

学校代码: 10285

学 号: 20144227003



# 硕士学位论文

(学术学位)



基于多形态非主属性数据的实体匹配算法研究

Entity Matching Based on Polymorphic Non-Key Attributes

研究生姓名	杨强
指导教师姓名	李直旭
专业名称	计算机科学与技术
研究方向	数据管理与数据分析
所在院部	计算机科学与技术学院
论文提交日期	2017 年 3 月



## 苏州大学学位论文独创性声明

本人郑重声明：所提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。除文中已经注明引用的内容外，本论文不含其他个人或集体已经发表或撰写过的研究成果，也不含为获得苏州大学或其它教育机构的学位证书而使用过的材料。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本人承担本声明的法律责任。

论文作者签名：\_\_\_\_\_日 期：\_\_\_\_\_



## 苏州大学学位论文使用授权声明

本人完全了解苏州大学关于收集、保存和使用学位论文的规定，即：学位论文著作权归属苏州大学。本学位论文电子文档的内容和纸质论文的内容相一致。苏州大学有权向国家图书馆、中国社科院文献信息情报中心、中国科学技术信息研究所（含万方数据电子出版社）、中国学术期刊（光盘版）电子杂志社送交本学位论文的复印件和电子文档，允许论文被查阅和借阅，可以采用影印、缩印或其他复制手段保存和汇编学位论文，可以将学位论文的全部或部分内容编入有关数据库进行检索。

涉密论文 ☐

本学位论文属 \_\_\_\_\_ 在 \_\_\_\_\_ 年 \_\_\_\_\_ 月解密后适用本规定。

非涉密论文 ☐

论文作者签名： \_\_\_\_\_ 日 期： \_\_\_\_\_

导 师 签 名： \_\_\_\_\_ 日 期： \_\_\_\_\_



# 基于深度学习技术的信息抽取工具的研究与开发

## 摘 要

信息抽取表示自动的从文本中抽取结构化信息，比如实体，实体间的关系，事件或者包含在有噪音的非结构化文本中的属性描述性实体，将非结构化文本抽取有用的信息存储在结构化形式中，比如表格，XML，图谱，有着非常重要的价值，信息抽取是构建知识图谱的基础理论和实践支撑。信息抽取又包含了若干个子问题，比如，命名体识别，开放域信息抽取，关系抽取以及文本分割。分割式信息抽取是一个特殊的信息抽取问题，它是通过分割半结构化文本信息来抽取隐含其中的属性值。

目前主流的解决方案是采用机器学习的方法，包括使用人工标注的训练集使用监督式机器学习方案，或者利用事先存在的知识库来辅助实施非监督式的机器学习方案，在非监督式方法中，一个杰出的解决方案是借助于知识库，讲整个信息抽取流程分为分割，匹配，调整等若干个模块，取得了目前为止最好的表现。但是，当采用监督式方案是，获取标注好的训练集要花费非常昂贵的代价，并且往往只限制在某个领域内。而通过知识库辅助的方案，则会出现两个主要问题，1. 固定领域文本的属性顺序限制在固定的顺序上，2.匹配准确率底下。

为了解决上述方案的问题，本文提出了结合深度学习和概率模型方案，充分利用深度学习强大的特征抽取和组合能力，并结合概率模型的适用性和可解释性。构建了完整的高效的解决方案。本文提出的方法同样借助知识库来克服人工标注的问题，并且有效克服了匹配率底下以及限定属性顺序的问题。具体研究内容如下：

(1)如何选择深度学习模型，如何将深度学习模型与概率模型完美的结合起来，并探索深度学习模型构建和优化的策略，包括构建有效的丰富的特征。

(2)提出了基于卷积神经网络的贪婪式概率匹配算法。本算法采用深度卷积神经网络模型，提出了一种贪婪式的概率匹配算法，并在抽取过程中学习出一个双向的序列与位置模型来完成抽取任务，采用了步步提升的设计方案，有效的解决了上述的问题，经过比较F1值取得了非常明显的提升，并在抽取效率上也有非常好的表现。

(3)提出CNN+LSTM+attention model+CRF的解决方案

**关键词：** 深度学习, 信息抽取, 文本分割

**作    者：** 胡猛

**指导教师：** 李直旭



# Information Extraction based on Deep Learning technology

## Abstract

With the rapid development of information technology, the data from different field explosively grow, which makes data quality problems increasingly highlight, such as data distortion, data staleness, data missing, inconsistent data expression and so on. This paper mainly do research on one of the important data quality subjects, so-called Entity Matching (EM for short). EM aims at identifying records referring to the same entity within or across databases.

So far, most existing EM algorithms depend on string similarity metrics to measure the similarity between key attribute values of entities and then make decisions according to a predefined similarity threshold. But an arbitrary threshold is bad for either the matching precision or the recall.

So as to solve the problems of the existing methods, we propose entity matching algorithms based on polymorphic non-key attributes by analysing textual data and computing similarity do EM. Our methods are orthodox to the existing EM methods based on key attributes. We mainly pay attention on how to use non-key attributes smartly to improve the precision and recall of EM based on key attribute only. More details are shown as follows:

- (1) We focus on the problem of EM in the paper. Some existing EM methods are introduced here and the advantages and disadvantages of them are also analysed.
- (2) We propose non-key attributes based EM algorithms which select non-key attributes according to their identification ability to do EM. With the proposed methods, we can not only solve the problem of different expression but also overcome the problem of missing values.
- (3) We propose textual data based EM algorithms which mine the key information from textual data. The precision and recall of EM using the proposed methods are greatly improved .

We demonstrate the effectiveness and availability of the proposed methods on on real-

world datasets. Our empirical study shows that our proposed EM methods outperform the state-of-the-art EM methods by reaching a higher EM precision and recall. And the efficiency of EM is also improved greatly by employing the proposed data block algorithm to reduce the times of comparison.

**Keywords:** Data Quality, Entity Matching, Polymorphic Non-key Attributes Data, Accuracy, Efficiency

**Written by** Meng Hu

**Supervised by** Zhixu Li



# 目 录

<b>第一章 绪论</b>	<b>1</b>
1.1 课题研究背景	1
1.2 课题研究现状	2
1.3 课题研究内容	3
1.4 课题研究意义	4
1.5 文章组织结构	5
<b>第二章 相关理论及方法</b>	<b>6</b>
2.1 深度学习在自然语言处理上的应用	6
2.1.1 词向量	7
2.1.2 各种深度学习模型的使用	9
2.2 分割式信息抽取概念以及现有的方法	12
2.2.1 基于图的概率模型	12
2.2.2 基于知识库实现无监督的抽取模型	14
2.3 本章小结	14
<b>第三章 基于卷积神经网络的贪婪式概率标注算法</b>	<b>16</b>
3.1 算法概述	16
3.2 基于卷积神经网络的贪婪式概率标注算法	18
3.2.1 基于知识库的初始分割和标注	18
3.2.2 基于卷积神经网络模型的构建和文本块的标注	20
3.2.3 输入文本的贪婪式概率标注	22
3.2.4 基于序列和位置特征的结果修正	22
3.3 实验结果与分析	22
3.4 本章小结	22
3.5 预备工作	23
3.6 基于非主属性区分度的实体匹配算法	24
3.7 基于规则的概率决策树的实体匹配算法	25
3.7.1 算法基本思路	25
3.7.2 基于规则的概率决策树的构建	27

3.7.3	实体匹配过程 .....	31
3.7.4	基于PRTree的NokeaEM算法复杂度分析 .....	34
3.7.5	基于PRTree的NokeaEM算法的扩充算法 .....	35
3.8	实验结果及分析 .....	36
3.8.1	参数设定 .....	37
3.8.2	与以往方法在匹配效果方面的比较 .....	37
3.8.3	与以往方法在匹配效率方面的比较 .....	39
3.8.4	算法的扩展性评估 .....	40
3.9	本章小结 .....	41
<b>第四章</b>	<b>基于文本类型非主属性数据的实体匹配 .....</b>	<b>42</b>
4.1	问题定义 .....	42
4.2	传统的文本数据处理方法 .....	44
4.3	数据分块算法 .....	45
4.4	基于IDF的迭代式实体匹配算法 .....	46
4.4.1	Baseline算法的工作流程 .....	46
4.4.2	短语IDF得分的迭代式更新 .....	47
4.5	基于短语共现的实体匹配算法 .....	47
4.5.1	共现图的构建 .....	48
4.5.2	PC-Graph的分割 .....	49
4.5.3	子主题及其权重的获取 .....	51
4.5.4	实体匹配过程 .....	52
4.6	实验结果及分析 .....	53
4.6.1	与以往方法在匹配效果方面的比较 .....	53
4.6.2	算法提取结果的评估 .....	55
4.6.3	算法的扩展性评估 .....	56
4.7	本章小结 .....	57
<b>第五章</b>	<b>总结与展望 .....</b>	<b>58</b>
5.1	全文总结 .....	58
5.2	工作展望 .....	59
	攻读硕士学位期间发表的论文 .....	<b>61</b>

致谢 .....	63
----------	----

# 第一章 绪论

## 1.1 课题研究背景

在这个瞬息万变的信息时代，我们接触的信息量呈现爆炸式的增长，特别是文本信息，它们在购物网站，新闻网站，论坛交流以及社交活动等中被广泛地创造出来，是一种最丰富也是最杂乱的信息格式。通常，这些文本信息可以免费获取，但是却覆盖了各种主题和内容，内容的书写格式也是各种各样，没有任何一个统一的严格要求的的出版格式。图1展示了几种在互联网上常见的文本信息来源。大量有价值的信息和知识就隐藏在这些杂乱的文本信息中，如何从文本中抽取出有价值的东西是一个非常重要的任务。

这些信息来源蕴含了大量的各种各样的有价值的数据，根据信息种类，我们能从中找到个人信息，商品信息，出版信息，公司信息，城市信息，天气信息等，我们从这些信息中便可以推理出一些有价值的关系，去认识用户的偏好，来检测某种趋势或者去命名一下东西等。但是，文本多样性在赋予文本信息丰富的表述意义的同时，也带来了更大的限制，这种限制体现在这些文本信息中存储的信息和知识很难被通用化的组织起来。更特殊的一种情形，对于一些语义丰富的文本段，比如商品信息描述，引文信息，地址信息，分类广告，电影评论等等，这些文本来源也是一种无结构的形式，所以抽取文本内容很难找到某种自动化处理的方法。此外，在这种文本类型中，隐含其中的信息是由若干个没有关系的个体属性值组合起来的，这就更增加了抽取的难度。

现如今人工智能的发展，在算法和硬件上都得到了长足的进步，但现在的人工智能大多还处于感知智能阶段，比如如何去识别一个东西，但是从感知智能到认知智能的进化，就需要赋予机器认知能力。人之所以具备认知能力，就是因为人的大脑中存储了大量的知识，这些知识不断的增加，被完美的组织起来，并且可以流畅的推理和运用这些知识。而对机器来讲，构建知识图谱——通过不同知识的关联性形成成一个网状的知识结构，这个过程本质就是建立认知，理解世界。现在越来越多的人意识到知识图谱的对于人工智能发展的意义，构建知识图谱是一个庞大的工程，这个工程的第一步便是信息抽取，或者称为知识获取，包括命名题识别，关

系抽取，属性值抽取，构建三元组等。无疑，信息抽取的能力直接决定了知识图谱构建的准确性和多样性，为了构建一个准确的、丰富的知识图谱，首先需要我们能够从各种各样的文本信息中抽取出有价值、有联系的知识点。在如今的信息洪流中，有无穷无尽的知识点蕴藏在这些文本中，这导致了信息抽取的必要性和价值，但是也因为文本格式和内容的多样性，又导致了信息抽取的棘手性和难度。分割式信息抽取作为信息抽取中重要的一类，这种信息抽取的文本格式具有无噪音、属性值间无关联等特点，是一种复杂又重要的抽取任务，同样具有相同重要的研究价值。因此，如何建立快速、准确的信息抽取模型成为了研究如何构建知识工程和知识图谱的重点和难点问题。

## 1.2 课题研究现状

信息抽取大致分类可以从两个层次分析，一是封闭语料和开放语料的区别，即文本信息源来自限定领域的新闻语料和整个Web页面，二是限定类别和开放类别的区别，即有限的实体、关系、事件和维基百科级别的条目。分割式信息抽取，属于封闭式语料和限定类别的信息抽取任务，由于它在实际应用中的重要性，有很多研究致力于解决这个问题，主要的解决方案是使用机器学习的技术，分为监督式[1,2,3,4,5]的和非监督式的[1,2,4,5]。

在监督式机器学习方法中，基于隐马尔可夫模型和随即向量场模型的概率图模型被广泛应用，目前表现最好的是基于随即向量场模型的解决方案，但是，取得了非常好的效果，但是，当采用监督式机器学习方法时，需要大量的标注数据，这在实际的而应用中将付出非常昂贵的代价，并且在某些领域根本无法构建标注语料。

为了避免监督式机器学习方法的缺点，提前构建一个语料库，就是固定领域的文本信息，把这个语料库看做一个参考表，实现非监督的抽取模型。简要来说，首先使用机器学习模型，比如隐马尔可夫模型或者随机向量场模型，先从这个语料库中学习到这一语料的属性值顺序，然后使用这个模型在这个领域新的文本中做抽取任务，这中方案限制了属性值顺序，只能应付固定的应用场景，对于属性值顺序不固定的领域便无法适用。最近出现了一种使用知识库，这种知识库中存的不再是整句文本记录，而是将某一领域的各个属性值分开存储，不显性的训练模型来实现非



监督的抽取模型。文献[1]提出了一种基于这种知识库的非监督式方法，输入文本首先基于知识库做初始的分割，得到很多粗略的文本块，这些文本块再通过一些固定的匹配方程来匹配到指定的属性上，最后通过一个位置和序列模型做进一步的修改和补充。但是由于这些匹配函数表征力的缺失，以及这种位置和序列模型的先天缺陷，使得这种方法的抽取质量差强人意，但是这种方案无疑提供了一种很好的实施非监督式模型的解决思路。

上述方法虽然可以解决部分领域或部分场景的信息抽取任务，但仍旧存在一定的缺陷与不足。因此，如何引入深度学习模型，并将深度学习模型和概率模型结合起来更好的完成分割式信息抽取的任务，并以小见大，探索深度学习在信息抽取领域更多场景的使用，是本文的研究重点。

### 1.3 课题研究内容

本文研究了基于深度学习模型的分割式信息抽取问题，提出了结合深度卷积神经网络和概率模型的非监督式信息抽取模型，大大提高了抽取质量，抽取效率也得到了明显的提升。此外，还搭建了一个定制训练模型、执行抽取任务、展示抽取结果的可视化系统，可以更方便地将此模型移植到更多场景，覆盖更多的领域，并简化训练模型的操作难度。本文主要解决三个问题：(1)如何引入深度学习模型，如何通过构造特征将文本信息输入变成丰富的向量化表示。(2)如何将深度学习模型的结果与知识库结合起来，实现文本分割和属性标注。(3)如何更好的学习到输入文本信息的内在规律和特征表述，用以进一步优化抽取表现。本文的主要内容如下：

(1)本文主要研究分割式信息抽取问题，介绍已有的信息抽取方法，描述具有代表性的各类算法和优势，并分析其中存在的问题。

(2)介绍一些关于深度学习和信息抽取相关的理论知识和研究现状。

(3)提出了一种基于深度学习模型的贪婪式概率标注算法。传统的机器学习算法受限于标注语料缺失、应用场景限制、匹配函数表征力弱等影响，使得抽取的准确率难以得到保障。而随着应用场景的扩增，越来越多领域的文本信息需要被考虑进来，面对的文本格式也越来越杂乱，因此必须提出一种适用更强、准确率更高的模型。另外，我们需要处理大量的文本信息，因此抽取效率也是不得不考虑的一个重

要方面。因此，本文开创性的使用深度学习模型来解决分割式信息抽取这个问题；提出了一种基于深度学习模型输出结果的贪婪式概率标注算法；进一步改进了现有方法，提出了一种双向的序列与位置模型。

