.2中。

方法	精度	召回率	F1-值
主题模型 + SVM	0.98	0.86	0.92
语言模型	0.73	0.92	0.82
DAKSE	<b>0.99</b>	<b>0.97</b>	<b>0.98</b>

表 7.2: 中国移动客户服务结果评估

同时使用手工标注来评估DKS的有效性。实验随机选择100个DKS。并邀请来自相关领域的志愿者标记DKS是否包含目标领域的有效知识。所有DKS被手动分成三种类型：（1）正确 如果DKS是目标领域的有效知识；（2）不相关 如果DKS包含知识，但与目标领域不相关；（3）不正确 如果DKS没有包含知识信息。结果展示在表7.3中。

方法	正确率	不相关率	不正确率
主题模型 + SVM	52%	9%	39%
语言模型	56%	5%	39%
DAKSE	<b>68%</b>	3%	29%

表 7.3: 中国移动客户服务人工标签结果

表7.2和表7.3中的结果验证了DAKSE能够以很高的准确率和召回率抽取出DKS。这里手工标注的正确率略小于精度，因为表7.2中的正确标准中包含一些噪声（即一些标注为种子DKS的实际上是非DKS）。

### 6.3. 百度百科

本节评估DAKSE在中文百度百科上的效果。与中国移动的客户服​​务相比，百科领域更为宽泛。实验使用该语料库来展示DAKSE在更一般领域的有效性。

**DKS概览：**DAKSE抽取共1806239个DKS。表7.4列出了前5个DKS，用来对抽取结果提供一些直观的展示。实验发现DAKSE为百度百科领域提取出了有效的DKS。

ID	DKS
<i>dks<sub>6</sub></i>	Lower part of the middle member of the Wahweap Formation was formed in the Middle Campanian Age, almost between 79.9 million years and 80.6 million years from nowadays.
<i>dks<sub>7</sub></i>	Cindy went on the ninth season of an NBC reality TV show named The Celebrity Apprentice and donated all his prize money to the True Colors Fund.
<i>dks<sub>8</sub></i>	Ascidacea animals that are widely spread in our country include Leptoclinummitsukurii, Amarouciumconstellatum, Ascidialongistriata, Cionaintestinalis, Botryllusssp, Styelacanops, Molgulamanhattensis, Chelyosoma, and Botrylloidessimodensis.
<i>dks<sub>9</sub></i>	“Cifa” is short for China International Freight Forwarder Association.
<i>dks<sub>10</sub></i>	The merger between state-owned companies or between state-owned company and non-state-owned companies, the enterprise after merger is state-owned industrial company.

表 7.4: 百度百科前5的DKS

**实验结果的有效性：**类似于中国移动客户服​​务中的度量方法，假设种子DKS是正确的。评估结果展示在表7.5中。表7.6中展示了手动标记的结果，该结果验证了本章方法的有效性。

方法	精度	召回率	F1-值
主题模型 + SVM	0.98	0.86	0.91
语言模型	0.44	0.75	0.56
DAKSE	0.88	0.97	<b>0.92</b>

表 7.5: 百度百科的评估结果

方法	准确	不相关	不正确
主题模型 + SVM	69%	0%	31%
语言模型	65%	0%	35%
DAKSE	<b>80%</b>	0%	20%

表 7.6: 百度百科的人工标注结果

#### 6.4. 特征贡献

DAKSE将目标语句本身（Inner）和其上下文语句（Context）视为特征。实验评估了这些特征在百度百科上做DKS分类的贡献，结果展示在图7.3中。这里“Inner”指模型只使用目标句子作为特征。“Context”指模型只使用上下文作为特征。“Inner+Context”是本章提出的模型。

这些模型的召回率几乎是一样的，然而通过添加上下文特征，精度有所提高。该结果验证了这两个特征的贡献。

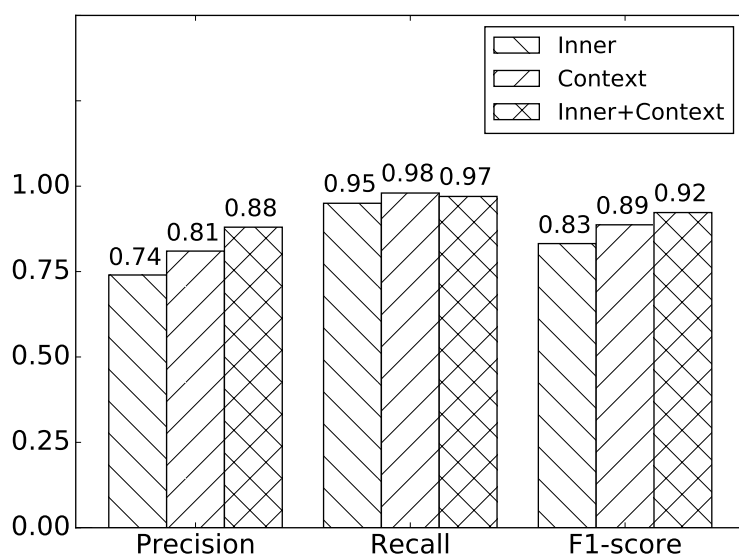


图 7.3: 特征贡献

#### 6.5. 应用：领域信息抽取

开放信息提取系统是从自然语言语料库提取所有结构化元组。因此，通过使用DKS作为语料库，可以实现特定领域的信息提取，所有提取的元组都属于该特定领域。

实验使用Stanford Open IE[4]进行信息提取。由人工来评估提取的元组是否正确。中国移动客户服务语料上抽取的元组精度为74%。百度百科语料上抽取的元组精

度为62%。

为了给出提取的元组的直观描述, 表格7.7展示了来自中国移动客户服务语料库的DKS的元组中频率前10的关系, 并将结果与原始文本语料库中抽取出的关系进行比较。结果表明, 根据DKS 抽取出的元组中的关系更加清晰。这些关系直接指向用户的需求 (例如*can visit link for*)。相比之下, 直接从原始语料库中抽取出的关系更模糊。

DKSs	can go through, can visit, contain, mainly contain, can visit link for, please send, reply to, welcome to, directly reply to, be divide into
original	will, can, have, see, can get, will receive, can participate in, can use, can enjoy, can apply for

表 7.7: 中国移动客服服务语料中的前10关系

提取的前几个DKS的元组展示在表7.8中。可以看出, 这些元组具有很高的质量并且与相应的领域相关。这些元组将是构建特定领域结构化知识库的良好数据源。这也验证了DAKSE有益于领域信息提取。