(4)介绍可操控和可视化的系统，包括如何定制模型训练参数，查看模型训练过程，查看抽取各步骤的结果，查看学习到的双向位置和序列模型等。

## 1.4 课题研究意义

文本信息抽取是指从自然语言文本中自动抽取指定类型的实体，关系，事件等信息的应用技术。面对日益增多的海量信息，人们迫切需要一种自动化工具来帮助自己从这些海量文本信息中快速发现有价值的信息，并将这些信息自动化的进行分类，提取，重构并最终存储在结构化的容器中，便可以做进一步的查询，处理和分析。在这种背景下，信息抽取技术应运而生，信息抽取处理的对象可以是文本，图像，语音和视频等多种媒体，但通常指的是文本信息抽取。

信息抽取技术作为很多现实应用的支撑和补充，具有非常高的研究意义。从满足用户需求来讲，信息检索是一项综合了各种技术的重要问题，用于从一个大的文档集合中找出用户需求的相关文档，信息抽取技术可以从文档中抽取出粒度更小的实体，关系或事件，变可以满足用户更细粒度的请求和检索，从这个角度来讲，信息抽取可以作为信息索引的有益补充。信息抽取是随着互联网发展而兴起的信息处理技术，信息抽取是获取额外信息的有力补充手段，在人工智能的发展中，特别是从感知智能进化到认知智能，知识的获取和应用是至关重要的环节，知识是人工智能的基石，只有赋予机器更多知识，并让这些知识融会贯通，才能让机器学会推理和认知，知识图谱看做机器的知识库，被赋予了很多的期待。要想从更多的信息源、更复杂的信息格式、更多的信息量中准确的、高效的抽取出知识，信息抽取是至关重要的第一步。如何从巨大的网络信息洪流中抓获有用信息，构建一个覆盖面更广的知识图谱，是一个非常重要的研究方向，目前也是一个非常困难的问题。

正因为文本信息来源的多样性和文本格式的复杂性，传统的信息抽取方法渐渐展现出很多不足之处，包括抽取质量差，应用领域受限制或者抽取效率低等。随着深度学习算法的发展，首先在图像和语音上取得了突破性的进展，现如今已经成为

发展人工智能的基础算法，越来越多成熟的解决方案都采用深度学习算法，深度学习以其强大的特征抽取和特征组合特质，具有意想不到的表征能力。现在，深度学习也被越来越多的应用到处理自然语言上，信息抽取作为自然语言理解领域重要的一部分，同样也不会错过深度学习这个强大的工具。因此，探索如何将深度学习应用到本文的研究重点——分割式信息抽取中，具有很重要的研究意义和现实意义。

## 1.5 文章组织结构

全文共分为五章，其组织结构如下：

第一章简要介绍了该课题的研究背景、研究现状、研究内容、研究意义与文章的组织结构。

第二章介绍了一些相关理论及方法，给出了信息抽取的相关概念，并结合例子阐述了分割式信息抽取问题。此外，介绍了深度学习的一些理论基础，着重介绍了深度学习在自然语言理解上的应用。通过例子说了信息抽取遇到的主要问题，面临的挑战。最后介绍了一些常见的分割式信息抽取方法，并简述了各个方法的优缺点。

第三章介绍了本文提出的基于深度学习的贪婪式概率标注模型，首先给出问题定义，然后介绍如何构建深度学习模型，介绍如何结合知识库和深度学习模型结果来处理分割和标注问题，并介绍本文提出的贪婪式概率标注算法，最后介绍双向序列和位置模型，用来对之前的标注结果做进一步的补充和修改。

第四章完整信息抽取系统的可视化页面展示，包括在线定制参数，模型训练结果可视化，提出方法的抽取过程展示，最终结果的展示等。

第五章对本文所做的工作进行总结，并对未来的工作进行展望。

## 第二章 相关理论及方法

本章首先介绍了深度学习用于自然语言处理的一般处理方案，介绍了深度学习在一些信息抽取领域的应用，并分析了深度学习用于自然语言理解上的优缺点。然后介绍了信息抽取的相关概念及其定义，重点阐述了分割式信息抽取的问题定义。最后详细介绍了常见的分割式信息抽取方法，并分析了各类方法的优缺点。

### 2.1 深度学习在自然语言处理上的应用

最近几年来，深度学习架构和算法在诸如图像识别和语音处理上取得了突破性的进展，不断刷新各个榜单，应用场景越来越多，在工业界也有越来越多的落地产品，可以说深度学习的发展引导了这一波人工智能的热潮。图像和语音是自然界的产物，它的特征是一种更自然更丰富的表征，而深度神经网络的优势就在于它的特征抽取和特征组合能力，这是深度学习能在图像和语音上取得成功的原因。然后，人类的语言，属于人类文明创造的事物，不具备自然的表征能力，因此表达形式更主观，具有高度结构化、高抽象化、数据量相对小等特点，是一种更粗粒度的表现形式。因此，一开始，深度学习在自然语言处理领域的应用效果非常一般，随着算法的发展和思路的提升，特别是词向量的引入使得我们可以更好地将自然语言向量化，现在已经在越来越多的自然语言处理任务中证明，深度学习可以有更优秀的表现，甚至是最佳的表现。比如，深度神经网络模型在诸如文本分类，关系抽取，命名体识别，机器翻译等任务中的表现已经大大超越了传统方法，并且深度学习也在继续拓展它在自然语言理解上的应用。

在很长一段时间，自然语言处理的研究方法都是采用这些浅层的模型，来学习到非常高维且稀疏的特征表示。在传统的机器学习中，我们使用各种算法的基础是需要手工设计特征，因此特征工程是一个非常基础性的工作。只有当人们对特定领域的知识有非常透彻的理解时，才能构造出足够多、足够优质的特征。比如，对于命名题识别任务，当要识别地名和机构名，我们首先需要构造出如下的特征列表：然后将特征喂给某个机器学习算法，比如线性分类器，分类器构造出目标函数，再通过凸优化策略不断调整模型的权重和偏置，使误差优化到最小，在这样的过程中为这些特征找到最合适的权重。这样的人工设计的特征，常常需要定义过多，并且

Feature	NER
Current Word	✓
Previous Word	✓
Next Word	✓
Current Word Character n-gram	all
Current POS Tag	✓
Surrounding POS Tag Sequence	✓
Current Word Shape	✓
Surrounding Word Shape Sequence	✓
Presence of Word in Left Window	size 4
Presence of Word in Right Window	size 4

图 2-1 命名题识别任务特征列表

一般是不可能做到完整的，需要花费大量的时间去设计和验证。

然而在自然语言处理上，由于语言的特殊性，设计有价值的特征是非常困难的事情，想要从如此抽象的文本信息中抽取出有用的特征，必须经过不断的迭代和实验，这是非常耗时的。深度学习是一种端到端的模型，我们只需要提供输入，不需要做额外的特征工程，深度神经网络会自动的进行特征抽取、特征组合。我们知道，图像信息是以像素点作为表示单位，声音是以声波作为表示单位，这些都输入较为底层的原始输入信号，可以直接数字化，作为深度神经网络的输入，进行端到端的模型训练。然而，一句文本却是以一个独立的单词或字组成的，虽然单词或字是独立的，但是组合成一起就成为了一句有意义的表述，如何将一句文本向量化表述，同时体现单词之间和字之间的联系，是一个重要的切入点。近些年，基于稠密向量表示的深度神经网络在很多自然语言处理任务上取得了更好的表现，也就是词向量的横空出世，可以说是将深度学习引入到自然语言处理领域的重要转折点。下面介绍一下词向量以及词向量结合深度学习在自然语言中的应用。

### 2.1.1 词向量

想要将自然语言理解问题转化成机器学习问题，首先需要将自然语言数字化，就如何将图像、语音数字化相同。最直观的一种方式，也是最传统的方法——采用one-hot 的编码，将每个单词表示成 $R^{|V| \times 1}$  维的向量，其中 $|V|$ 表示词典中词语的个数，对于第 $i$ 个词语的向量，只有 $i$ 下标处为1，其它维都为0。比如'food', 'eat',

'laptop' 作为前三个词, 则他们的one-hot 向量表示为:

$$w^{food} = [1, 0, 0, 0, \dots, 0] \quad w^{eat} = [0, 1, 0, 0, \dots, 0] \quad w^{laptop} = [0, 0, 1, 0, \dots, 0] \quad \dots$$

这种向量的表示形式, 实现了将自然语言数字化的目的, 在实际的应用中, 配以Hash的处理, 再结合一些机器学习算法就可以很好的解决自然语言处理领域各种主要的任务了。但是它存在两个最主要的问题:

(1)纬度灾难[Bengio 2003]。当某一语料词典数目过大, 这个向量的纬度会变得很大, 数据会变得特别稀疏, 导致统计语言模型会出现很多为零的条件概率, 这需要花费大量的精力来处理零概率的问题。这导致在训练模型进行矩阵时计算会非常困难, 特别是应用在深度学习模型上, 容易导致纬度灾难。

(2)不能很好地刻画单词之间的相似性。也就是常常说的'词汇鸿沟',对于两个词性或者语义上有关联的词语, 他们的向量表示应该能体现出一些关联性和区分性, 但是使用ont-hot 编码无法实现这个目的。比如对于'food','eat','laptop'这三个词:

$$(w^{food})^T \times w^{eat} = (w^{eat})^T \times w^{laptop} = 0$$

他们的向量相乘结果是相同的, 且都等于0, 这就无法体现'food'和'eat'之间的相关性, 也不能体现这三者的区分性了。

因此我们想要去学习出一种使用低纬度空间的分散式表示来向量化一个词语, 同时又需要将相似性和关联性融入到这个低维向量中, 这就需要我们用语言模型来训练这个词向量, 因此构建词向量的原理依据就是具有相似含义的词语往往出现在相似的语境之中。使用神经网络训练语言模型在[Bengio 2003]中被首次提出, 作者阐述了传统基于统计的语言模型的主要问题, 并说明了使用神经网络训练语言模型的优势。词向量得到革命性的发展是[Mikolov et al]提出的CBOW模型和Skim-gram模型, 也就是著名的word2vec。

首先对词典中的所有单词进行one-hot编码, 词典中词语数目为 $V$ 。CBOW 模型是使用周围的 $2C$  个词语来预测目标词语, 比如对于一句话"狗喜欢吃熟的骨头", 我们可以将{'Σ''U','æ','~'} 看做周围语境(两边各 $C$  个词语), 来预测和生成目标词语{''},

而Skim-gram模型是跟这个思路相反的,它使用目标词语来预测它周围的 $2C$ 个词语。CBOW模型的简单形式如图2-3所示,这里只考虑了句子中的一个词语。CBOW模型本质上就是一个三层的全连接神经网络(包含一个隐藏层),输入层和输出层都具有 $V$ 个神经元结点,隐藏层具有 $N$ 个神经元结点,往往 $N$ 要大大小于 $V$ 。则CBOW模型的输入层就是通过one-hot编码的 $V$ 维度向量,这样one-hot编码的向量的每一维的值便可与输入层每个神经元结点对应起来。更一般的形式如图2-4所示,模型的输入就是目标词语的周围语境,即 $C$ 个词语。假设输入的one-hot词向量表示为 $x^{(i)}$ ,模型的输出表示为 $y^i$ ,这个模型的唯一输出 $y^i$ 也就是我们的目标词语 $y$ ,同样也是使用one-hot编码的向量。另外,我们需要创造两个矩阵,也是这么模型输入层到隐层和隐层到输出层的权重矩阵,  $W^1 \in \mathbb{R}^{N \times |V|}$  和  $W^2 \in \mathbb{R}^{|V| \times N}$ , 这里 $N$ 就是隐层神经元数目,也表示最终词向量的纬度空间。

这个语言模型的训练过程可以分解为如下几个步骤,对于第 $i$ 个目标词语:

- (1)首先生成 $2C$ 大小上下文词语的one-hot向量,记做 $(x^{i-C}, \dots, x^{i-1}, x^{i+1}, \dots, x^{i+C})$
- (2)对于这个上下文,计算其中所有词语的嵌入向量( $u^{i-C} = W^1 x^{i-C}, u^{i-C+1} = W^1 x^{i-C+1}, \dots, u^{i+C} = W^1 x^{i+C}$ )
- (3)计算这些向量的平均向量 $h = \frac{u^{i-C} + u^{i-C+1} + \dots + u^{i+C}}{2C}$
- (4)计算输出层的得分向量 $z = W^2 h$
- (5)使用softmax将输出层得分向量变成归一化的概率值 $\hat{y} = \text{softmax}(z)$
- (6)使用这个输出概率值与真正的目标词语 $y$ 比较,通过计算交叉熵 $H(\hat{y}^i, y^i)$ 来定义损失函数,再通过梯度下降一步一步调整权重矩阵,以此来优化这个模型。

而最终训练处的模型的输入层到隐藏层的权重矩阵 $W^1$ 才是我们需要的词向量表示,具体来讲,第 $i$ 个词语的词向量就是 $W^1$ 的第 $i$ 列的向量 $w^i \in \mathbb{R}^{N \times 1}$ 。图2-2展示了词向量样例,可以看到可以体现词语之间的相似性。当然这只是最基础的词向量构造方法,还有很多版本的词向量构造思路,最近基于字符级别的字符向量可以捕获到更多句法级别的信息,展现了巨大的应用潜力。

### 2.1.2 各种深度学习模型的使用

理解了词向量的构造原理,就解决了深度学习应用在自然语言处理上最基本的

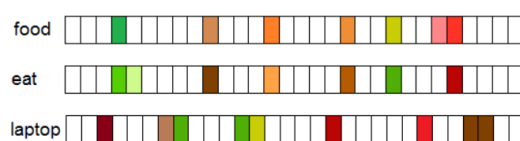


图 2-2 词向量的样例

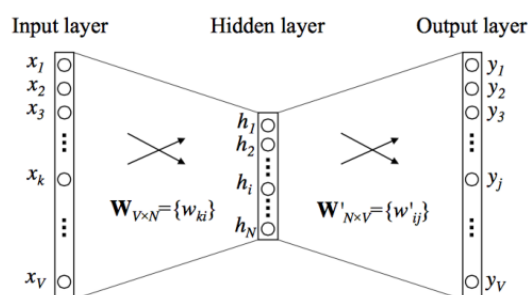


图 2-3 CBOW模型的简单形式

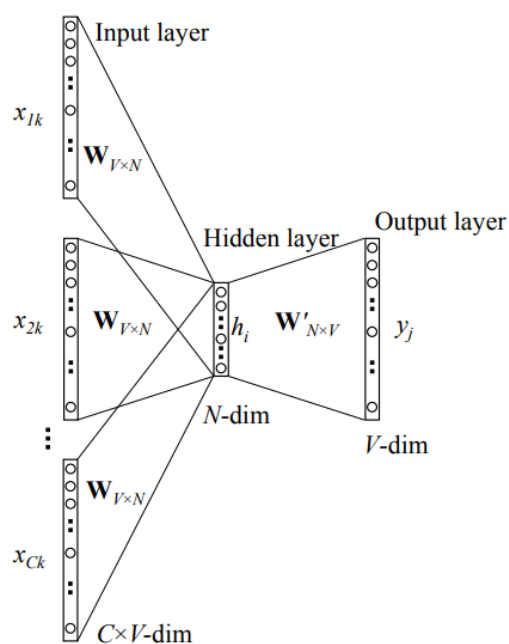


图 2-4 CBOW模型的一般形式

问题——输入的表达。剩下的就是根据具体问题，如何选择模型，如何设计解决方案的问题了。有很多深度学习的模型可供选择，比如深度神经网络DNN, 卷积神经网络CNN, 循环神经网络RNN, 长短期记忆神经网络LSTM, GRU(LSTM的一种变体), 递归神经网络, 基于注意力机制的深度神经网络模型, Encoder-Decoder模型, 记忆网络模型等等。



基于深度学习模型的方法已经在很多自然语言理解领域取得了更好的表现，比如，在文本分类问题上，文章[1][2]提出了基于卷积神经网络的模型，文章[1]使用比较普通的卷积神经网络结构，文章[2]中使用一种多层的交织最大池化层和卷积层的网络结构。文章[3][4]都使用基于循环神经网络模型，两者的对比体现了LSTM 相比普通RNN 的优势，并且文章[4]使用的是一种树形长短期记忆网络(Tree-LSTM)，证明了树形LSTM 相比于双向的LSTM 的优势，这表明Tree-LSTM 能够更好的捕捉到自然语言的句法信息。文章[5]使用的基于记忆网络模型目前为止取得了最优的表现。

在机器翻译任务上，文章[]使用一个四层的seq2seq 模型，使用LSTM，然后用这个模型对从传统SMT方法产生的1000个候选翻译重新排序。文章[]使用一个带有8个编码器和8个解码器的深度LSTM 网络组成，并使用了注意机制（attention）和残差连接（residual connections）。最近，[]提出了一种基于CNN的seq2seq模型，输入的每个词语通过CNN由并行方式组成的注意力架构计算出其表示，另外，解码状态需要同时考虑之前已经产生的结果。[]中不在是使用RNN或者CNN来作为encoder-decoder的模型基础，只使用注意力机制处理序列模型相关的问题，比如机器翻译，这样可以高度并行的工作，在提升翻译性能的同时也大大提升了翻译速度。

在信息抽取任务上，文章[]提出的基于卷积神经网络的模型，特殊之处在于作者将位置信息与词向量组合作为模型的输入矩阵，在关系抽取上取得了非常好的表现。文章[]利用共享神经网络底层表示来进行联合学习，同时进行实体识别和实体的关系抽取。具体来讲，对于输入的句子共享词向量输入层，通过双向LSTM对输入进行编码，然后分别使用LSTM进行命名体识别和CNN来进行关系分类，这种方法可以很好的解决传统的流水线方法的很多弊端。

以上列出的一些自然语言处理的几个领域，使用深度学习已经取得了突破性的进展，在其他一些领域，比如POS标注，语义解析，问答系统等，使用深度学习模型也都是目前最好的途径。当然也有很多工作是使用深度学习结合传统的机器学习算法实现的，深度神经网络具有它天生的优势，会有更多的应用场景。

## 2.2 分割式信息抽取概念以及现有的方法

分割式信息抽取是一个经常被研究的问题，用来分割一段连续文本，然后提取出隐含在这段文本中的各属性值。比如，对于这样一段连续文本，'Mercedes-Benz E250 Auto, \$84888, Obsidian Black, 7speed Sports Automatic. 2 doors 4 seat Coupe, 4 cylinder Petrol TurboIntercooled2.0L;6L/100km'，它有这样几个特点：(1)文本中包含若干属性项，且各个属性值之间是语义相关但不是语法相关的，(2)各属性值不通过固定标点分割，(3)通常属性值顺序不固定，(4)同一语料中不同记录所包含的属性值数目不一定相同，(5)文本中没有噪音数据。这些半结构化文本出现在很多场景下，比如商品信息描述，引文信息，个人信息，地址信息，分类广告，商品评价等等，图[]列出了一些例子。正因为这种文本格式在互联网上广泛的出现，且文本中包含了大量的有价值的信息，研究如何抽取这类文本在学术界和产业界都引起了大量的关注。表[]列举了各个场景下的一些文本样例，可以观察到这类文本的上述一些特征。

对于上面的例子，一个正确的分割结果如图 3-1所示，这个任务主要解决两个问题，一是正确的分割出各属性值，二是给各个分块打上正确的标签。各属性值之间虽然是语义相关的，但是因为其在语法上没有任何关联，所组成的一段文本不是一个常规的自然语言，不符合语法规则，这给这个任务带来了巨大的挑战，我们无法单纯从内容或者语法上来解决这个问题。解决这个问题的方法大体可以分为三个阶段：(1)使用基于图的概率模型,比如隐马尔可夫模型(HMM)和随机向量场模型(CRF)，(2)使用知识库实现无监督的抽取模型

### 2.2.1 基于图的概率模型

因为分割式信息抽取可以看做是一种序列标注问题，使用机器学习技术，特别是一些基于图的概率模型，比如隐马尔可夫模型和随机向量场模型，展现了很好的效果。当使用机器学习技术时，既有使用人工标注语料的监督式模型也有非监督式模型。文章<sup>[2]</sup>首次提出了采用基于图的概率模型来解决信息抽取问题，作者训练独立的隐马尔可夫模型来识别各个属性值，这种方法被扩展成一个DATAMOLD抽取工具<sup>[3]</sup>，在这种方法中，各属性的隐马尔可夫模型，也可称作内部隐马尔可夫模型，被包裹起来组成一个外部隐马尔可夫模型。这些外部的隐马尔可夫模型目的是来刻

画目标文本的各属性值序列状态。内部隐马尔可夫模型和外部隐马尔可夫模型使用标注数据分别训练，在两个数据集上展现了非常出色的效果。

隐马尔可夫模型是一种生成式序列模型，它设计出一种交叉概率来匹配目标文段和标注序列，但是这种模型不擅长表达多层交叉特征和处理较长依赖的序列。后来，基于随机向量场模型的方法被提出来解决此类问题<sup>[2,3]</sup>。相比较隐马尔可夫模型，随机向量场模型具有更强的推理能力，能够使用复杂的，重复的，非独立的特征来序列和推理，因此它能够更充分地利用上下文信息，并且不同于隐马尔可夫模型，它可以使用其他的外部特征来获取更丰富的信息。<sup>[3]</sup>分析，对于不同的输入序列，在隐层状态具有不同的状态转移和发射概率，随机向量场模型更适合用来对这种情况进行建模。尽管这些基于图的概率模型的监督式信息抽取算法表现非常好，但是上面这些方法都是监督式模型，这需要我们标注大量的训练数据，这个工作的花费往往非常昂贵。还有一些情况，训练数据根本就难以获得。为了解决上述的这些问题，出现了一些使用事先存在的数据集来减轻训练过程的方法<sup>[2,3,3,3]</sup>，这些方法使用已经收集的大量的语料，无需使用人力或者只需要一点点人力，就可以完成抽取的任务。这种方法的策略是使用这些已经存在的数据集中的属性值训练出模型，再来在新的输入文本中识别各属性值。<sup>[3]</sup>提出的方法基于随机向量场模型来抽取属性值，先从数据集中学习到一些内容相关的特征，再用一些人工标注的数据来学习结构相关的特征，然后结合两者完成抽取过程。<sup>[2,3]</sup>提出的方法能够仅仅依赖数据集来训练模型，然后再使用这个模型在分割的文本中识别属性值，这种方法不需要使用人工标注的数据集，一旦属性值被识别到，整个抽取过程就算完成了。但是，这个方法基于一个假设：一个数据集中所有文本中的属性值服从同一个序列顺序，这个顺序就是从一批测试样例中学习到的。这两篇文章的不同之处在于，<sup>[3]</sup>使用隐马尔可夫模型，<sup>[2]</sup>使用随机向量场。但是，这种策略虽然避免了人工的依赖，但是因为它的应用前提限制在具有固定属性值顺序的语料上，所以在现实中的使用到限制。然后，<sup>[3]</sup>提出的方法能够解决不同顺序的问题，但是却需要依赖人力的帮助来学习结构相关的特征，增加一些人工的依赖和花费，以便应用在更多的实际场景中，是一种折中的方法。

### 2.2.2 基于知识库实现无监督的抽取模型

不使用上面讨论的基于图的概率模型，也有一些借助知识库实现非监督式模型。<sup>[2]</sup>提出了一种不一样的思路，首先，作者需要构建一种特殊的文本库，使用简单的空间向量模型计算输入文本和文本库中文本集的相似度，系统可以自动找到对于指定抽取任务最相关的文本集。既然文本集已经确定，系统再借助已经定义好的文本相似度矩阵，例如Jaro-Winkler 和Smith-Waterman，来调整一个阈值抽取出有用的信息。这个思路不同于上述的基于机器学习模型的算法，上述方法需要先学习到一些内容相关的特征再做抽取，这种方法是使用一个事先定义好的文本相似度函数来直接做抽取，但是这种方法需要大量的文本集做支撑。

ONDUX<sup>[2]</sup>通过一种不同的思路实现了无监督的抽取模型。这种方法依赖的知识库与之前的都有所不同，它存储的内容不再是整条文本，而是将各个属性值分开来存储构建知识库。整个结构大体可以分为三步，第一步，根据词语在知识库中的共现来分割输入文本，将输入文本分割成若干个文本块。第二步，作者定义了一些匹配函数，比如匹配文本类属性值函数，匹配数字类属性值函数，匹配URL和e-mail类属性值函数，使用这些匹配函数直接去匹配第一步中生成的文本块。第三步，在执行抽取的过程中，作者构建了一个位置和序列模型 $PSM$ ，这个模型用来刻画文本的结构特征，然后使用 $PSM$ 模型最后再对标注结果进行补充和修改。这种方法主要的瓶颈在两点，第一，匹配函数的表现力不足，第二， $PSM$ 模型不能够充分的描述文本的结构特征，它只能单向的提现序列和位置的转移和分布特征，虽然可以提供很大的帮助，但这个模型作为最终的修改参考还不够完美。

## 2.3 本章小结

本章第一部分，首先介绍了深度学习在自然语言理解上的应用，特别是详细地介绍了词向量的生成原理和应用优势，然后列举了一些深度学习模型，并简要介绍了在一些自然语言理解领域使用深度学习模型的方法。第二部分，首先给出了分割式信息抽取任务的概念，然后介绍了两种主要的解决思路，分别列举了一些典型的解决方案，并分析了这些方法的优缺点。

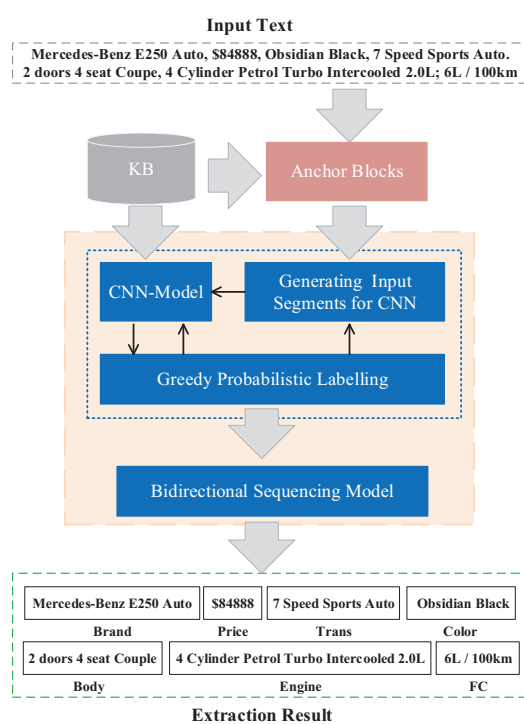


图 2-5 CBOW模型的一般形式

## 第三章 基于卷积神经网络的贪婪式概率标注算法

本章首先给出了基于卷积神经网络的贪婪式概率标注算法(简称为CNN - IETS)概述, 然后给出了分割式信息抽取的正式定义, 并介绍了知识库的结构。接下来详细介绍了算法每一部分的内容, 最后给出了详细的实验结果来证明算法的效果和表现。

### 3.1 算法概述

这一章, 我们提出了一种非监督式概率标注算法, 它基于深度卷积神经网络(CNN)来构建模型, 借助提前构造的知识库来克服需要人工标注语料的难题。卷积神经网络是一种端到端的模型, 先天具有抽取特征和组合特征的优势, 使用深度卷积神经网络能够自动得选择出高质量的特征来更好的刻画各个属性值。而这个知识库有两个作用, 第一个作用是用来训练我们的卷积神经网络模型分类器, 第二个作用是协助生成文本段, 再输入到我们的分类器中。具体一点讲, 如图 3-1, 输入一个文本, 我们首先根据词语在知识库中的共现来初始分隔这条文本, 形成若干个文本段。对于这些文本段, 如果它在某个属性组中出现的频率比较高而在其他属性组中出现的频率比较低, 并且得分高于某个阈值, 它就会被标注成这个属性。然后根据这些初始的分割状态和标注结果, 哪些没有标注的文本段会被输入到下一步的CNN分类器中。在下一步, 我们并不是直接接受CNN分类器的输出结果, 我们根据CNN分类器的输出概率发明了一种贪婪式概率标注算法, 这个算法会从全局考虑分割和标注状态来为输入文本执行最合理的分割和标注方式。对于一些有上面过程产生的漏标注和错误标注的情形, 我们提出了一种双向位置与序列模型, 进一步对标注结果进行调整和改善。

为了构建我们的卷积神经网络模型, 我们使用了一组全面的语义特征和句法特征加入到模型的输入中, 这些特征都是直接从我们的知识库中构造的, 这样我们的模型就可以充分利用这些特征来标注每一个独立的文本段。虽然, 我们的模型可以直接提供一个标注的结果, 但是这些结果仅仅是面对单独的文本段, 没有考虑在同一个输入文本中各文本段之间的内在关系, 这也不能找到最佳的分割和标注结果。因此, 我们需要从全局考虑分割和标注状态来执行概率标注。因此, 最理想的标注

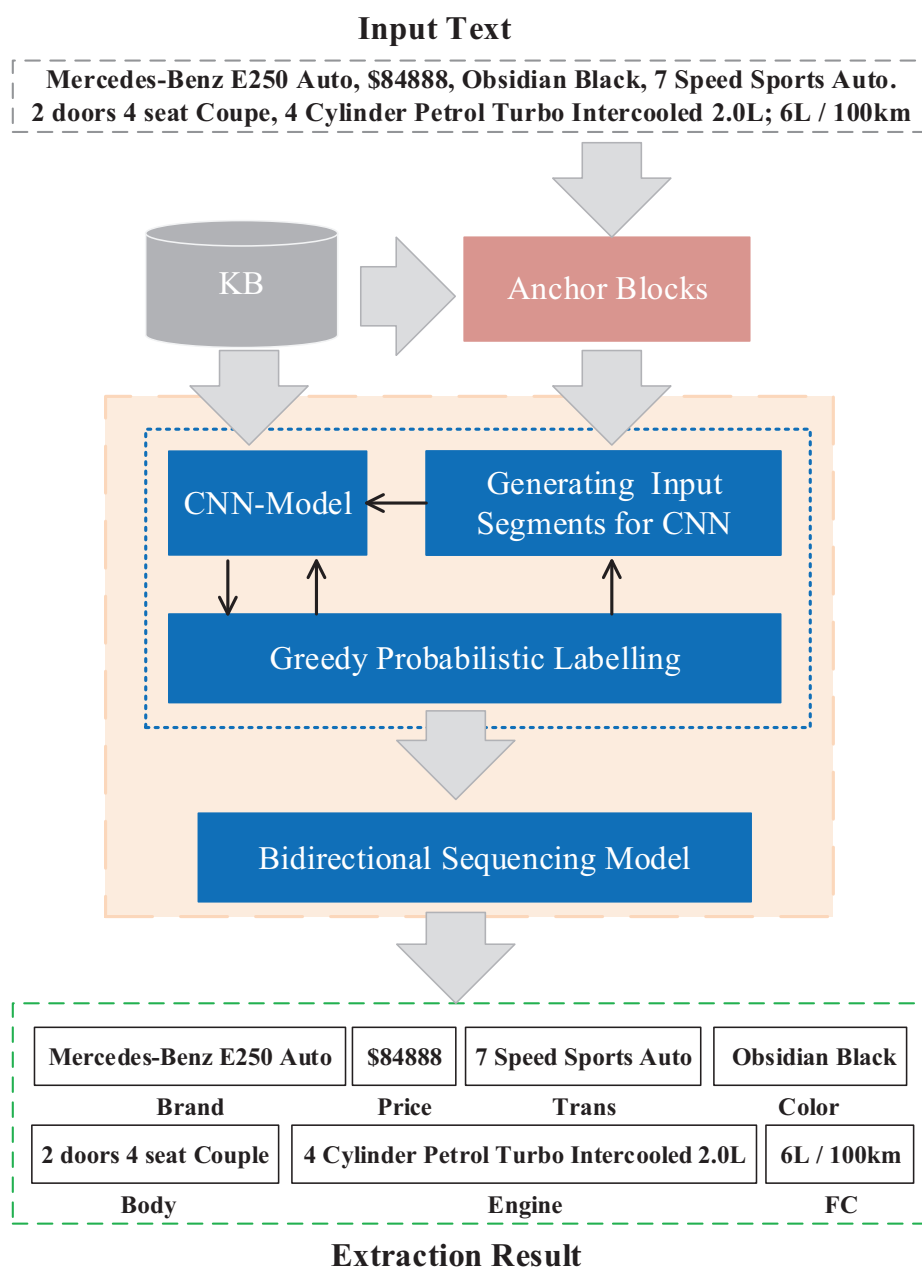


图 3-1 模型的整体结构

方式是，我们依次标注每个文本块来最大化所有标注结果的概率和，但在这个过程中，必须满足一个前提条件，即每个属性值只能包含连续的文本段。我们首先证明了这是一个NP-hard的问题，然后提出了一种贪婪式概率标注算法，此算法能够有效的实现最佳标注方式。