ID	Tuple		
	subject	relation	object
$dk_{s1}$	corresponding service center	is in	Shanghai
	you	upgrade	their softwares
$dk_{s2}$	service	take fees with	message
	service	take fees with	following message
	XXT platform	will inform	trial user
	fee	is	10 yuan/month/person
$dk_{s3}$	10086	will send reminder message before	three days
	10086	will send	reminder message
	XXT platform	will inform	trial user
	10086	will send	first reminder message
$dk_4$	we	will send	two messages
$dk_5$	roaming GPRS network flow fee	receive abroad	multimedia message

表 7.8: 中国移动客服语料抽取出的信息

## 第 7 节 小结

本章研究了针对特定领域的富含知识的句子提取的问题。在没有给定领域的预定义模式的情况下，本章利用领域QA语料库标记种子DKS，构建了DAKSE系统，实现了文本语料库中的DKS的自动识别。

本章在两个不同的领域进行了充分的实验，结果验证DAKSE能够有效地识别DKS。实验还将抽取出的结果应用于开放信息抽取技术上，提取出了特定领域的结构化的知识。





## 第八章 基于Freebase的KBQA问答系统展示

作为本文成果系统性展示，本章介绍了基于Freebase的KBQA系统，直观性展示本文的综合成果。该系统的回答过程中集成了本文中的多项核心技术。其具有若干优秀特性，包括多类型问题回答、答题过程可解释性、允许用户反馈等。基于Freebase的KBQA系统印证了本文系统性研究的有效性。

### 第1节 架构

图8.1展示了KBQA的架构。线下过程学习从模板到属性的映射。这主要是通过使用最大似然估计器在模板提取模块中实现的。这样的估计器由实体 - 值标识的结果训练。此外，KBQA通过属性扩展模块理解知识图谱中的复杂属性形式（例如 *marriage*  $\rightarrow$  *person*  $\rightarrow$  *name* 在指的是“spouse”）。当一个问题进入线上过程，KBQA首先将其解析为一个或多个BFQ问题。对于每个BFQ问题，KBQA提取其模板并从模板库中查找属性。最后KBQA返回实体的值以及在知识库中检索到的属性为回答。

#### 1.1. 线上过程

**问题解析** 系统通过模板来理解问题。例如，How many people are there in \$city?是一个模板。无论\$city是指檀香山或者其他城市，模板问的总是population。为了将问题转换为模板，KBQA通过Stanford NER [33]识别问题的实体，并通过概念化[87]用概念替换实体。概念化机制基于由大量实体和概念组成的语义网络 Probase[103]。它保证了KBQA表示多种问题。

此外，为了回答一个复杂的问题，KBQA解析原始问题，并获得若干二元事实型问题。例如，对于问题 When was Barack Obama's wife born, KBQA将其解析为 Barack Obama's wife (Michelle Obama) 和 When was Michelle Obama born。这样一来，通过顺序地回答这些BFQ问题，可以得到原始问题的答案。为了找到这样的BFQ序列，KBQA枚举所有可能的分解并选择最可能的分解。当所有的BFQ问题都是有效的时候，KBQA认为此分解是有效的。而单个BFQ问题的有效性是通过QA语料库中与该BFQ问题具有相同模板的问题的数量度量的。换句话说，有效的BFQ问题模板应该被人类用户（在QA语料库中）频繁地询问。

**属性查找** 得到问题的模板后，KBQA在模板库中查找模板的属性。该属性是在线下过程学习到的。接下来的步骤很直接：KBQA在知识图谱中检索实体和属性对应的

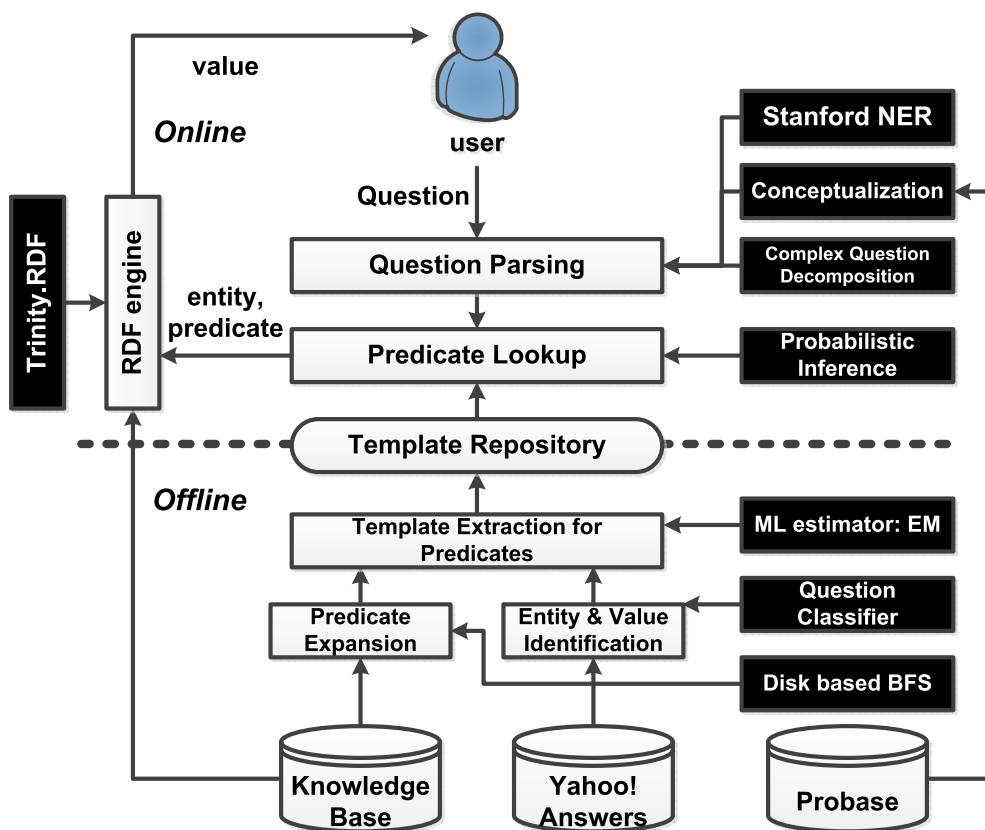


图 8.1: 系统架构

值，并将其返回给用户。

## 1.2. 线下过程

**模板提取** KBQA通过Yahoo! Answers学习模板与知识图谱中属性的映射。首先，对于Yahoo! Answers中的每个问答对，KBQA提取问题中的实体和相应的答案。本章在下面的实体-值识别模块中展示其细节。然后，KBQA在知识图谱中的实体和值之间查找属性。模板抽取的基本思想是，KBQA将Yahoo! Answers中的问答对作为训练数据。对于每个模板，如果QA语料库中的大多数问题实例有同样的属性，KBQA会将模板映射到此属性。例如，不管\$city具体指代哪个城市，模板how many people are there in \$city? 导出的问题总是映射到属性population。由此KBQA学习出这个模板映射到属性population。