使用上述的基于卷积神经网络的贪婪式概率标注算法，大部分的输入文本都能

够被正确分割和标注了,但是还是会出现一些漏标注和错误标注问题。为了解决此类问题,我们在执行抽取的过程中权衡考虑了已经标注文本段的位置和序列信息,相比较文章<sup>[2]</sup>中提出的单向序列模型,我们的是双向模型,从前向和后向同时考虑序列关系,因此能够减少更多的标注错误并提供更大的帮助。

### 3.2 基于卷积神经网络的贪婪式概率标注算法

在这一节,我们首先给出分割式信息抽取任务的定义,然后详细介绍借助知识库卷积神经网络构建的贪婪式概率标注算法(CNN-IETS)。

#### 定义1.

(分割式信息抽取任务) 对于某个领域的属性集 $\mathbb{A}$ ,假设输入文本 $I$ 包含了一些属性集 $\mathbb{A}$ 中的属性值,分割式信息抽取任务目的是分割 $I$ 成一些列的文本段 $S = \{s_1, s_2, \dots, s_m\}$ ,然后使用一个属性值 $A_i \in \mathbb{A}$ 来标注每一个文本段 $s_i$  ( $1 \leq i \leq m$ ),这里 $s_i$ 数属性 $A_i$ 中的一个属性值。

知识库: 我们使用实现存在的语料库来构建一个固定领域的知识库(简称为KB),就像 [?, ?]中做的一样。简单讲,知识库(KB)包含一组特定的属性以及对应的属性值,我们记做 $KB = \{ \langle A_1, V_1 \rangle, \langle A_2, V_2 \rangle, \dots, \langle A_n, V_n \rangle \}$ ,这里每一个属性 $A_i$  ( $1 \leq i \leq n$ )都是这个领域属性集 $\mathbb{A}$ 中的一个独立属性, $V_i$ 是一组词语,是在属性 $A_i$ 中出现的典型的、合理的值,并且记录每个属性值在记录中的位置 $\{(v_{i,1}, PS_{i,1}), (v_{i,2}, PS_{i,2}), \dots, (v_{i,n_i}, PS_{i,n_i})\}$ ,这里 $PS_{i,k}$ 是一组位置相关的数值,来表示这个属性值 $v_{i,k}$ 出现在不同记录中的位置属性。一个简化的知识库如图 3-2所示,为了方便理解,这里我们只是使用了一些简化的单词而不是使用原始的属性名。

#### 3.2.1 基于知识库的初始分割和标注

这一模块的主要目的是识别出输入文本中所谓的锚点块(Anchor Blocks),就是对一些文本段,我们会根据知识库计算出来一个标注可信度,当这个可信度大于某个阈值时,便可以直接将这个文本段标注成某一个属性。然后,这些识别出来的锚点块将被视作一个基础锚点,执行后面步骤中的分割和标注。

#### 定义2.

(锚点块) 给定一个输入文本 $I$ ,如果它满足下面的两个条件, $I$ 中的一个文本



$$KB = \{ \langle A, V_A \rangle, \langle B, V_B \rangle, \langle C, V_C \rangle, \langle D, V_D \rangle, \langle E, V_E \rangle, \langle F, V_F \rangle \}$$

$$\begin{aligned} V_A &= \{ (a_1, \{1, 1, 2\}), (a_2, \{1, 2, 2\}), (a_3, \{2, 2\}) \} \\ V_B &= \{ (b_1 b_2, \{4, 4, 5, 5\}), (a_3, \{4\}) \} \\ V_C &= \{ (c_1, \{6, 6, 7\}), (c_2, \{8\}), (c_3, \{6\}), (c_4, \{9\}), (e_1, \{5\}) \} \\ V_D &= \{ (d_1, \{9, 9, 10\}), (d_3, \{10, 11, 11\}), (e_1, \{10\}) \} \\ V_E &= \{ (e_1, \{13, 13\}), (a_1, \{12\}) \} \\ V_F &= \{ (f_1, \{14\}), (f_2, \{15, 16\}), (e_1, \{10\}) \} \end{aligned}$$

图 3-2 一个简化的知识库

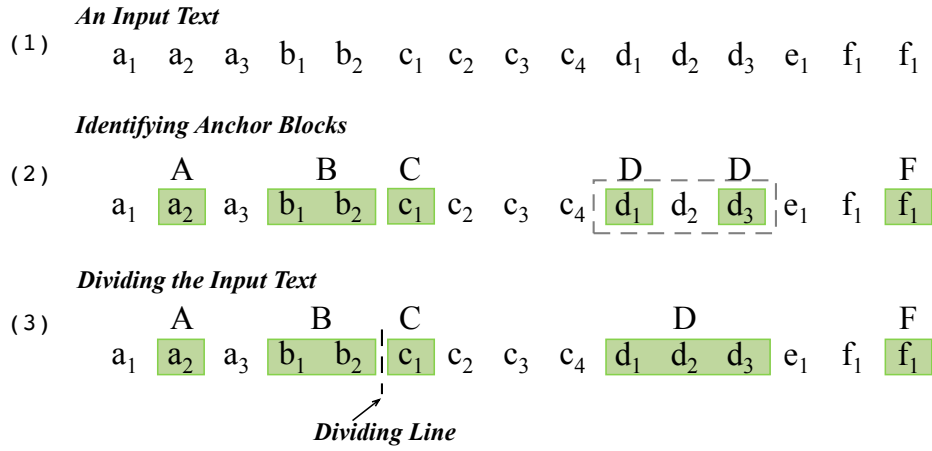


图 3-3 基于知识库的初始分割和标注例子

段 $b$ 才可以被标注成具有属性 $A$ 的锚点块：1)没有任何一个其他属性 $A' \in \mathbb{A}$ 满足 $p(b, A') \geq p(b, A)$ ，这里 $p(b, A)$ 是我们定义的一个方程用来计算一个文本段 $b$ 属于属性 $A$ 的概率。2)  $p(b, A) > \theta$ ，这个阈值 $\theta$ 是我们自己定义的，用来提高锚点块的标注质量。

假设一个文本段 $b$ 出现在知识库中的 $m$ 个不同属性 $\{(A_1, A_2, \dots, A_m)\}$ 的属性值里面，这里 $A_i \in \mathbb{A}$  ( $1 \leq i \leq m$ )表示 $b$ 的一个可能的属性。对于每一个 $A_i$ ，假设 $b$ 出现在一组不同的属性值中，并且具有不同的出现频率，我们记做 $\{(v_{i,1}, f_{i,1}), (v_{i,2}, f_{i,2}), \dots, (v_{i,r_i}, f_{i,r_i})\}$ ，假设所有属性值都是互相独立的，则我们可以通过下面的公式，计算出这个文本段 $b$ 属于属性 $A_i$ 的概率为：

$$p(b, A_i) = \frac{1 + \sum_{1 \leq j \leq r_i} \sum_{1 \leq x \leq f_{i,j}} \frac{1}{x}}{1 + \sum_{1 \leq k \leq m} \sum_{1 \leq j \leq r_k} \sum_{1 \leq x \leq f_{k,j}} \frac{1}{x}} \quad (3.1)$$

这里，我们在分母使用加1的操作是为了防止分母出现为零的情况。有了这个公式和我们定义的阈值 $\theta$ ，我们便能够在输入文本 $I$ 中标注出一些锚点块了。因为，我们的方法的抽取质量和效率是对阈值 $\theta$ 敏感的，所以我们会通过后面的实验部分找到一个合适的阈值。

注意，一个单独的属性值可能被分割成了两个或更多个文本块，比如图 3-3中，属性 $D$ 便有两个文本块 $d_1, d_3$ 。考虑到在一个输入文本中每个属性可能有一个属性值，所以，我们可以将这两个锚点块和他们之间所有的词语都合并到一起。这里可能出现的一个冲突是，在一个属性的两个锚点块之间出现了另一个属性的锚点块，通常，一个合适的阈值 $\theta$ 便可以避免出现这种冲突。但是，如果这种情况仍然存在，我们可以通过弃掉冲突锚点块中得分更低的那个来解决这个问题。比如，给定图 3-3(a)中的知识库，对于一个图 3-3(b-1)中输入文本，我们能够识别出下面这些锚点块： $a_2, b_1b_2, c_1, d_1, d_3$  以及 $f_2$ ，就是图 3-3(b-2)中绿色方格这些文本段。另外，既然 $d_1$ 和 $d_3$ 都是属性 $D$ 的锚点块，我们合并这三个块 $d_1d_2d_3$ ，并共同标注为属性 $D$ ，如图 3-3(b-3)所示。

### 3.2.2 基于卷积神经网络模型的构建和文本块的标注

为了处理一些未标注的文本段，比如上述例子中第一步产生的 $a_1, a_3$ 和 $c_2$ ，我们需要执行一个基于卷积神经网络模型分类结果的概率标注。简要来讲，我们首先使用知识库训练一个卷积神经网络分类器(CNN分类器)，假设 $|\mathbb{A}|$ 表示给定领域中属性的个数，然后对于每一个输入文本 $s$ ，我们的CNN分类器会生成一个 $|\mathbb{A}|$ -维的输出向量 $P(s) = [\rho_1, \rho_2, \dots, \rho_{|\mathbb{A}|}]$ ，这里每一个 $\rho_i$  ( $1 \leq i \leq |\mathbb{A}|$ )表示 $s$ 属于属性 $A_i \in \mathbb{A}$ 的概率。对于每一个未标注的文本段，我们使用我们的CNN模型来决定它属于哪一个属性，然后综合考虑所有未标注文本段的CNN模型输出的概率结果和那些锚点块，得到整个输入文本的最佳分割和标注结果。下面，我们解释如何构建CNN模型，并且介绍如何找到最佳的标注方式。

#### 1. 构建CNN分类模型

图 3-4给出了我们构建的CNN分类模型的架构图，它的输入是一个文本段，输出一个关于所有可能的属性的概率向量。整体来看，我们的CNN模型包括特征输入层，

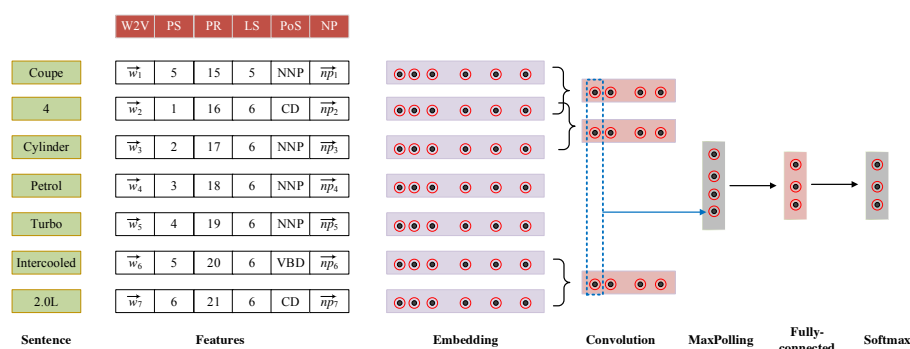


图 3-4 CNN分类模型的架构图

嵌入层，卷基层，最大值池化层，全连接层和最后的Softmax分类器层。特征输入层接受一个输入文本段的特征矩阵，这个特征矩阵包含了丰富的句法和语义特征。嵌入层负责将这些特征值对应到相应的特征向量然后将他们合并在一起。然后我们在特征矩阵上执行卷积操作来提取关于输入文本段的特征，在这一步我们使用了不同大小的过滤核来在输入文本段上尽可能多的覆盖n-gram的文本范围，这样我们就可以生成多层次的，且不同尺寸的特征矩阵。然后，我们再使用最大值池化来对卷积层的结果进行标准化，使其最终都成为相同形状的特征矩阵，且这一步也是进一步抽取特征和降维的操作。池化层得到的特征矩阵再通过一个全连接层，就是一个普通的神经网络，做进一步的特征组合和推理。最终，我们使用Softmax分类器作为我们的输出层，为每一个输入文本段产生分类结果。更多的细节，可以去参考[?,?]

特征:我们为每个输入文本段都构造了一些列丰富的语义特征，包括一些合成的特征。对于文本段中的每一个单词，他的特征矩阵中包括了，提前训练的词向量Word2vec，文本段的位置信息，文本段在输入文本中的位置信息，词性标注信息(POS)，文本段长度，以及一种人为构造的归一化类别概率。下面给出每一个特征的详细介绍：

- 1) **Word2Vec:** 对于词向量，我们直接使用文章 [?]中用 [?]词向量模型在Pubmed文章上提前训练好的词向量，这个词向量模型是由Tomas Mikolov在Google发明，我们在之前的章节有详细介绍。对于中文实验集，我们在自己的中文语料上使用Google的word2vec工具<sup>1</sup>训练了一个词向量。

<sup>1</sup><https://code.google.com/archive/p/word2vec/>

- 2) **在文本段中位置特征:** 这个位置特征是指一个单词在输入文本段中的位置, 我们之所以使用这个特征, 是因为我们观察到, 在一些特定的属性值中, 一些特殊的词语会出现在相同的位置上。
- 3) **在输入文本中的位置特征:** 这个位置特征是指一个单词在原始输入文本中的位置, 我们选择这个特征是因为一个单词在文本中的所在位置往往与它属于哪个属性有密切的关联, 这个一个比较容易想到的特征项。
- 4) **文本段的长度:** 也就是文本段中单词的个数, 使用这个特征是因为不同属性的属性值的平均长度往往是有差别的, 这说明, 文本段的长度是一个非常有区分性的特征。
- 5) **语法和词性标注特征(POS):** 构造POS特征的过程, 对一句话中的每个单词, 依赖这个单词的词性定义和上下文来给它标注一个标签, 用以说明这个单词在这句话中的角色。这里, 我们使用GENIA tagger<sup>2</sup>来得到每个词的POS特征。
- 6) **归一化类别概率:** 我们同样使用公式 3.1 来计算一个概率向量  $c\vec{p}$ , 向量每一维的值表示一个单词属于某个属性的概率。

### 3.2.3 输入文本的贪婪式概率标注

### 3.2.4 基于序列和位置特征的结果修正

## 3.3 实验结果与分析

## 3.4 本章小结

对于给定两个关系表  $T_1 = \{t_1, t_2, \dots, t_n\}$  和  $T_2 = \{s_1, s_2, \dots, s_m\}$ ,  $T_1$  中的每一个实体  $t_i \in T_1 (1 \leq i \leq n)$ ,  $T_2$  中的每一个实体  $s_j \in T_2 (1 \leq j \leq m)$ , 实体匹配旨在发现两个表间表示同一实体的实体对  $(t_i, s_j)$ 。以往的实体匹配方法主要是使用主属性的值, 依赖于一些字符串相似度函数 (如编辑距离) 以及一个相似度阈值决定两个表中的实体是否匹配。其形式化表示如下所示: 设定  $A_0$  是表  $T_1$  和表  $T_2$  的主属性, 给定一个字符串相似度函数  $\text{sim}(\cdot, \cdot)$ , 以及一个相似度阈值  $\theta$ , 对于实体对  $(t_i, s_j)$ , 如果满足  $\text{sim}(t_i[A_0], s_j[A_0]) \geq \theta$ , 则  $t_i$  和  $s_j$  为同一实体。

<sup>2</sup><http://www.nactem.ac.uk/GENIA/tagger/>

然而，仅仅使用主属性值判定实体对是否匹配是不够的，因为主属性值通常具有多种表现形式，如姓名为“Li Hua”的实体和姓名为“Hua Li”的实则表示同一个人。又或是主属性值直接无法使用，如丢失数据和噪声数据。本章中，我们使用结构化非主属性值进行实体匹配。假定表 $T_1$ 和表 $T_2$ 具有相同的非主属性集合 $S_{NK} = \{A_1, A_2, \dots, A_p\}$ ，使用非主属性进行实体匹配的定义如下：