**实体-值识别** 此模块从QA语料库中提取问题中的实体和答案的值。与传统的NER方法不同，KBQA使用答案来帮助识别。其思路是，(1) 有效的实体-值对在知识图谱中具有某种关系；(2) 正确的值和问题应该有相同的类别。KBQA通过(1)在知识图谱中查找候选实体-值对(2)使用问题分类器[66]对齐问题和值的类别。

**属性扩展** 在知识图谱中，许多事物之间的关系不是由一个属性直接表示的，而

是由一系列属性组成的路径表示的。如图1.1所示，*spouse of*由知识图谱中的三个属性 $marriage \rightarrow person \rightarrow name$ 来表示。这些多属性路径被称为扩展属性。回答扩展属性上的问题大大提高了KBQA的覆盖率。为此，KBQA需要生成所有有效的（实体，扩展属性，值）三元组。对于具有1.1TB磁盘大小的知识图谱，KBQA使用基于磁盘的多源BFS算法来节省内存。它扫描磁盘上的知识图谱 $k$ 次，其中 $k$ 是扩展属性的最长长度限制。在开始时，KBQA将出现在QA语料库中的所有实体加载到内存中。在第一轮中，通过扫描驻留在磁盘上的所有RDF三元组并加入先前的实体，KBQA获得所有具有属性长度1的（实体，扩展属性，值）三元组。在第二轮中，KBQA将到目前为止找到的所有三元组加载到内存中。它再次扫描RDF并与所有之前的实体连接。现在，KBQA获得所有具有属性长度2的SPO三元组。KBQA重复上述操作 $k$ 次。由于BFS从QA语料库中的所有实体开始，该过程将生成所有有效的三元组。在实际系统中设置 $k = 3$ 。

## 第2节 展示

KBQA在网上为用户提供了一个直接的网页交互平台。网页包括三个部分：（1）问答部分显示主要结果；（2）反馈部分允许用户对回答投票；（3）解释部分展示答案是如何被抽取的，以及为什么KBQA可以提取答案。图8.2展示了一个例子。下面将详细说明每个部分。

### 2.1. 多种问题类别

QA系统的主要交互功能很简单：用户只需要提交一个自然语言问题，即可得到答案。在这种直接的交互方式背后，KBQA实际上满足了用户不同类型的需求。下面解释KBQA如何回答用户提出的不同类型的问题。

- **简单BFQ问题** 用户可以询问简单BFQ问题，例如有关一个实体的属性的问题。图8.2展示了关于莎士比亚生日的问题。
- **依赖于扩展属性的问题** 属性扩展模块使KBQA能够理解扩展属性。因此，用户可以提出依赖于扩展属性的问题。图8.3展示了关于*spouse*的问题，其在知识图谱中由路径 $marriage \rightarrow person \rightarrow name$ 表示。
- **复杂问题** 问题解析模块将一个复杂问题分解为多个BFQ问题。这使KBQA能够理解和回答复杂的问题。图8.4展示了一个复杂的问题。KBQA将它分解为两个问题：*author of harry potter (J. K. Rowling)*，和*what are the books written by J. K. Rowling*。通过顺序回答这两个问题，KBQA成功地返回了答案。