### 定义3.

非主属性的实体匹配(*EM with Non-Key Attributes (NokeaEM)*): 给定两个数据表 $T_1 = \{t_1, t_2, \dots, t_n\}$ 和 $T_2 = \{s_1, s_2, \dots, s_m\}$ ，二者具有相同的结构化非主属性集合 $S_{NK} = \{A_1, A_2, \dots, A_p\}$ ，*NokeaEM*问题旨在找到一个基于 $S_{NK}$ 的函数 $F(t_i, s_j)$ 和一个相似度阈值 $\tau$ ，使得 $\forall t_i \in T_1 (1 \leq i \leq n)$ ， $\forall s_j \in T_2 (1 \leq j \leq m)$ ， $(t_i, s_j)$ 为指向同一实体的实体，当且仅当它们满足：

$$(1) F(t_i, s_j) \geq \tau;$$

$$(2) \forall s_k \in T_2, F(t_i, s_j) \geq F(t_i, s_k)$$

## 3.5 预备工作

使用结构化非主属性进行实体匹配并非易事，其中面临着众多困难与挑战。与主属性相比，结构化非主属性中存在更多的噪声值，而且数据间的不一致问题更为严重。通常情况下数据集中的结构化非主属性个数要比主属性个数多，使得基于非主属性的匹配算法存在着严重的匹配效率问题。当使用结构化非主属性进行实体匹配时至少面临三个挑战：

(1) 非主属性选择：面对众多的结构化非主属性，需要考虑如何对每一个结构化非主属性识别一实体不同于其它实体的能力进行评估，从而选择识别能力高的结构化非主属性进行匹配。

(2) 噪数据以及丢失数据的影响：结构化非主属性中可能存在噪数据和丢失数据，这些噪数据和丢失数据影响实体匹配的结果，因此如何处理数据集中的噪数据和丢失数据也是一个很大的问题。

(3) 实体匹配效率问题：基于结构化非主属性的实体匹配方法与传统的只使用主属性的实体匹配方法相比，它的时间开销要多出很多倍，因为基于结构化的实体匹配方法需要比较更多的非主属性值。

我们发现文献 [?]与本章的工作很相似，提出了一种基于树结构的方法进行实体匹配，该方法使用结构化非主属性创建类似于决策树的Matching tree 进行实体匹配，Matching tree 是一种基于0-1结构的二分树，通过一定的概率计算方式选择属性作为Matching tree 的节点，然后从根节点到叶节点遍历进行匹配，但是这种方法存在适用性问题，当属性值出现缺失时，此次的匹配过程便无法正常进行。但是本章提出的方法却能很好的处理该问题，在遍历过程中若出现缺失值就会产生独立的分支，继续遍历具有缺失值属性下的其它节点，进行后续的匹配。此外，Matching tree-based 方法没有考虑Matching tree 中不同层次的结构化非主属性对匹配结果的影响程度，而粗糙地将靠近根节点的结构化非主属性给予较大的权重，将靠近叶子节点的结构化非主属性对匹配结果给予较小的权重，但实际情况中应该根据节点所在的层次给予合适的权重分配。本章提出的基于结构化非属性的概率决策树的实体匹配算法通过平衡函数合理地分配非主属性的权重，解决了上述问题。

### 3.6 基于非主属性区分度的实体匹配算法

基于非主属性区分度的实体匹配算法（baseline 算法）主要考虑使用结构化非主属性区分一实体不同于其他实体的能力，结构化非主属性的这种能力被定义为属性的区分度得分（Differentiation Degree Score，简称difScore）。属性的difScore 大致上反应了该结构化非主属性下的属性值分布情况。例如，在没有重复实体的情况下数据表中主属性值都是不同的，因此主属性具有最高的difScore 也即是1.0。通过分析可以发现，在表 ??中结构化非主属性“Size”具有较高的difScore，因为该非主属性值的分布比较多样，而非主属性“RAM”则具有一个相对较低的difScore。结构化非主属性 $A_i (1 \leq i \leq p)$ 的difScore 具体计算方法如下所示：

$$difScore(A_i, T) = \frac{distinct(A_i, T)}{|T|} \quad (3.2)$$

其中 $|T|$ 表示数据集 $T$ 中实体的总个数， $distinct(A_i, T)$ 表示结构化非主属性 $A_i$ 下不同

的属性值个数。当获得关系表中每个结构化非主属性的difScore后，使用如下的公式计算实体 $t \in T_1$ 和 $s \in T_2$ 之间的相似度：

$$\mathcal{F}_{baseline}(t, s) = \frac{\sum_{A \in S_{NK}} (difScore(A) \times sim(t[A], s[A]))}{\sum_{A \in S_{NK}} difScore(A)} \quad (3.3)$$

其中 $S_{NK}$ 表示两表中共同的非主属性集合， $t[A]$ 、 $s[A]$ 分别表示实体 $t$ 和 $s$ 在非主属性 $A$ 下的属性值， $sim(t[A], s[A])$ 是指属性值 $t[A]$ 和 $s[A]$ 之间的相似度。

然而，通过实验发现，本章提出的baseline算法只能识别出一半左右的匹配实体（大约45-55%的召回率）。通过分析发现其原因是：一方面，difScore并不能完全反应非主属性的区分度，因为该度量方式并未考虑属性之间的组合关系；另一方面，丢失数据影响了相似度的计算。更为重要的是baseline算法的复杂度为 $O(pnm)$ ，其中 $|T_1| = n$ ,  $|T_2| = m$ ,  $|S_{NK}| = p$ 。

### 3.7 基于规则的概率决策树的实体匹配算法

为了解决baseline算法中的问题，考虑到决策树能够提高效率以及表示节点重要性的优点，具体体现在树中的每一个节点都有机会对输入的数据做出最终的决策，而且不同层次的节点具有不同的重要性，本章提出了一个更高级的基于树结构的实体匹配算法，该算法使用结构化非主属性建立基于规则的概率决策树（Probabilistic Rule-based Decision Tree，简称PRTree）进行实体匹配，期望能够在尽可能早的时间内获取匹配实体对或者排除不匹配实体对。通过PRTree可以很好地克服baseline算法的缺陷，因为该结构既考虑了非主属性之间的组合关系，又不易受缺失值的影响。接下来，首先介绍该算法的基本思路，然后具体介绍如何使用非主属性建立PRTree，并给出使用该PRTree的NokeaEM算法，最后通过实验与其他已有实体匹配算法比较验证baseline算法和PRTree算法的效果。

#### 3.7.1 算法基本思路

对于给定的实体对 $t_i \in T_1$ ,  $s_j \in T_2 (1 \leq i \leq n, 1 \leq j \leq m)$ ，它们的匹配结果是由其各个非主属性下的属性值是否匹配所决定的。考虑到不同的非主属性在唯一确定实体能力上的不同，本章考虑选择一组具有最高决策能力的属性集进行实体匹配。此外，为了尽可能地减少实体匹配的匹配开销，我们期望能够尽早发现匹配的实体或

者尽早排除那些不匹配实体，因此创建了一种具有类似于决策树结构特征的结构进行实体匹配，将各个非主属性放入决策树的不同节点中对实体进行逐层联合判定最终确定匹配结果；此外，通过分析发现非主属性集中各个非主属性在决定匹配结果时并不是相互独立的，而是存在层次上的相互依赖关系，因此在创建PRTree的过程中以及使用PRTree进行匹配时还需要考虑非主属性之间的依赖度。

我们对这种依赖关系进行了可视化分析，如图 3-5所示，实体的匹配结果 $R$ 是由非主属性集 $S = \{A_1, A_2, \dots, A_5\}$ 中的属性共同决定的。以路线1为例，可以发现属性 $A_2$ 的结果是在属性 $A_1$ 的条件下而得到的，而属性 $A_4$ 的结果是在属性 $A_2$ 的条件下而得到的，其他属性以此类推。最终所得结果是在以上各步的结果基础上综合作用得到的。

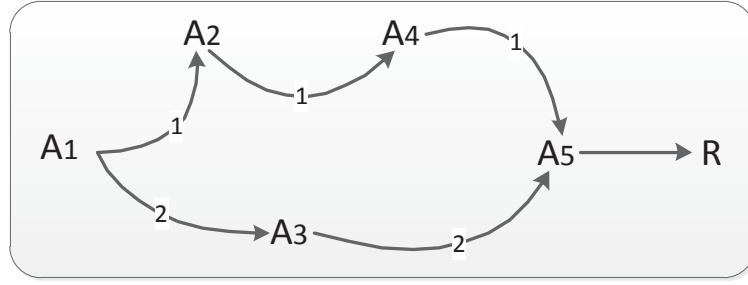


图 3-5 属性间的相关性

为了方便起见，我们使用非0即1的方式表示实体 $t_i$ 与 $s_j$ 的匹配结果，其形式化表示为：

$$R = \begin{cases} 1, & \text{if } t_i = s_j \\ 0, & \text{otherwise} \end{cases} \quad (3.4)$$

此外，我们用 $M_k (1 \leq k \leq p)$ 表示实体所在非主属性组合下的第 $k$ 个非主属性对于匹配结果 $R$ 的一个贡献，如果二者具有相同或者相似的值则认为是一个正的贡献，否则认为是一个负的贡献。具体计算方式如下所示：

$$M_k = \begin{cases} 1, & \text{if } t_i[A_k] = s_j[A_k] \\ 0, & \text{otherwise} \end{cases} \quad (3.5)$$

对于关系表中的非主属性的集合 $S_{NK} = \{A_1, A_2, \dots, A_p\}$ ，实体间的匹配结果可通过



条件概率进行量化，其概率可通过贝叶斯条件概率进行评估，其公式如下所示：

$$p(S_{NK}) = Pr(R = 1|S_{NK}) = \frac{Pr(S_{NK}|R = 1)Pr(R = 1)}{Pr(S_{NK}|R = 1)Pr(R = 1) + Pr(S_{NK}|R = 0)Pr(R = 0)} \quad (3.6)$$

### 3.7.2 基于规则的概率决策树的构建

为了便于展示PRTree的构建过程以及如何在创建的PRTree上进行实体匹配，我们给出了数据集的部分数据，如表3-1和表3-2所示。

表 3-1 来自天猫（Tmall）中的手机信息

	Product Name	Manufacturer	Size	RAM	Release	Type	OS	...
t1	w2013	SAMSUNG	3.7 inches	1GB	2013.04	Flip	Android 4	...
t2	8295	Coolpad	4.7 inches	1GB	2013.01	Bar	Android 4.1	...
t3	MXII	MeiZu	4.4 inches	2GB	2012.12	Bar	Flyme 2	...
t4	Ericsson U1	Sony	3.5 inches	512MB	2009.10	Bar	S60V5	...
t5	4S	Apple	3.5 inches	512MB	2011.01	Bar	IOS 5	...
t6	A880	Lenove	6 inches	1GB	2013.04	Bar	Android 4.2	...
t7	Ascend M2	HuaWei	6.1 inches	2GB	2014.03	Bar	Android 4.3	...
t8	S930	Doov	2.8 inches	512MB	2010.09	Slide	-	...
t9	G9098	SAMSUNG	3.67 inches	2GB	2014.09	Bar	Android 4.3	...
t10	8730L	Coolpad	5.5 inches	1GB	2012.05	Bar	Android 4.3	...
...	...	...	...	...	...	...	...	...

表 3-2 来自太平洋电脑（Pconline）中的手机信息

	Product Name	Manufacturer	Size	RAM	Release	Type	OS	...
s1	Galaxy w2013	SAMSUNG	3.7 ”	1GB	2013-04	Flip	Android 4.0	...
s2	8295	Coolpad	4.7 ”	-	2013-01	Bar	Android 4.1	...
s3	Meizu MX2	MeiZu	4.5 ”	2.0G	2012-04	-	Flyme 2.0	...
s4	3s	XIAOMI	5.1 ”	3G	-	Bar	MIUI V5	...
s5	IPhone 4s	Apple	-	512MB	2011-10	Bar	IOS 5.0	...
s6	A880	Lenove	6.0 ”	1G	2013-04	Bar	Android 4.1	...
s7	Mate2	HuaWei	6.1 ”	2GB	2014-03	Bar	Android 4.3	...
s8	S930	Doov	2.8 ”	512M	2010-09	Slide	-	...
s9	8730L	Coolpad	-	1G	2012-05	Bar	Android 4.3	...
...	...	...	...	...	...	...	...	...

基于第3.7.1节的分析，我们给出PRTree的理论依据：（1）决策树的结构有利于减少实体匹配过程中使用的属性数目，进而减少匹配的时间开销；（2）贝叶斯概率决策树能够反应具有一定决策能力的非主属性之间的依赖关系；（3）不同的非主属性在匹配过程中具有不同的决策能力，表现为非主属性的两个重要性质，即决策匹配的充分性（Sufficiency）以及决策匹配的必要性（Necessity），其定义如下：

**定义4.**

决策匹配的充分性和必要性 (*Sufficiency and Necessity*): 对于给定的两个关系表  $T_1 = \{t_1, t_2, \dots, t_n\}$  和  $T_2 = \{s_1, s_2, \dots, s_m\}$ , 假定二者具有相同的非主属性集合  $S$ , 属性  $A_k \in S (1 \leq k \leq p)$  的充分性概率是指当满足  $t_i[A_k] = s_j[A_k]$  时, 实体  $t_i \in T_1 (1 \leq i \leq n)$  和  $s_j \in T_2 (1 \leq j \leq m)$  为同一实体的概率, 而属性  $A_k$  的必要性概率是指当  $t_i$  和  $s_j$  为同一实体时, 也即当  $t_i = s_j$  时,  $t_i[A_k] = s_j[A_k]$  的概率。

非主属性的充分性特性和必要性特性在构建PRTree时起着重要的决定作用, 非主属性的充分性反应了属性识别匹配实体的能力, 即当实体的非主属性值相同时, 能够在多大概率上判定其所对应的实体为同一实体; 而非主属性的必要性则反应了属性排除不匹配实体的能力, 即当实体对表示为同一实体时, 实体对在某一属性上也具备相同或相似属性值的概率。

根据定义4所述, 属性的充分性概率和必要性概率可通过训练得到, 其中训练集中的已知两个数据表中匹配的实体对, 其定义为  $(Tr_1, Tr_2)$ 。假定训练集中的数据分布接近于实体匹配的两个真实数据表, 则我们可以得到充分性特性和必要性特性的计算公式如下:

$$suf(A_k) = \sum_{\{(t_i, s_j) | Tr_c\}} \frac{B(t_i = s_j | t_i[A] = s_j[A])}{B(t_i = s_j | t_i[A] = s_j[A]) + B(t_i \neq s_j | t_i[A] = s_j[A])} \quad (3.7)$$

$$nec(A_k) = \sum_{\{(t_i, s_j) | Tr_c\}} \frac{B(t_i[A] = s_j[A] | t_i = s_j)}{B(t_i[A] = s_j[A] | t_i = s_j) + B(t_i[A] \neq s_j[A] | t_i = s_j)} \quad (3.8)$$

其中, 函数  $B(bool_1 | bool_2)$  为布尔函数, 其具体表示如下所示:

$$B(bool_1 | bool_2) = \begin{cases} 1, & \text{if } bool_2 = \text{TRUE and } bool_1 = \text{TRUE} \\ 1, & \text{if } bool_2 = \text{TRUE and } bool_1 = \text{FALSE} \\ 0, & \text{if } bool_2 = \text{FALSE} \end{cases} \quad (3.9)$$

其中,  $Tr_c$  是一个在建立PRTree的不同阶段中动态改变的限制集, 其初始化为:

$$Tr_c = \{(t_i, s_j) | t_i \in Tr_1, s_j \in Tr_2\}。$$

下面我们将详细讨论如何建立PRTree 匹配树结构。根据上节中的描述，实体对的相似度计算需要考虑属性间的相互依赖关系，因此我们并不是一次找出top-k 个具有较大difScore 的非主属性作为概率决策树上的非主属性节点，而是使用贪心算法，每次只在上一步的决策结果的条件下，计算未使用过的非主属性的条件充分性概率和必要性概率，从而选择出当前条件下具有最大决策力的非主属性放入树中的对应位置。该建树过程将迭代地进行下去，直至达到一定的停止条件。基于上述思路，以下介绍建立PRTree 的过程中的两个重要问题：一方面是属性的选择，另一方面是停止条件的设定。

(1) 属性选择：首先根据当前待选节点的类别（包括根节点、非叶节点和叶节点），根据已经确定的候选节点集每次选择具有最大充分性或最大必要性的属性作为当前节点，然后将该属性加入到候选节点集中。通过上述选择过程，能够尽可能早地发现匹配实体，排除不匹配实体。具有较大充分性概率的非主属性说明该属性识别实体对匹配的能力较强，而具有较大必要性概率的非主属性表明该属性在排除不匹配实体的能力较强。

(2) 停止条件：停止条件的设定是为了防止建立的树产生过拟合现象，当发现所有的待考察属性在当前节点位置的条件充分性概率或条件必要性概率都低于某一设定的阈值（ $\tau_{stop}$ ）时，此时应停止创建更多的节点，然后将上一轮刚加入候选节点集的非主属性作为树的叶子节点。

PRTree 中共包含三类节点，根节点、非叶节点和叶节点，各类节点都有其特性。下面详细给出了PRTree 的创建过程：

(1) 根节点的创建（Root Node）：当计算得出每个非主属性的充分性概率和必要性概率后，我们选择具有最大充分性和必要性得分的属性作为根节点。对于根节点设定三个分支，即匹配分支（Matched (Y)），不匹配分支（Unmatched (N)）和无效分支（Invalid (Null)）。如果根节点的类型是具有最高充分性得分的非主属性，则节点的匹配分支是叶节点，并将以一定的置信度成为“Matched”分支，而“Unmatched”分支以及“Invalid”分支则是非叶节点。相反地，如果根节点是具有最大必要性得分的非主属性，节点的“Unmatched”分支则成为叶节点，并将以一定的置信度成为“Unmatched”分支，而“Matched”分支和“Invalid”分支都应是而非叶

节点。PRTree 的根节点的计算公式如下所示：

$$P_{root} = \max_{A \in S_{NK}} (suf(A), nec(A)) \quad (3.10)$$

(2) 非叶节点的创建 (Non-Leaf Node): PRTree 的非叶节点表示该节点的充分性概率或必要性概率较低不能直接判断出匹配的结果, 需要结合其它的非主属性, 直至能够做出判断为止, 即需要到达PRTree 的根节点。我们在目前已经确定的候选节点集的基础上, 在计算剩余的其它非主属性的充分性概率和必要性概率时, 必须要在满足候选集中所有元素的条件下进行计算。为了创建PRTree 的非叶节点, 在满足目前已有的所有祖先节点集的条件下计算剩余非主属性的充分性概率和必要性概率, 选择具有最大条件充分性概率或条件必要性概率的非主属性作为当前非叶节点。非主属性的条件充分性和条件必要性可通过公式 3.7 和 3.8 计算所得, 并将 $Tr_c$ 改变为一个更小的训练集, 其中所有的实体都应满足祖先节点的条件。非叶节点的计算方法如公式 3.11所示:

$$P_{nonleaf} = P_{A|Candit(PRTree)} \max_{A \in (S_{NK} - Candit(PRTree))} (suf(A), nec(A)) \quad (3.11)$$

其中 $Candit(PRTree)$ 表示已选中的祖先节点所构成的集合。

(3) 叶节点的创建 (Leaf Node): PRTree 的叶节点表示从根节点到当前的节点构成的属性集合可以判断给定的实体对是否匹配或者目前已有的数据无法给出明确的结果。PRTree 的每个叶节点会输出“Matched”结果或“Unmatched”结果, 通过输出结果我们可以判定实体对是否匹配。当非主属性的 $suf(A)$ 和 $nec(A)$ 一直很低时, 即其值低于一个合理的阈值 $\tau_{stop}$ 时, 我们将停止获取更多的非主属性用以创建PRTree的节点。然后, 将创建一个“Exit (Not Decided)”节点作为最后的节点, 该节点表示无法判断实体对是否匹配。叶节点的计算方法如公式 3.12所示:

$$P_{leaf} = \max_{A \in (S_{NK} - Candit(PRTree), suf(A) \approx \tau_{stop}, nec(A) \approx \tau_{stop})} (suf(A), nec(A)) \quad (3.12)$$

利用非主属性的充分性概率和必要性概率创建PRTree的方法如算法 1所示。通过对数据集中大量数据的训练, 我们建立了如图 3-6所示的不包含于非主属性

的PRTree。在接下来的部分中，我们将介绍如何使用PRTree进行实体匹配。

---

**算法 1: BuildPRTree: 构建类决策树**


---

**Input** : 具有属性集 $S$ 的训练表 $Tr_1$ 和 $Tr_2$ , 充分性阈值 $\tau_s$ 和必要性阈值 $\tau_n$ 以及不再继续创建PRTree的阈值 $\tau_{stop}$

**Output**: PRTree的根节点的引用 $root$

```

1 初始化 $root = null, cur = root, AncsCond = \emptyset$ ;
2 while  $S \neq \emptyset$  do
3   foreach  $A \in S$  do
4     根据公式 3.7和公式 3.8计算或更新每个模式当前条件下
       的 $suf(A|AncsCond)$ 和 $nec(A|AncsCond)$ ;
5     if  $A.suf < \tau_s$  then  $A.suf = 0$  if  $A.nec < \tau_n$  then  $A.nec = 0$ 
6     在所有的非主属性中将具有最大值的 $suf(A|AncsCond)$ 或 $nec(A|AncsCond)$ 
       属性 $A$ 赋给 $cur$ ,  $cur = A$ ;
7      $cur.conf = \text{Max}(suf(A|AncsCond), nec(A|AncsCond))$ ;
8     if  $cur.conf < \tau_{stop}$  then
9        $cur = \text{"Exit(NotDecided)"};$ 
10      Break;
11     $cur.NullChild = \text{BuildPRTree}(Tr_1, Tr_2, S = S - \{A\}, \tau_s, \tau_n, \tau_{stop})$ ;
12    if  $cur.conf = suf(A|AncsCond)$  then
13       $cur.YChild = \text{"Matched"};$ 
14       $AncsCond = AncsCond + \{(A, unmatched, cur.suf)\};$ 
15       $cur = cur.NChild$ ;
16    else
17       $cur.NChild = \text{"Unmatched"};$ 
18       $AncsCond = AncsCond + \{(A, matched, cur.nec)\};$ 
19       $cur = cur.YChild$ ;
20 return  $root$ ;
```

---

### 3.7.3 实体匹配过程

在算法 2中我们给出了如何使用建立的PRTree执行NokeaEM算法。测试集数据集中的每一个实体对 $(t, s)$ ，我们从PRTree的根节点开始访问，并在到达叶子节点时停止。每次当实体对 $(t, s)$ 到达某一非主属性 $A_k$ 节点时，检查实例对在当前节点 $A_k$ 是否具有相同或相似的属性值。如果相同或相似，则访问 $A_k$ 节点的“Matched”子节点，否则访问 $A_k$ 节点的“Unmatched”子节点。但是，如果实体对在节点 $A_k$ 的非主属性值为空时，则应访问该“Invalid”子节点，表明待比较的实体对在属性 $A_k$ 下出现了缺失值。当实体对访问至PRTree的叶节点时，将会根据叶节点的性质从而决定输出

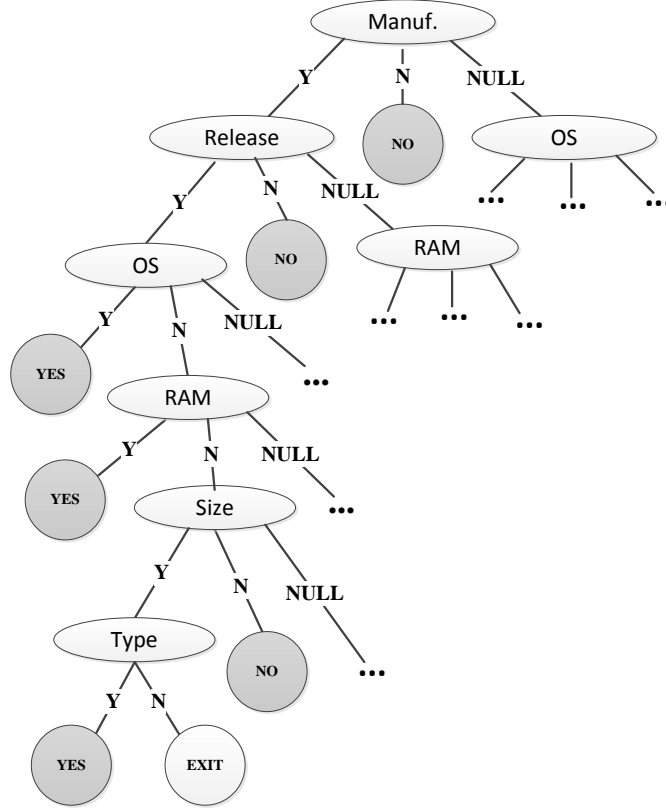


图 3-6 根据算法 1 建立的不包含主属性的 PRTree

“Matched” 或者 “Unmatched”。决策的置信度是由实体对所经过的从根节点到当前节点组成的所有非主属性节点组合共同决定的。除根节点之外，PRTree 中每一个节点的结果的计算并不是独立于其它的节点而得到的，而是在满足当前节点所经历的所有祖先节点的条件下所得到的。此外，为了达到更好的匹配效果，PRTree 中处于不同高度的节点，在计算置信度时应给与不同的权重分配，因为越是靠近根节点的非主属性，受到其它非主属性的影响就越小，而越是远离根节点的非主属性，受到其他属性的影响就越大。为此，本章中采用了一定的方式保证非主属性权重的合理分配。综合以上的考虑我们根据如下公式 3.13 计算实例对之间的相似性：

$$\mathcal{F}_{PRTree}(t, s) = \prod_{1 \leq i \leq H(A)} (Conf(A_i) * sim(t[A_i], s[A_i]))^{\frac{1}{i}} \quad (3.13)$$

其中， $H(A)$  表示当前节点在 PRTree 结构中的高度，使用  $(\cdot)^{1/i}$  调整第  $i$  个节点对匹配结果的影响，因为越接近当前节点的节点，对当前节点的决策影响越大。当节点所

处的层次越低，其削弱的力度也就越大；相反地，当节点所处的层次越高，其削弱的力度也就越小。由于高层次的节点是在低层次节点的基础上创建的，即说高层次的节点依赖低层次的节点，对整棵PRTree的影响较低层次节点影响小，故应给与较低的削弱力度；相反地，对于高层次的节点具有类似的原理。我们使用 $Conf(A_i)$ 表示路径节点 $A_i$ 的置信度，其计算方法如公式 3.14 所示：

$$Conf(A_i) = \begin{cases} suf(A_i), & \text{if } A_i\text{'s "Matched" branch is in passed} \\ 1 - nec(A_i), & \text{if } A_i\text{'s "Unmatched" branch is passed} \\ 1, & \text{if } A_i\text{'s "Invalid" branch is passedd} \end{cases} \quad (3.14)$$

通过上述的步骤结合公式 3.7至公式 3.14可以计算出给定实体对的相似度，并通过与相似度阈值的比较判断给定的实体对是否表示为同一实体。

---

**算法 2:** 使用PRTree 实现实体匹配的NokeaEM 算法.

---

**Input :** PRTree 的根节点 $root$ , 实体对 $(t, s)$

**Output:** 实体对 $(ts)$  的相似度:  $FPRTree(t, s)$

---

```

1   $cur = root; level = 1; sim = 1;$ 
2  while true do
3      if  $cur.output = "Exit"$  then
4          return -1; // 表明算法不能给出一个适当的相似度
5      if  $cur.isleafNode$  then
6          return  $sim$ ;
7      if  $t[cur] = null$  or  $s[cur] = null$  then
8           $cur = cur.NullChild;$ 
9      else
10          $sim = sim * (cur.conf * sim(t[cur], s[cur]))^{\frac{1}{level}};$ 
11         if  $t[cur] = s[cur]$  then
12              $cur = cur.YChild;$ 
13         else
14              $cur = cur.NChild;$ 
15          $level ++;$ 
16 return  $similarity;$ 

```

---

下面以表 3-1 中实体 $t_1$  和表 3-2 中的实体 $s_1$  为例介绍如何使用创建的PRTree 进行实体匹配。

(1) 根据公式 3.7—公式 3.14通过不断地迭代运算，找出每一次迭代中符合要求的候选非主属性，得到了如图 3-6所示的PRTree。从图中可以看出“Manufacturer”具有较大的必要性得分，也即说对于手机实体而言当“Manufacturer”不同或者不相似时，两个手机实体表示同一实体的概率基本为0；当“Manufacturer”和“Release”具有相同或相似的值时，“OS”具有较大的充分性概率，如果“OS”的值相同或相似，则可判定这两个手机为同一实体，如果“OS”不相同或不相似，则需要从“OS”的“N”分支继续寻找符合要求的其它非主属性。

(2) 在第一步完成后，利用公式 3.14，我们得到了图 3-6 中节点的Conf值，如表 3-3所示。

(3) 对于给定的表 3-1中的实体 $t_1$  和表 3-2中的实体 $s_1$ ，根据算法 2描述如何使用图 3-6中的PRTree 进行实体的匹配，文本中设定非主属性值的编辑距离相似度阈值为0.7，实体的相似度阈值为0.55。首先从根节点“Manufacturer”开始遍历PRTree，由于 $\text{sim}(t_1[\text{Manuf.}], s_1[\text{Manuf.}]) = 1 > 0.7$ ，所以遍历“Manufacturer”的“Matched”子节点即“Release”；因为 $\text{sim}(t_1[\text{Release}], s_1[\text{Release}]) = 0.86 > 0.7$ ，因此遍历“Release”的子节点即“OS”；接下来继续计算 $\text{sim}(t_1[\text{OS}], s_1[\text{OS}]) = 0.86 > 0.7$ ，所以我们到达“OS”的“YES”节点也即是PRTree 的叶子节点；最终，通过公式 3.13结合表 3-3中节点的conf 值，计算实体 $t_1$  和实体 $s_1$  的相似度： $\text{sim}(t_1, s_1) = (1 * 0.94)^1 * (0.89 * 0.86)^{\frac{1}{2}} * (0.92 * 0.82)^{\frac{1}{3}} \approx 0.7485 > 0.55$ 。因此，通过比较认定实体 $t_1$  和实体 $s_1$  二者是匹配的，也即 $t_1$  和 $s_1$  为同一实体。

表 3-3 图 3-6中节点的置信度

Node	Manufacturer	Release	OS	RAM	Size	Type
Confidence	0.94	0.89	0.92	0.81	0.83	0.76

#### 3.7.4 基于PRTree 的NokeaEM 算法复杂度分析

基于PRTree 的NokeaEM 算法相对于baseline 算法极大地减少了匹配的时间开销，因为基于PRTree 的NokeaEM 算法使用尽可能少的非主属性做出了最终的决策，减少了实体对之间非主属性的比较次数，而baseline算法使用了全部的非主属性才能做出决策。算法 2的时间复杂度为 $O(qnm)$ ，其中变量 $q$  表示在匹配过程中每个实体对所到达的平均高度。值得注意的是 $q \ll p$ ，所以可以排除一大部分不匹配的实体对，并在



尽可能早的时间内找到匹配的实体对。

### 3.7.5 基于PRTree的NokeaEM算法的扩充算法

鉴于主属性在唯一标识某一实体的特点，因此可以在一定程度上辅助非主属性进行实体匹配。在本小节将讨论如何将基于PRTree的NokeaEM算法与基于主属性的实体匹配方法（keyEM）相结合进行实体匹配。其中一种可能的方式是分别从keyEM和NokeaEM中获取实体对的相似度权重之和，然而该方式存在的问题是没有一个合适的权重能够满足所有的情形。而另外一种方式是先执行KeyEM或者NokeaEM，并给出一个严格的准确率阈值，然后执行另外一种EM算法用以提高匹配的召回率。然而，通过观察，发现这种方式也不能达到较高的匹配准确率和召回率。最终，我们发现在建立PRTree的过程中将主属性参与进去可以达到最好的性能。通过上述方法，我们见了如图3-7所示的带有主属性的PRTree，该PRTree将主属性“Product”作为其一个节点。我们将在实验部分对这种结合方式所得的PRTree的匹配效果进行评估。