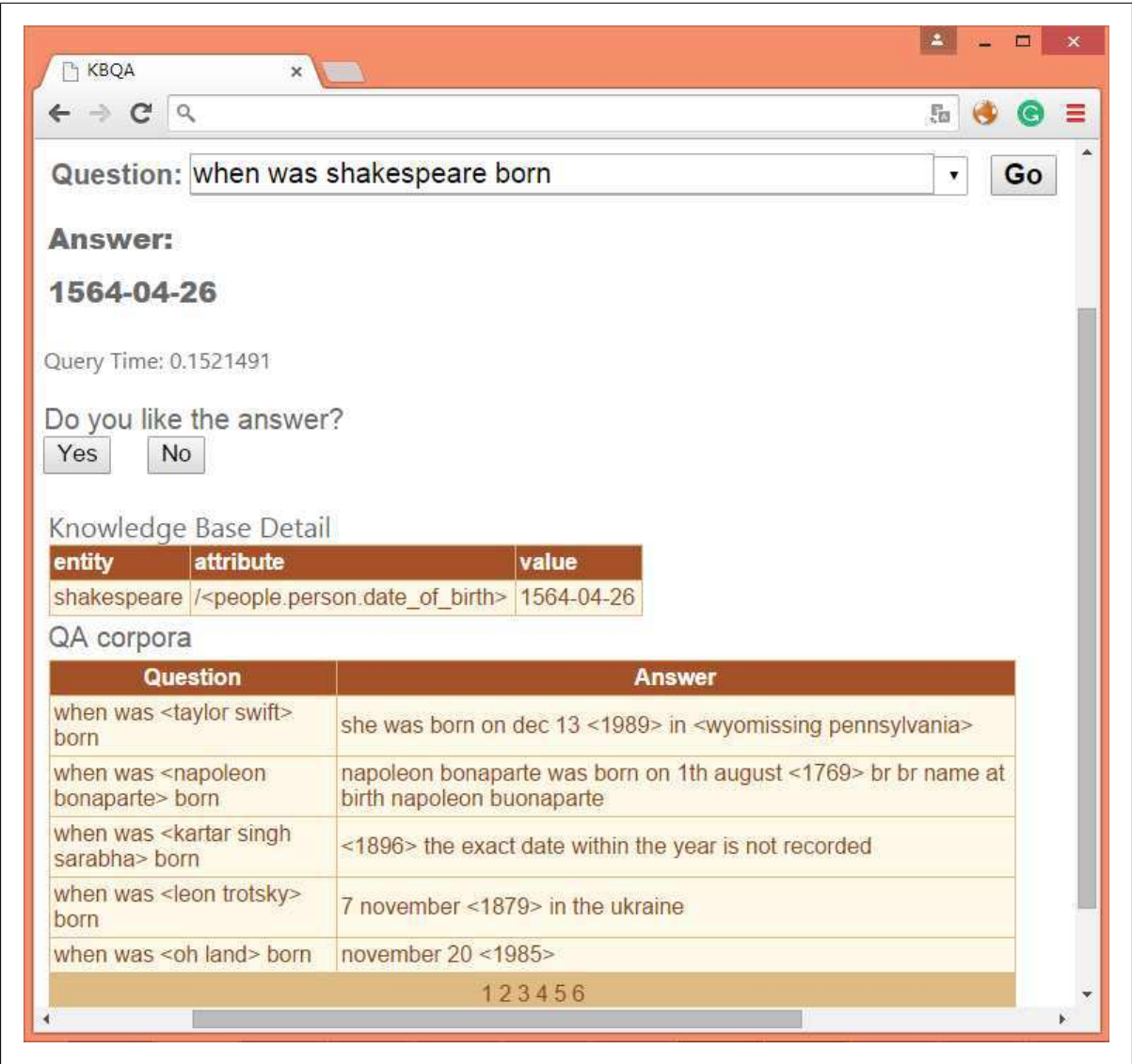


图 8.2: KBQA 回答简单 BFQ问题

2.2. 可解释性

如图8.2所示，KBQA答题的过程是可解释的，表中显示了整个QA过程的详细信息。可解释性具体表现在在几个方面：

- 知识图谱对问题中知识的表示是可见的。在图8.4的例子中，实体和值是通过知识图谱中的属性people.person.date\_of\_birth相连接的。
- 问题回答的过程是可见的。在图8.4的例子中，KBQA标识问题的实体Shakespeare。然后找到属性“when was \$person born” 的模板，并得到people.person.date\_of\_birth。所以最后KBQA 能从知识图谱中检索正确的

**Question:**  Go

**Answer:**

**Michelle Obama**

Query Time: 0.2442339

图 8.3: KBQA回答依赖于扩展属性的问题

值。

- 从QA语料库中学习模板的依据是可见的。在图8.4中，*people.person.date\_of\_birth*是正确的属性。KBQA展示了属性是如何从QA语料库中被学习的。QA语料库包含许多模板也是*When was \$person born*的问答对。由于这些问答对中大多数值指向population，KBQA能够正确学习映射。

### 2.3. 用户反馈

在反馈部分，KBQA允许用户能够投票回答。KBQA的反馈作为输入反过来被用于改善系统。为此，对于每个模板，KBQA使用相应的反馈来校正其属性。一旦模板接收到No，KBQA减少其当前属性的权重。如果反馈是Yes，KBQA增加相应权重。因此，对于一个模板和一个属性之间的错误映射，KBQA意图减少其权重，并且用其它属性来替换该属性。

实验进一步分析了模板的类别分布，如表8.1所示。可以看出，KBQA学习了不同类别的模板。模板的多样性支持了KBQA回答不同类别的问题。

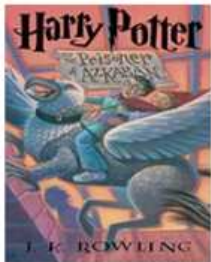


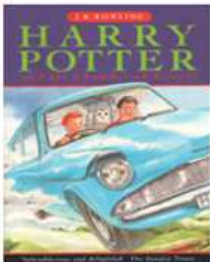
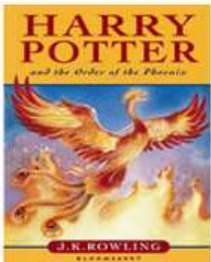
类别	日期	实体	位置	人	数字
比例	39.4%	59.0%	13.8%	22.7%	0.6%

表 8.1: 模板的种类分布



**Question:**

**Answer:**

				
<b>Harry Potter and the Prisoner of Azkaban</b>	<b>The Tales of Beedle the Bard</b>	<b>Harry Potter and the Half- Blood Prince</b>	<b>Harry Potter and the Chamber of Secrets</b>	<b>Harry Potter and the Order of the Phoenix</b>

Query Time: 0.0910091

图 8.4: KBQA 回答复杂问题



## 第九章 总结和展望

问答系统是当前工业界和学术界的热点。近年来的知识图谱、深度学习等相关技术突破，给了问答系统新的提升。但是当前问答系统依然距离人的要求有很大差距。在此总结本文的工作，讨论现有方法的局限性，并展望其前进方向和空间。

### 第1节 研究总结

基于知识图谱的问答系统是一个复杂的系统性工作。一个优秀的问题系统既要有底层的语义理解，又要做好上层的系统支持。本文系统论述了基于知识图谱的问答系统的关键工作，包括实体语义、短文本语义、问题语义等语义层面的研究，以及具体领域的应用适配。本文对每一模块都进行了深入而具体的研究。主要包括如下内容：

- **实体语义** 研究了语义社团搜索问题，语义社团为问题实体理解提供了丰富的关联语义信息。本文提出了基于最小度的语义社团度量，并在此度量下进行了深入的理论研究和算法分析。
- **动词语义** 研究了短文本语义的核心：动词理解。提出了使用动词模板来表示同一动词的不同语义。本文使用最小描述长度对问题进行建模，并证明该模型对于动词模板一般性和特殊性的体现。本文还将动词模板应用在上下文相关的实体概念化问题上。
- **问题语义** 讨论并指出了传统问题表示模型的语义局限性，提出了问题的模板表示法。利用动词模板，将同一语义的问题进行归类，并将不同语义的问题进行区分。
- **语义关联** 讨论了利用问答社区的问答语料进行语义关联学习的可能性，用概率图模型表示整个问答过程中的不确定性，并使用最大似然估计和EM算法进行概率估计。
- **领域适配** 研究了开放领域问答系统在领域适配中的两个核心问题：自然语言处理模型的迁移学习，以及领域知识点挖掘。对于前者，本文使用深度神经网络的模型。对于后者，本文使用领域问答语料进行学习，并提出数据驱动的方法实现自动化知识点抽取。

## 第2节 研究展望

当前的问答系统距离完全实用依然存在许多问题。从回答有效性的角度，系统需要在准确率和召回率上做提升；从应用实用性的角度，系统需要适配不同领域。这些不同角度的系统提升，需要针对性的提出方案。本文认为，系统准确率的提升上，核心手段是常识的引入；召回率的提升上，核心手段是引入文本描述作为额外知识源，利用信息检索和特征工程的方法，统筹考虑文本语义信息和知识图谱知识；领域应用的角度上，系统需要进一步做好适配工作。下面本文会对这三点的未来研究设想作进一步阐述。

### 2.1. 常识引入

Ernie Davis在1998年的《常识1998年题目页》中，提出了一个经典的常识问题：

描述：一个厨师正在将一个鸡蛋敲在一个碗上。在正确的操作后，鸡蛋加在碗边缘的力会导致鸡蛋壳碎成两半。将鸡蛋盛在碗里后，厨师将...