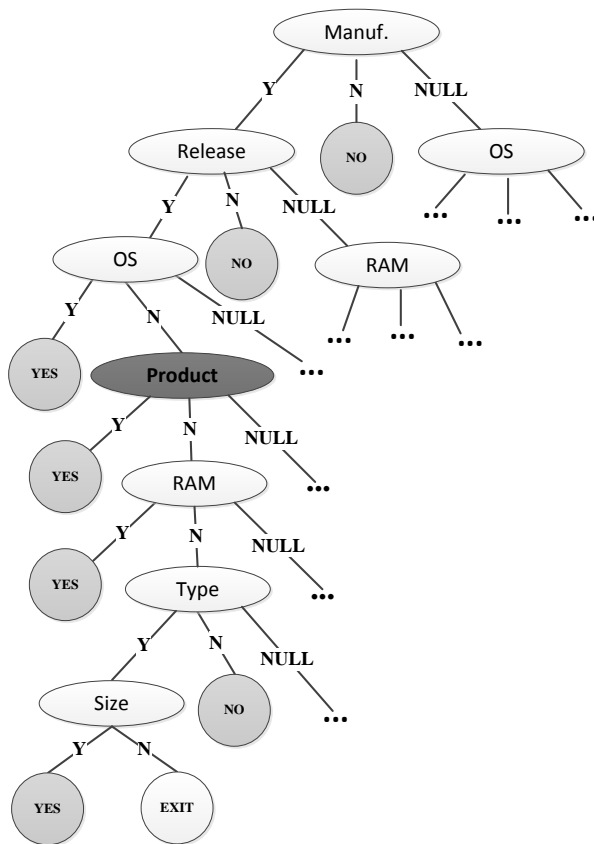


图 3-7 根据算法 1 建立的包含主属性的PRTree

### 3.8 实验结果及分析

我们在两个真实数据集和一个合成数据集上对本章中提出的基于结构化非主属性数据的实体匹配算法进行验证：

(1) 手机数据集 (*Mobile*)：该数据集是从天猫网 (Tmall) 和太平洋电脑网 (PConline) 上搜集的在售手机信息。其中Tmall手机数据集中共有4k个记录，53个属性，而PConline手机数据集中共有5.6k个元组，46个属性。其中这两个数据集所共有的实体个数为3.8k，属性个数为38，部分共有的属性如“Release Date”，“Operation System”，“RAM”，“Screen Size”，“Type”等。手机数据集的性质如表 3-4 所示。

(2) 相机数据集 (*Camera*)：该数据集是从天极网 (Yesky) 和太平洋电脑网 (PConline) 上搜集的在售的数码相机信息。其中Yesky相机数据集中共有2.5k个元组，50个属性，而PConline相机数据集中共有3.4k个元组，44个属性。其中这两个数据集共有的实体个数为2.5k，属性个数为31，部分共有的属性如“Type”，“Pixels”，“Panel”，“Wifi”，“Manufacturer”等。相机数据集的性质如表 3-4 所示。

(3) 仿真数据集 (*Synthetic*)：我们设计生成了两个仿真数据集，其中具有100k个元组，60个共有属性。为了使用该仿真数据集中数据的分布近似于真实数据集我们使用了一定的设计规则，如随机产生一定数目的缺失值，对相同的实体设计相似的非主属性值等，从而使得匹配实体的非主属性值相似度能够统一地分布在0和1之间。仿真数据集的性质如表 3-4 所示。

表 3-4 数据集信息表

	Mobile		Camera		Synthetic
	Tmall	PConline	Yesky	PConline	
#Attribute	53	46	50	44	60
#Record	4k	5.6k	2.5k	3.4k	100k

为了估方法的效果我们使用了3个度量标准评：准确率 (Precision)，是指通过实验获得的所有匹配的实体中正确匹配的实体所占的比例；召回率 (Recall)，是指所有应该匹配的实体中正确匹配的实体所占的比例；F1 Score，是指对准确率和召回率的综合考虑，计算方法为： $F1 = (2 * precision * recall) / (precision + recall)$ 。我们使用算法的时间开销对方法的效率进行评估。

### 3.8.1 参数设定

在所提出的算法中有一些很重要的参数需要设定，如实体相似度阈值，字符串相似度阈值，以及建立PRTree时的阈值等，这些参数影响了提出算法的效果和效率。在对我们提出的算法评估之前，先对其中的一些参数进行说明：（1）创建PRTree所用到的阈值（Thresholds for Building the PRTree）：如图 3-8所示，可以发现F1 Score 随着两个阈值 $\tau_s$  和 $\tau_n$  在0和1之间变动时发生的改变。通过我们的观察发现，当 $\tau_s = 0.8$ ， $\tau_n = 1$  时，F1 score 恰好达到最大值。同时为了防止过拟合的产生，我们也使用一个阈值 $\tau_{stop}$  用于控制树的高度和质量，通过观察发现当 $\tau_{stop} \approx 0.6$  时，PRTree的效果达到最好。注意到在图 3-8（b）中，由于 $\tau_s$  的值太大，导致一些比较重要的非主属性被排除而不能用于创建PRTree，导致了F1 Score 的急剧下降。（2）字符串相似度阈值的设定（Threshold for String Similarity）：通过在三个数据集的实验我们发现最佳的字符串相似度阈值大约为0.7。（3）匹配实体的相似度阈值（Threshold for Matching Instances Similarity）：同时我们也设定了一个用于判定匹配结果的阈值 $\tau_{matched}$ 。通过在三个数据集上的实验发现当 $\tau_{matched} = 0.55$  时，是一个比较合适的取值。

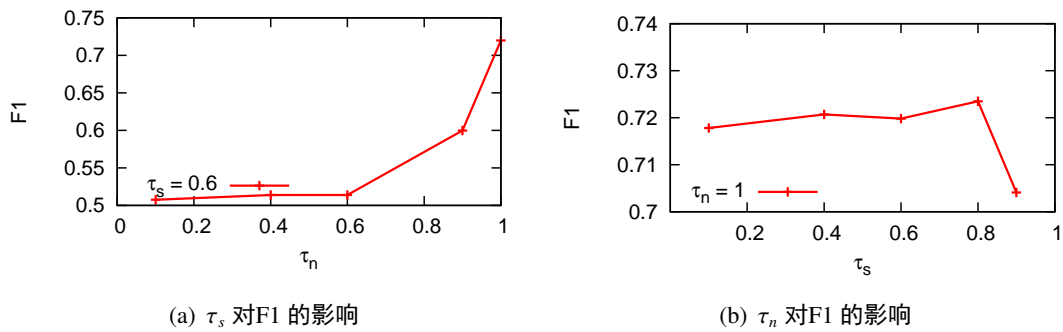


图 3-8 PRTree 的阈值对NokeaEM 的影响

### 3.8.2 与以往方法在匹配效果方面的比较

我们将本章提出的基于结构化非主属性数据的实体匹配算法（baseline 算法和PRTree (Nokey)-based 算法）与现有的使用多种技术合成的最先进的Key-based 匹配方法和最先进的使用结构化非主属性进行实体匹配方法Matching tree-based<sup>[2]</sup>进行比较，通过匹配的结果比较其各自的准确率和召回率。本章提出的算法中使用的字符串相似度函数统一使用Edit distance。

(1) *Key-based*: 该方法集成了多篇文章中所提到的基于键值匹配的最新技术。在本章中我们首先创建基于Q-gram 的后缀数组, 然后将那些可能匹配的实体进行分块处理, 具体来说就是根据已知的可信匹配实体中多提取的“分块主键”把实体进行分块操作<sup>[2]</sup>。除此之外,我们还融入了诸如前缀过滤(prefix-based filtering)<sup>[2]</sup>, 倒排索引(Inverted Indices)<sup>[2]</sup>等多种优化过滤技术进行块内实体间的相似性计算, 通过这些方法提升了实体匹配的效率。

(2) *Matching tree-based*: 该方法是一种基于结构化非主属性的匹配算法, 目前唯有Dey 等人在[?]中提出的另外一种基于0-1分支决策树的匹配树算法Matching tree-based。简而言之, 该方法在选择Matching tree 的节点时根据非主属性的匹配概率选择具有最大信息熵的非主属性。该方法只是简单地根据属性值是否一致构建0-1分支从而建立Matching tree, 但没有考虑缺失值的影响以及不同层次的节点应给予不同的权重。

在本节中我们除了对上述方法进行了比较, 同时也考虑了与基于PRTree 的扩展方法进行比较, 该方法将基于PRTree 的方法和基于Key-based 的方法结合, 通过三个数据集上的实验结果可以发现将主属性也用于创建PRTree 时匹配效果达到最优, 即让主属性融合到建树的过程中。因此, 我们也使用了PRTree(Key)-based 参与实验结果的比较。

首先, 我们评估上述5种匹配方法的准确率和召回率, 如图 3-9 所示, 通过实验结果可以发现上述5种匹配方法随着匹配召回率地不断增大, 匹配准确率整体上呈下降的趋势。但是不同的方法下降的平稳程度各不相同。通过比较发现Key-based 方法与本章中提出的方法相比, 在匹配准确率和匹配召回率方面效果最差。此外还可发现baseline 算法的效果比Key-based方法好, 而低于Matching tree-based 方法, 但是Matching tree-based 方法与PRTree 方法相比匹配效果略差, 注意到使用主属性的PRTree 方法在准确率和召回率方面获得了最优的效果。没有使用主属性的PRTree 匹配方法也达到一个较好的匹配效果。之所以没有使用主属性的PRTree匹配方法方法比使用主属性的PRTree 方法效果略差是因为该方法缺少了主属性的提供的信息。

现在我们对上述各种匹配方法所达到的匹配效果做出如下的分析: Key-based 方法整体上波动较大, 因为该方法易受数据表达方式的影响, 注意到当数据在表达方

式上相差较大时, 该方法的匹配准确率和匹配召回率会受到极大的影响, 导致那些本应该匹配的实体无法匹配到一起。如图 3-9 (b) 所示, 我们发现当Key-based 方法的召回率超过0.75后, 该方法的匹配准确率急剧下降。baseline 算法在稳定性上较key-based 方法好, 因为baseline 算法使用了结构化非主属性, 从而受数据表达方式多样的影响较小。如图 3-10 所示, Matching tree-based 方法和基于PRTree 的方法的匹配准确率随匹配召回率稳定变化, 但前者的变化率与使用主属性的PRTree 方法想比更大, 原因在于: 一方面是缺失值的影响, 导致Matching tree-based 方法的匹配效果受到了影响, 该方法并未考虑缺失值的情况, 而使用主属性的PRTree 方法在建树的过程中考虑了缺失值的影响; 另一方面, 前者没有考虑匹配树中不同层次的节点对匹配结果的影响, 而使用主属性的PRTree 方法则考虑了上述影响。使用主属性的PRTree 方法的匹配效果之所以比不使用属性的PRTree 方法好, 是因为使用主属性的PRTree 方法通过通过主属性获取了匹配是所需的一些信息。从整体上看, 通过在三个数据集上的实验结果可以发现本章提出的方法在匹配准确率方面提升了近15%, 而在匹配召回率方面提升了20%左右。

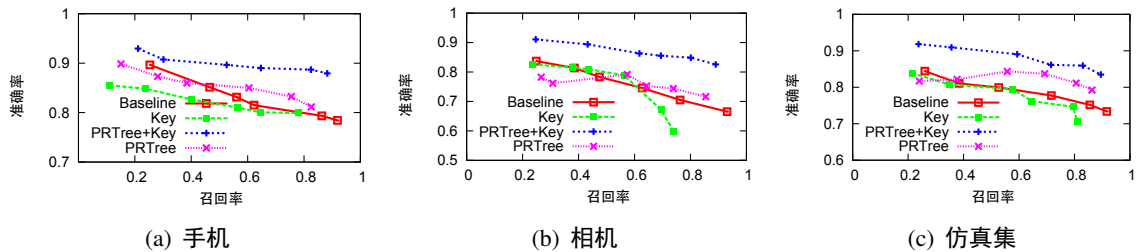


图 3-9 在准确率和召回率方面与以往方法的对比

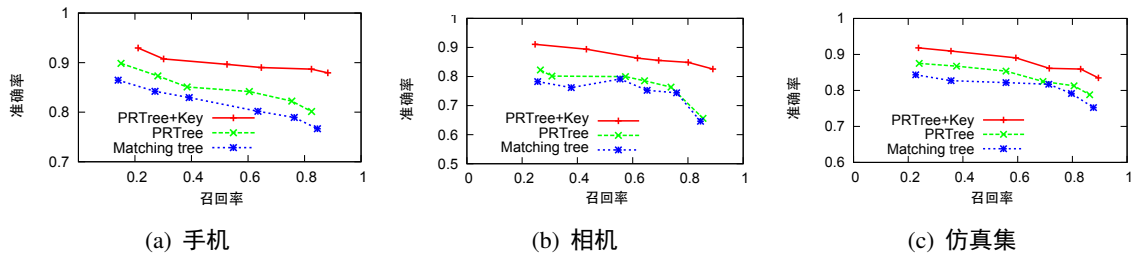


图 3-10 在准确率和召回率方面与Matching tree-based方法的对比

### 3.8.3 与以往方法在匹配效率方面的比较

在本节中我们对上述5种匹配方法在匹配效率方面进行了评估, 评估方式是比较

了方法在三个数据集上的匹配时间开销。如图 3-11所示，Key-based 方法具有最少的匹配时间开销，因为该方法只使用了主属性进行实体匹配，而且使用了block、prefix filtering 和inverted indices 等技术用于减少匹配的时间开销；通过实验发现baseline 算法的平均时间开销大约是Key-based方法的30倍，因为baseline 方法使用了所有的非主属性进行匹配，而基于PRTree 的匹配算法的时间开销相对于baseline 算法少10倍多，因为基于PRTree 的方法选择了具有最大充分性或必要性概率的非主属性，并结合停止建树阈值排除一些影响因子小的非主属性，通过这种方式能够尽早做出匹配决策，极大地减少属性值的比较次数，进而提高实体匹配效率。此外，从图 3-12中我们可以得到基于PRTree 的方法在匹配效率上与Matching tree-based 方法不相上下，但在平均效率上看基于Matching tree-based 的匹配方法稍低一些，因为Matching tree-based 方法在创建树时用于选择属性的开销较大，尤其是匹配过程中出现假阳性错误的时候。

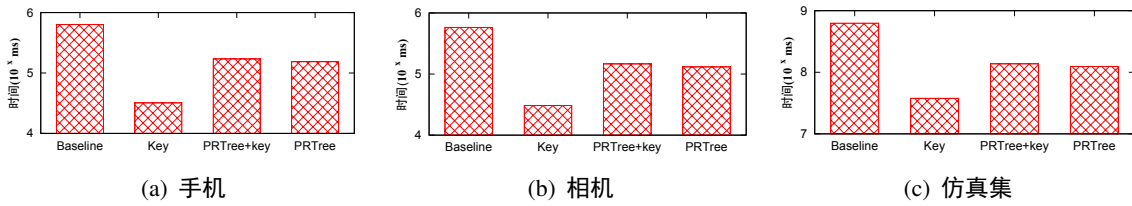


图 3-11 在效率方面与以往方法的对比

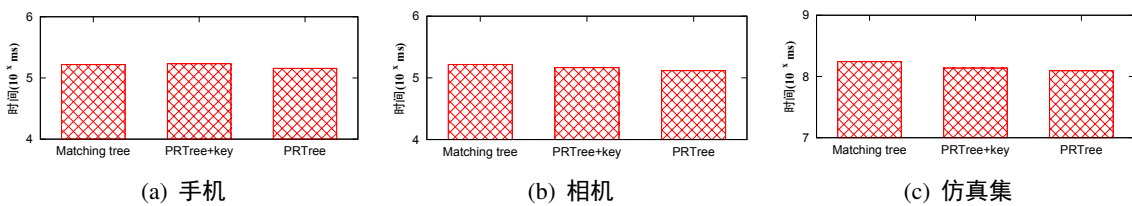


图 3-12 在效率方面与Matching tree-based方法的对比

### 3.8.4 算法的扩展性评估

如图 3-13所示，为了对PRTree 方法的可扩展性进行评估，我们在仿真数据集上进行了实验。如图 3-13（a）所示，我们可以看出随着记录的数目从100变化到10W，该方法的时间开销以近似于指数增长方式的形式缓慢增加，该结果也从另一方面证明了PRTree 可以极大地减少比较时间。从图 3-13（b）中可以看出，随着仿真集中属