问题：...(q<sub>1</sub>) 碗是由活页纸制造的还是由软粘土制造的？(q<sub>2</sub>) 鸡蛋比碗大还是比碗小？ ...

很显然，人很容易在阅读完描述之后回答这些问题。问题q<sub>1</sub>的答案取决于鸡蛋和碗的硬度。而描述鸡蛋加在碗边缘的力会导致鸡蛋壳碎成两半则表明碗的硬度比鸡蛋高。问题q<sub>2</sub>的答案取决于碗和鸡蛋的大小。而描述将鸡蛋盛在碗里后则表明碗的尺寸比鸡蛋大。

人可以很容易的从文本描述中推断出鸡蛋和碗的硬度及尺寸的关系。这一理解和推断来自于人的常识。这种潜在的常识可以导致两个的现象：（1）同样的上下文可以对不同的实体对推断出相同的实体关系。例如将entity<sub>1</sub>盛在entity<sub>2</sub>里后总是表明entity<sub>2</sub>具有更大的尺寸。（2）不仅仅是以上的文本，很多其它的描述文本中，人也可以利用常识推断实体的关系。

所以，如何让计算机可以像人一样从自然语言中推理出实体关系？如何显式的描述这些实体关系，并使计算机可以直接使用？这一方向的问题研究，将加强问答系统的准确率和推理能力。

### 2.2. 领域适配

问答的具体落地在于领域，而当前大量的问答系统研究是基于开放领域的。本文第六章和第七章已经对如何做好问答系统的领域适配进行了大量的研究，包括如何将开放领域的自然模型迁移到具体领域中，以及如何挖掘领域知识，并利用open IE技术进行知识结构化和三元组化。然而距离完全实现领域适配，依然有大量的工作需要解决。其核心在于领域知识的关联化与语义化。

**知识关联化** 知识的本质在于关联。在知识图谱中，一个实体不止作为一些三元组的主语存在，同时也作为另一些三元组的宾语存在。这样就实现了图谱的关联性表示。类似的，需要对领域知识图谱作相关关联性分析。然而当前直接使用open IE抽取的结果仅抽取了主谓宾三元组，忽视了多个主谓宾三元组的关联，特别是不同主语、宾语间的关联。因此，需要实现领域知识关联化。

**知识语义化** 利用open IE抽取知识的本质，是对句子主谓宾的抽取。其核心语义在于谓词。然而，同一语义的谓词具有多种表现形式。直接利用谓词不代表三元组就具备了语义关系。例如，“嫁给”和“结为夫妻”这两个谓词具有相同的语义。对相同语义的谓词做聚类，有助于理解知识的语义信息，解决知识抽取的稀疏化、碎片化问题。

### 2.3. 文本描述+知识图谱

尽管当前的知识图谱规模可以达到十亿甚至更多，其知识量对于问答来讲依然具有巨大的局限性。这一局限性主要体现在以下两点：

**知识图谱的属性缺失** 知识图谱一般通过人工结构定义的方式，预先定义要表示的不同类型的知识。一般来说，一个大规模知识图谱的不同属性个数会达到上千种。然而人工定义的知识描述类别总是有限的，而在不同场景、不同领域中的问题类型个数则是无法预估的。例如，“谁是Beyond乐队的头面人物？”中的属性“头面人物”，不存在于包括Freebase、Yago2等在内的常见知识图谱。因此知识图谱有限的关系表示，无法覆盖来自用户的多样性知识需求。

**知识图谱的值缺失** 即便对于知识图谱已有的属性，其不完备性依然非常明显：对于某一类实体的某一属性，大部分该类别实体的该属性值是缺失的。例如在Freebase中，著名的Beyond乐队头面人物“黄家驹”的相关父母信息都是不存在的。这样“黄家驹的父母是谁”即无法返回答案。这导致即使问答系统正确理解了问题，将其转换为正确的格式化查询后，依然无法检索到回答。

另一方面，文本语料中描述了大量的实体相关信息，为知识图谱的不完备性提供了数据补充。例如可以从文本描述“作为Beyond的头面人物，黄家驹生于1962年”中，得到问题“谁是Beyond乐队的头面人物？”的答案。即文本描述是知识图谱的有效补充。同时，文本描述对于提高问题与知识图谱的语义解析（semantic parsing）有着重要意义。例如当发现知识图谱中的“父母”属性，总是可以被“父亲”、“父子”等自然语言中的其他方式描述时，系统就可以回答“谁和黄家驹是父子关系”这样的问题。

文本信息对知识图谱的数据补充和匹配效果提升，促使系统综合使用文本信息与知识图谱作为知识来源，实现更优质的问答。



## 参考文献

- [1] Eugene Agichtein, Silviu Cucerzan, and Eric Brill. Analysis of factoid questions for effective relation extraction. In *SIGIR*, pages 567–568, 2005.
- [2] Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. Domain-adversarial neural networks. *arXiv preprint arXiv:1412.4446*, 2014.
- [3] Ion Androutsopoulos, Graeme D Ritchie, and Peter Thanisch. Natural language interfaces to databases—an introduction. *Natural language engineering*, 1(01):29–81, 1995.
- [4] Gabor Angeli, Melvin Johnson Premkumar, and Christopher D Manning. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2015.
- [5] Chidanand Apté, Fred Damerau, and Sholom M Weiss. Automated learning of decision rules for text categorization. *TOIS*, 12(3):233–251, 1994.
- [6] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. 2007.
- [7] Collin F Baker, Charles J Fillmore, and John B Lowe. The berkeley framenet project. In *COLING*, volume 1, pages 86–90, 1998.
- [8] Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. Open information extraction from the web. In *IJCAI*, volume 7, pages 2670–2676, 2007.
- [9] Andrew Barron, Jorma Rissanen, and Bin Yu. The minimum description length principle in coding and modeling. *Information Theory, IEEE Transactions on*, 44(6):2743–2760, 1998.
- [10] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*, pages 1247–1250, 2008.

- [11] Béla Bollobás. The evolution of sparse graphs. *Graph theory and combinatorics*, pages 35–57, 1984.
- [12] John Burger, Claire Cardie, Vinay Chaudhri, Robert Gaizauskas, Sanda Harabagiu, David Israel, Christian Jacquemin, Chin-Yew Lin, Steve Maiorano, George Miller, et al. Issues, tasks and program structures to roadmap research in question & answering (q&a). In *DUC Roadmapping Documents*, pages 1–35, 2001.
- [13] Qingqing Cai and Alexander Yates. Large-scale Semantic Parsing via Schema Matching and Lexicon Extension. In *ACL*, pages 423–433, 2013.
- [14] Qingqing Cai and Alexander Yates. Semantic parsing freebase: Towards open-domain semantic parsing. *Atlanta, Georgia, USA*, page 328, 2013.
- [15] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka Jr, and Tom M Mitchell. Toward an architecture for never-ending language learning. In *AAAI*, volume 5, page 3, 2010.
- [16] Danqi Chen and Christopher D Manning. A fast and accurate dependency parser using neural networks. In *EMNLP*, pages 740–750, 2014.
- [17] Laura Chiticariu, Rajasekar Krishnamurthy, Yunyao Li, Frederick Reiss, and Shivakumar Vaithyanathan. Domain adaptation of rule-based annotators for named-entity recognition tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1002–1012, 2010.
- [18] Edgar F Codd. *Seven steps to rendezvous with the casual user*. IBM Thomas J. Watson Research Division, 1974.
- [19] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167, 2008.
- [20] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.
- [21] Wanyun Cui, Yanghua Xiao, Haixun Wang, Yangqiu Song, Seung-won Hwang, and Wei Wang. Kbqa: Learning question answering over qa corpora and knowledge bases. *Proceedings of the VLDB Endowment*, 10(5), 2017.

- [22] Hoa Trang Dang, Diane Kelly, and Jimmy J Lin. Overview of the trec 2007 question answering track. In *TREC*, volume 7, page 63, 2007.
- [23] Hal Daumé III. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*, 2009.
- [24] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.
- [25] Sergey N Dorogovtsev, Alexander V Goltsev, and Jose Ferreira F Mendes. K-core organization of complex networks. *Physical review letters*, 96(4):040601, 2006.
- [26] Cícero Nogueira dos Santos and Maira Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In *COLING*, pages 69–78, 2014.
- [27] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for on-line learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.
- [28] Ralf Engbert, Antje Nuthmann, Eike M Richter, and Reinhold Kliegl. Swift: a dynamical model of saccade generation during reading. *Psychological review*, 112(4):777, 2005.
- [29] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, 2011.
- [30] Fernanda Ferreira and John M Henderson. Use of verb information in syntactic parsing: evidence from eye movements and word-by-word self-paced reading. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 16(4):555, 1990.
- [31] David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. Building watson: An overview of the deepqa project. *AI magazine*, pages 59–79, 2010.
- [32] Charles J Fillmore. Frame semantics and the nature of language. In *Annals of the New York Academy of Sciences: Conference on the origin and development of language and speech*, volume 280, pages 20–32, 1976.



- [33] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*, pages 363–370, 2005.
- [34] Marco Gaertler and Maurizio Patrignani. Dynamic analysis of the autonomous system graph. In *International Workshop on Inter-domain Performance and Simulation*, pages 13–24, 2004.
- [35] René Arnulfo García-Hernández and Yulia Ledeneva. Word sequence models for single text summarization. In *Advances in Computer-Human Interactions, 2009. ACHI'09. Second International Conferences on*, pages 44–48, 2009.
- [36] Daniel Gerber and A-C Ngonga Ngomo. Bootstrapping the linked data web. In *1st Workshop on Web Scale Knowledge Extraction@ ISWC*, volume 2011, 2011.
- [37] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 513–520, 2011.
- [38] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1:12, 2009.
- [39] Travis R Goodwin and Sanda M Harabagiu. Medical question answering for clinical decision support. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 297–306. ACM, 2016.
- [40] Bert F Green Jr, Alice K Wolf, Carol Chomsky, and Kenneth Laughery. Baseball: an automatic question-answerer. In *Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference*, pages 219–224. ACM, 1961.
- [41] Vishal Gupta and Gurpreet Singh Lehal. A survey of text summarization extractive techniques. *Journal of emerging technologies in web intelligence*, 2(3):258–268, 2010.
- [42] Michael Heilman. *Automatic Factual Question Generation from Text*. PhD thesis, 2011.
- [43] Gary G Hendrix, Earl D Sacerdoti, Daniel Sagalowicz, and Jonathan Slocum. Developing a natural language interface to complex data. *ACM Transactions on Database Systems (TODS)*, 3(2):105–147, 1978.
- [44] Lynette Hirschman and Robert Gaizauskas. Natural language question answering: The view from here. *Natural Language Engineering*, pages 275–300, 2001.

- [45] Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, Edwin Lewis-Kelham, Gerard De Melo, and Gerhard Weikum. Yago2: exploring and querying world knowledge in time, space, context, and many languages. In *WWW*, pages 229–232, 2011.
- [46] Wen Hua, Zhongyuan Wang, Haixun Wang, Kai Zheng, and Xiaofang Zhou. Short text understanding through lexical-semantic analysis. In *2015 IEEE 31st International Conference on Data Engineering*, pages 495–506, 2015.
- [47] Ozan Irsoy and Claire Cardie. Deep recursive neural networks for compositionality in language. In *Advances in Neural Information Processing Systems*, pages 2096–2104, 2014.
- [48] Jing Jiang and ChengXiang Zhai. Instance weighting for domain adaptation in nlp. In *ACL*, volume 7, pages 264–271, 2007.
- [49] Khosrow Kaikhah. Automatic text summarization with neural networks. In *Intelligent Systems, 2004. Proceedings. 2004 2nd International IEEE Conference*, volume 1, pages 40–44, 2004.
- [50] Dongwoo Kim, Haixun Wang, and Alice Oh. Context-dependent conceptualization. In *IJCAI*, pages 2654–2661, 2013.
- [51] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.
- [52] Paul Kingsbury and Martha Palmer. From treebank to propbank. In *LREC*, 2002.
- [53] Canasai Kruengkrai and Chuleerat Jaruskulchai. Generic text summarization using local and global properties of sentences. In *Web Intelligence, 2003. WI 2003. Proceedings. IEEE/WIC International Conference on*, pages 201–206, 2003.
- [54] Henry Ku, Winthrop Nelson Francis, et al. Computational analysis of present-day {A}merican {E}nglish. 1967.
- [55] Julian Kupiec, Jan Pedersen, and Francine Chen. A trainable document summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 68–73, 1995.
- [56] Cody Kwok, Oren Etzioni, and Daniel S Weld. Scaling question answering to the web. *TOIS*, pages 242–262, 2001.

- [57] Thomas K Landauer and Susan T Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211, 1997.
- [58] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196, 2014.
- [59] Gary Geunbae Lee, Jungyun Seo, Seungwoo Lee, Hanmin Jung, Bong hyun Cho, Changki Lee, Byung-Kwan Kwak, Jeongwon Cha, Dongseok Kim, JooHui An, Hark-soo Kim, and Kyungsun Kim. Siteq: Engineering high performance qa system using lexico-semantic pattern matching and shallow nlp. In *In Proceedings of the Tenth Text REtrieval Conference (TREC)*, pages 442–451, 2001.
- [60] Beth Levin. *English verb classes and alternations: A preliminary investigation*. University of Chicago press, 1993.
- [61] Xin Li and Dan Roth. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7, 2002.
- [62] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Modelling interaction of sentence pair with coupled-lstms. *CoRR*, abs/1605.05573, 2016.
- [63] Vanessa Lopez, Michele Pasin, and Enrico Motta. Aqualog: An ontology-portable question answering system for the semantic web. In *European Semantic Web Conference*, pages 546–562, 2005.
- [64] Thang Luong, Richard Socher, and Christopher D Manning. Better word representations with recursive neural networks for morphology. In *CoNLL*, pages 104–113, 2013.
- [65] George W McConkie, Paul W Kerr, Michael D Reddix, and David Zola. Eye movement control during reading: I. the location of initial eye fixations on words. *Vision research*, 28(10):1107–1118, 1988.
- [66] Donald Metzler and W Bruce Croft. Analysis of statistical question classification for fact-based questions. *IR*, pages 481–504, 2005.
- [67] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

- [68] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.
- [69] Alan Mislove and et al. Measurement and analysis of online social networks. In *IMC'07*.
- [70] Karthik Narasimhan, Adam Yala, and Regina Barzilay. Improving information extraction by acquiring external evidence with reinforcement learning. *arXiv preprint arXiv:1603.07954*, 2016.
- [71] M. E. J. Newman and et al. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69(2):026113.
- [72] Shiyan Ou, Constantin Orasan, Dalila Mekhaldi, and Laura Hasler. Automatic question pattern generation for ontology-based question answering. In *FLAIRS*, pages 183–188, 2008.
- [73] Martha Palmer. Semlink: Linking propbank, verbnet and framenet. In *Proceedings of the Generative Lexicon Conference*, pages 9–15, 2009.
- [74] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.
- [75] Hao Peng, Lili Mou, Ge Li, Yunchuan Chen, Yangyang Lu, and Zhi Jin. A comparative study on regularization strategies for embedding-based neural networks. In *Empirical Methods in Natural Language Processing*, 2015.
- [76] Xipeng Qiu and Xuanjing Huang. Convolutional neural tensor network architecture for community-based question answering. In *Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI'15*, pages 1305–1311, 2015.
- [77] Malka Rappaport Hovav and Beth Levin. Building verb meanings. *The projection of arguments: Lexical and compositional factors*, pages 97–134, 1998.
- [78] Deepak Ravichandran and Eduard Hovy. Learning surface text patterns for a question answering system. In *ACL*, pages 41–47, 2002.
- [79] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. Okapi at trec-3. *NIST SPECIAL PUBLICATION SP*, 109:109, 1995.
- [80] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

- [81] Denis Savenkov. Ranking answers and web passages for non-factoid question answering: Emory university at trec liveqa. In *TREC*, 2015.
- [82] Karin Kipper Schuler. Verbnets: A broad-coverage, comprehensive verb lexicon. 2005.
- [83] Stephen B Seidman. Network structure and minimum degree. *Social networks*, 5(3):269–287, 1983.
- [84] John Sinclair. *Corpus, concordance, collocation*. Oxford University Press, 1991.
- [85] Richard Socher, Eric H Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Y Ng. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *NIPS*, pages 801–809, 2011.
- [86] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642, 2013.
- [87] Yangqiu Song, Haixun Wang, Zhongyuan Wang, Hongsong Li, and Weizhu Chen. Short text conceptualization using a probabilistic knowledgebase. In *IJCAI*, pages 2330–2336, 2011.
- [88] Mauro Sozio and et al. The community-search problem and how to plan a successful cocktail party. In *KDD’10*.
- [89] Michael Speriosu, Nikita Sudan, Sid Upadhyay, and Jason Baldrige. Twitter polarity classification with label propagation over lexical links and the follower graph. In *Proceedings of the First workshop on Unsupervised Learning in NLP*, pages 53–63, 2011.
- [90] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. Lstm neural networks for language modeling. In *Interspeech*, pages 194–197, 2012.
- [91] Kai Sheng et al. Tai. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*, 2015.
- [92] Erik F Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *CoNLL*, pages 142–147, 2003.
- [93] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the*

- 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, NAACL '03, pages 173–180, 2003.
- [94] William Tunstall-Pedoe. True knowledge: Open-domain question answering using structured knowledge and inference. *AI Magazine*, pages 80–92, 2010.
- [95] C Unger, P Cimiano, V Lopez, and E Motta. Qald-1 open challenge, 2011.
- [96] Christina Unger. Qald-3 open challenge. 2013.
- [97] Christina Unger, Lorenz Bühmann, Jens Lehmann, Axel-Cyrille Ngonga Ngomo, Daniel Gerber, and Philipp Cimiano. Template-based question answering over rdf data. In *Proceedings of the 21st international conference on World Wide Web*, pages 639–648, 2012.
- [98] Christina Unger and Philipp Cimiano. Pythia: Compositional meaning construction for ontology-based question answering on the semantic web. In *NLDB*, pages 153–160. 2011.
- [99] Christina Unger, Forascu Corina, Lopez Vanessa, Ngomo Axel-Cyrille, Ngonga, Cabrio Elena, Cimiano Philipp, and Walter Sebastian. Question answering over linked data (qald-5). 2013.
- [100] David L Waltz. An english language question answering system for a large relational database. *Communications of the ACM*, 21(7):526–539, 1978.
- [101] William A Woods and R Kaplan. Lunar rocks in natural english: Explorations in natural language question answering. *Linguistic structures processing*, 5:521–569, 1977.
- [102] William A Woods, Ronald M Kaplan, and Bonnie Nash-Webber. *The Lunar Sciences: Natural Language Information System: Final Report*. Bolt Beranek and Newman, 1972.
- [103] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. Probase: A probabilistic taxonomy for text understanding. In *SIGMOD*, pages 481–492, 2012.
- [104] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *ACL*, pages 133–138, 1994.
- [105] Chen Xinxiong, Xu Lei, Liu Zhiyuan, Sun Maosong, and Luan Huanbo. Joint learning of character and word embeddings. In *IJCAI*, 2015.

- [106] Kun Xu, Sheng Zhang, Yansong Feng, and Dongyan Zhao. Answering natural language questions via phrasal semantic parsing. In *Natural Language Processing and Chinese Computing*, pages 333–344. 2014.
- [107] Mohamed Yahya, Klaus Berberich, Shady Elbassuoni, Maya Ramanath, Volker Tresp, and Gerhard Weikum. Natural language questions for the web of data. In *EMNLP*, pages 379–390, 2012.
- [108] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.
- [109] Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. Distant supervision for relation extraction via piecewise convolutional neural networks. In *EMNLP*, pages 1753–1762, 2015.
- [110] Kai Zeng, Jiacheng Yang, Haixun Wang, Bin Shao, and Zhongyuan Wang. A distributed graph engine for web scale rdf data. In *VLDB*, pages 265–276, 2013.
- [111] Dell Zhang and Wee Sun Lee. Question classification using support vector machines. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 26–32, 2003.
- [112] Weiguo Zheng, Lei Zou, Xiang Lian, Jeffrey Xu Yu, Shaoxu Song, and Dongyan Zhao. How to build templates for rdf question/answering: An uncertain graph similarity join approach. In *SIGMOD*, pages 1809–1824, 2015.
- [113] Guangyou Zhou, Zhiwen Xie, Jimmy Xiangji Huang, and Tingting He. Bi-transferring deep neural networks for domain adaptation. In *ACL*, 2016.
- [114] Guangyou Zhou, Zhao Zeng, Jimmy Xiangji Huang, and Tingting He. Transfer learning for cross-lingual sentiment classification with weakly shared deep neural networks. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 245–254, 2016.
- [115] Hai Zhuge and et al. Query routing in a peer-to-peer semantic link network. *Computational Intelligence*, 21(2):197–216.
- [116] Lei Zou, Ruizhe Huang, Haixun Wang, Jeffer Xu Yu, Wenqiang He, and Dongyan Zhao. Natural language question answering over rdf: a graph data driven approach. In *SIGMOD*, pages 313–324, 2014.







## 致谢

不登高山，不知天之高也；不临深溪，不知地之厚也。博士生涯一晃接近尾声，收获，留恋，感恩。

感谢汪卫老师长期以来的指导。汪老师的学识渊博，对研究有着独特的洞察。他给我的学术指导，往往给我新的体会。同时汪老师为我的研究提供了充足的物质条件支持，以及自由的科研环境。在读博生涯中，我始终保持着对学术探索的乐趣。我想这对汪老师对我的自由研究环境的支持是分不开的。

感谢肖仰华老师的悉心教诲。肖老师是一位进取心强，责任感十足的老师。他经常和我进行一对一的长时间指导，这对于我的学术水平的提升起到了决定性的促进作用。和肖老师在一起的时间越长，越能感到他对事物的深刻理解与智慧，这一点也影响到了我对很多学术之外的事物的观察和理解。能得到这样一位老师的言传身教，是我的幸运。

还要感谢其他老师们。感谢Facebook研究院王海勋老师的指导。海勋的指引直接确立了我博士期间研究选题方向。同时和他在一起的研究，使我学到了研究要面向真实问题，做有实在价值的科研。感谢犹他大学的李飞飞教授，从他身上我感受到了科研的激情。他的高产成果，也成为了我奋斗的目标。感谢延世大学的Seung-won Hwang教授、复旦大学的阳德青教授、香港科技大学的宋阳秋教授，与他们的讨论交流，直接促进了我的研究进展。感谢复旦大学张亮老师、顾宁老师对我的研究的指导。

感谢复旦大学知识图谱研究组同学们的帮助。包括许勇、周西友、陈砺寒、蒋思航、许勇、许陆、林航宇、胡弋舟、陈一川、陈垚亮、鲁轶奇、刘琦、梁家卿、谢晨昊、徐波、梁斌、洪骥、张怡菲等。

最后，谨以此文献给我的父亲崔守海先生以及母亲张家艳女士。



## 攻读博士学位期间发表论文情况和研究成果

1. 崔万云, Yanghua Xiao, Haixun Wang, Yangqiu Song, Seung-won Hwang, Wei Wang, KBQA: Learning Question Answering over QA Corpora and Knowledge Bases, (**VLDB 2017**), **CCF Rank A Conference**
2. 崔万云, Yanghua Xiao, Wei Wang, KBQA: An Online Template Based Question Answering System over Freebase, (**IJCAI 2016**), **CCF Rank A Conference**, demo
3. 崔万云, Xiyu Zhou, Hangyu Lin, Yanghua Xiao, Haixun Wang, Seung-won Hwang, Wei Wang, Verb Pattern: A Probabilistic Semantic Representation on Verbs, (**AAAI 2016**), **CCF Rank A Conference**
4. 崔万云, Yanghua Xiao, Haixun Wang, Wei Wang, Local Search of Communities in Large Graphs, (**SIGMOD 2014**), **CCF Rank A Conference**
5. 崔万云, Yanghua Xiao, Haixun Wang, Yiqi Lu, and Wei Wang. Online Search of Overlapping Communities, (**SIGMOD 2013**), **CCF Rank A Conference**
6. Bo Xu, Yong Xu, Jiaqing Liang, Chenhao Xie, Bin Liang, 崔万云 and Yanghua Xiao, CN-DBpedia: A Never-Ending Chinese Knowledge Extraction System, IEA/AIE 2017, EI.
7. Yaoliang Chen, Ji Hong, 崔万云, Jacques Zaneveld, Wei Wang, Richard Gibbs, Yanghua Xiao and Rui Chen, CGAP-align: A High Performance DNA Short Read Alignment Tool, **Plos One**, 2013, **SCI**, **IF=4**
8. Deqing Yang, Yanghua Xiao, Hanghang Tong, 崔万云, Wei Wang, Towards Topic Following in Heterogeneous Information Networks, (**ASONAM 2015**), EI.
9. Hui Wang, 崔万云, Yanghua Xiao, Hanghang Tong, Robust Network Construction against Intentional Attack, (**BigComp 2015**), Invited Paper, EI.



## 复旦大学 学位论文独创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。论文中除特别标注的内容外，不包含任何其他个人或机构已经发表或撰写过的研究成果。对本研究做出重要贡献的个人和集体，均已在论文中作了明确的声明并表示了谢意。本声明的法律结果由本人承担。

作者签名： 崔万云 日期： 2017.5.30

## 复旦大学 学位论文使用授权声明

本人完全了解复旦大学有关收藏和利用博士、硕士学位论文的规定，即：学校有权收藏、使用并向国家有关部门或机构送交论文的印刷本和电子版本；允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其它复制手段保存论文。涉密学位论文在解密后遵守此规定。

作者签名： 崔万云 导师签名： 132 日期： 2017.5.30