性的数目从10增加至60，该方法的时间开销以近似于线性增长的方式增加。

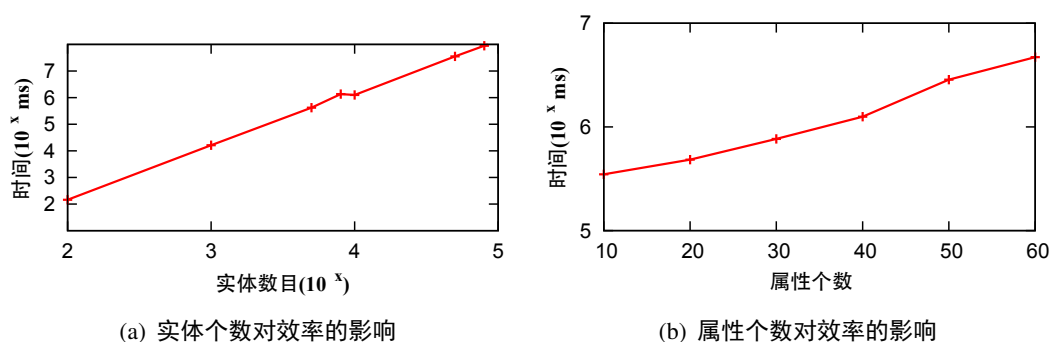


图 3-13 不同数目的记录数和属性数对PRTree在效率上的影响

### 3.9 本章小结

本章介绍了基于结构化非主属性的实体匹配算法的思路和问题定义，提出了两种匹配算法，分别给出了每种算法是如何进行实体匹配的。特别介绍了基于PRTree的NokeaEM算法的过程，包括如何创建PRTree以及利用PRTree进行实体匹配，并给出了算法的复杂度分析。最后，通过实验与其他已有的经典算法进行比较验证了提出算法的在准确性和效率方面较其它算法更为优秀。

## 第四章 基于文本类型非主属性数据的实体匹配

本章首先给出了基于文本类型非主属性数据实体匹配的问题定义。然后介绍了一些常见的文本处理方法，并分析了这些方法的优缺点。接下来提出了用于提升实体匹配效率的数据分块算法。最后详细介绍了本章提出的两个基于文本类型非主属性数据的实体匹配算法，并通过实验验证算法的效果。

### 4.1 问题定义

给定的一个关系表，实体匹配是指找出表中指代同一实体的数据库记录。本章中所使用的关系表既包含一些结构化数据（存在些缺失值），又包含文本类型的非结构化数据。在本章中对于包含实体关键信息的自由长文本描述数据，我们称之为Consolidated Textual Data（简称CText）。为了便于表述，在本章中我们称这种使用CText的实体匹配任务为CTextEM。CTextEM问题的形式化定义如下所示：

#### 定义5.

基于CText的实体匹配(CTextEM)：给定数据表 $T = \{r_1, r_2, \dots, r_n\}$ ，具有如下的模式 $S = \{[A_1, A_2, \dots, A_m], A_U\}$ ，其中 $m, n$ 是正整数， $r_i$ 表示一个实体( $1 \leq i \leq n$ )， $A_j$ 表示具有结构化数据的属性( $1 \leq j \leq m$ )， $A_U$ 表示带有CText数据的属性。CTextEM问题旨在找到一个函数 $\mathcal{F}(r_i, r_j, S)$ 和一个相似度阈值 $\theta$ ，使得 $\forall r_i, r_j \in T (1 \leq i, j \leq n, i \neq j), (r_i, r_j)$ 为指向同一实体的数据库记录，当且仅当 $\mathcal{F}(r_i, r_j, S) \geq \theta$ ，否则二者不是匹配的实体。

表 4-1给出了租房信息中的部分结构化非主属性，如“Residence Community”、“Location (District)”等，同时给出了文本类型的非主属性，如“General Supplemental Description”，其中包含了房源的某些特征信息，如朝向、绿化情况、装修类型等。从表 4-1中可以看出实体 $r_1$ 和 $r_4$ 尽管具有相同的结构化非主属性值，如“小区”、“地点”、“房源类型”等，但二者却不是同一房源，因为实体 $r_1$ 在属性“General Supplemental Description”中描述的房源朝向为“south”，而实体 $r_4$ 的朝向却是“east”，由此可判断二者并不表示同一房源，而是同一小区中具有相同楼层、装修类型的不同幢的房源。

在本章中我们同时使用结构化数据和CText进行实体匹配，图 4-1给出



表 4-1 带有CTexts 数据的租房信息表，其中实体 $r_1$  和 $r_2$  指的是同一房源，而实体 $r_3$  和 $r_5$  指的是同一房源

	Residence Community	Location (District)	Type	Size	Floor	General Supplemental Description
$r_1$	Eastern District Court	Canglang-Xujiang	Residence	75 $m^2$	3/15	1. Community Planning, unique warmth, flowers and trees patchwork, like a garden, world without dispute, furniture and appliances equipped well. 2. refined decoration, gentle color, facing south, ...
$r_2$	Eastern District Court	Canglang-Xujiang	Residence	75 $m^2$	3/15	1. Community Planning, unique warmth, flowers and trees patchwork, furniture and appliances equipped well. 2. fine decoration, mild color, facing south, ...
$r_3$	Oak Bay Garden	Xiangcheng-Yuanhe	Apartment	100 $m^2$	25/29	1. general decoration, south facing, nice view, good lighting, air conditioning and water heaters and closed kitchen equipped, 2. free of parking, free of property charges, ...
$r_4$	Eastern District Court	Canglang-Xujiang	Residence	75 $m^2$	3/15	1. Community Planning, flowers and trees patchwork. 2. good decoration, furniture and appliances equipped well, color matching gentle, facing east, ...
$r_5$	Oak Bay Garden	Xiangcheng-Yuanhe	Apartment	100 $m^2$	25/29	1. naïve decoration, south facing, good lighting, air conditioning and water heaters and washing machines proved, free of property charges, ...

了CTextEM 的工作流程图，对于给定的带有CText 数据的关系表，在数据预处理阶段，如果结构化数据中存在一些丢失的数据，则从CText 中抽取一些有用的信息（如“Area”，“District”，“Type”等）进行数据的填充，但仍有一些丢失数据无法填补。接下来，对结构化数据进行数据分块处理，获得带有缺失值的数据块和不含缺失值的数据块，然后构建两类比较实体对（块内实体对和块间实体对）；对于CText 数据，设计了CText Understanding 模块对CText 数据进行解析，提出了两种算法（IDF-based 提取算法和Sub-topics based 提取算法）提取CText 中的关键信息，然后利用提取的关键信息构建实体的比较向量，利用构建的比较向量计算实体的相似度，通过与预定义的相似度阈值的比较决定是否匹配。

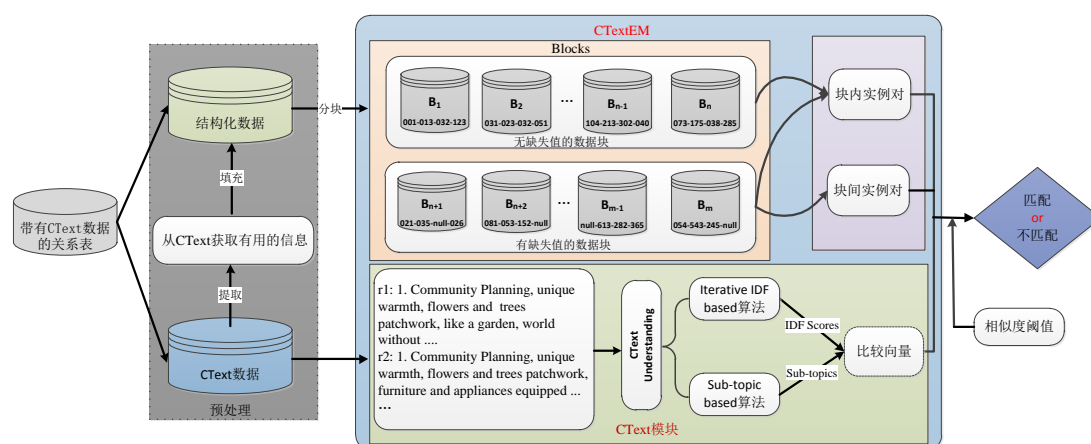


图 4-1 CTextEM 算法的工作流

## 4.2 传统的文本数据处理方法

作为结构化数据的补充，非结构化的文本数据（CText）经常出现在各种类型的数据集中，比如一些二手商品数据（二手车，二手房和二手家具等）。然而，使用CText并非易事，传统的字符串相似度度量方法如编辑距离（Edit distance）、词袋模型（bag-of-words）等并不适合度量CText之间的相似度，因为每个CText中包含了十几个句子甚至更多，而且还含有许多的噪数据，导致从CText中获取重要信息更加艰难。

近年来，有一些文本数据的处理方法被提出。文献[?]提出了一个信息抽取模型，借助WordNet从CText中抽取诸如“Address”，“City”，“Phone”，“Type”等固定类型的信息，但该方法存在适用性问题，当换成其它数据集时可能无法抽取有用的信息，而且对WordNet存在依赖性。文献[?]提出了一种基于语义特征的方法，该方法定义了一个形如{time, location, agentive, objective, activity}语义特征向量用于提取文本数据中的信息，但是这种方法受限于特征向量的维度，因为有些文本中可能并不包含这些特征向量。因此，此方法很难应用于其它数据集。

此外，文本处理和文本理解（Text Understanding）具有一定的相似性，都是从文本数据中获取一些关键信息，但文本理解侧重于如何理解文本数据中的主要信息，而不是考虑文本数据中短语之间的关联关系。文献[?]提出了一种利用深度学习算法（时间卷积网络）从字符级输入到抽象文本概念进行文本理解的方法。一些经典的主

题模型如LDA<sup>[2]</sup>, LSA<sup>[2]</sup>, PLSA<sup>[2]</sup>等可以从自由文本中识别隐藏的主题词(如“教育”、“经济”、“体育”和“文化”等),但是实体的CText数据中并不存在一些明显的主题可以用于获取CText的关键信息,也即是说CText并不能很好的被一些主题所表示,因为CText中的自由文本通常是对实体某一主题的多个方面进行描述,这些方面共享相同的主题,它们之间并没有明显的界限。如果仍使用上述模型可能无法提取有效的信息。另一方面,CText的子主题很短,比如几个单词,因此应用之前的方法很难学到一些有用的子主题。

近来有一些子主题挖掘方法被提出,文献[?]根据单词的依赖关系提出了基于单词共现的方法,借助网页中的锚节点文本挖掘隐藏的子主题,但这种方法受限于查询query的质量,而且还需要外部资源的支撑,这都影响了该方法的健壮性。文献[?]结合了LDA模型和短语共现的原理去发现文本中的主题,但此方法需要外部专家的干预去学习文本的主题分布,然而通常情况下这些外部专家并不容易找到或者开销太大。

### 4.3 数据分块算法

实体匹配的效率问题也是一个很热的研究课题,传统的匹配方法是实体间的一对一比较,也即是说假定一个数据源具有 $m$ 个实体,另一个数据源有 $n$ 个实体,那么匹配时的比较次数将是 $m \times n$ 。当 $m, n$ 很小时,匹配的时间是可以接受的,但当 $m, n$ 较大时,匹配的时间开销就会大大增加,使得匹配结果的获取异常艰难。为了减少实体之间的比较次数,避免一些不必要的比较,提高实体匹配的效率,提出了一个数据分块算法,利用结构化数据对实体进行分块处理。该算法假定具有相同的结构化非主属性值的实体有可能是匹配的,但并不排除不匹配的可能,通过这种方式将可能匹配的实体放到同一个数据块中。具体而言包含如下两个步骤:

(1) 实体分块: 首先,根据每个结构化非主属性识别一实体不同于其它实体的能力对非主属性进行考量,其评判标准如下,实体在同一属性下具有相同的值,而且实体之间是匹配的。数据集中满足此规则的实体数目越多,该属性的识别能力就越强。然后,利用将具有多个相同属性值的实体分配到同一个块中,并给每个数据块分配一个标识码。值得注意的是结构化非主属性中经常有一些缺失值,对数据分

块造成了很大的影响, 因为丢失的数据并不能确定是什么内容, 即使未丢失的属性值都一样, 也不能保证这些实体应该分到同一个数据块中。为此, 我们设定了另外一种数据块, 即包含缺失值的数据块, 该块内的实体除了缺失值外其他属性值都相同。

(2) 构建比较实体对: 对于同一数据块内的每个实体, 产生两两比较的实体对, 此时匹配的比较次数将远远小于  $m \times n$ , 而与数据块的大小和每个块内的实体数相关, 从而减少了大量不必要的比较。同时, 如果两个数据块除了缺失值外其他非主属性值都相同, 仍具有相同的标识码, 那么这两个数据块的实体也会进行两两比较。通过上述处理方法可以在不太影响匹配准确性的前提下, 尽可能的减少的实体匹配次数, 进而提升匹配效率。

#### 4.4 基于IDF 的迭代式实体匹配算法

信息检索中逆文档频率 (Inverted Document Frequency, 简称IDF) 可以度量短语的普遍重要性, 短语的IDF 值越大, 说明该短语越重要。为了从CText 中获取有用的信息, 我们提出了一个基于短语IDF 得分的实体匹配算法 (baseline 算法), 该算法假定具有高IDF 得分的短语能够近似的表达CText 数据的主要信息。因此, 可以通过余弦相似度函数 (Cosine Similarity Function) 表示计算CText 之间的相似性的相似度函数, 其输入为CText 中的重要短语集合。

##### 4.4.1 Baseline算法的工作流程

对于给定一个实体的CText 数据, 在移除了停顿词的前提下, 提取2-6个长度的单词作为候选短语, 接下来计算这些短语的IDF 得分, 选择得分较高的短语作为CText 的关键信息, 利用这些短语构建实体的比较向量, 然后计算CText 之间的相似度, 与预定义的相似度阈值比较, 如果相似度大于阈值则认为实体是匹配的, 否则不匹配。具体细节如下所述:

(1) 构建比较向量: 首先, 根据CText 所在的块计算每个短语的IDF 得分。然后, 根据IDF 得分对这些短语以降序的方式进行排序, 选择具有较高IDF 得分的短语表示其所对应的实体。最后, 将数据集中每个实体所获得的不同短语进行收集整理, 建立全局短语集合  $\vec{P}_g = \{w_1, w_2, \dots, w_g\}$ , 对于每个实体  $r_i (1 \leq i \leq n)$  根据  $\vec{P}_g$  构建布

尔类型的比较向量  $\vec{v}_i = \{bool(r_i, w_1), bool(r_i, w_2), \dots, bool(r_i, w_n)\}$ , 其中:

$$bool(r_i, w_j) = \begin{cases} 1, & \text{if } w_j \text{ exists in the phrase set of } r_i \\ 0, & \text{otherwise} \end{cases} \quad (4.1)$$

(2) 计算实体相似度: 对于实体  $r_i$  和  $r_j$  的比较向量  $\vec{v}_i$  和  $\vec{v}_j$  ( $1 \leq i, j \leq n$ ), 通过如下的方法计算两个实体的相似度:

$$sim(r_i, r_j) = \frac{\vec{v}_i \times \vec{v}_j}{\|\vec{v}_i\| \cdot \|\vec{v}_j\|} = \frac{\sum_{p=1}^g bool(r_i, w_p) bool(r_j, w_p)}{\sqrt{\sum_{p=1}^g bool(r_i, w_p)^2} \cdot \sqrt{\sum_{q=1}^g bool(r_j, w_q)^2}} \quad (4.2)$$

(3) 调整数据块: 假定  $\theta$  表示预定义的相似度阈值。如果  $sim(r_i, r_j) > \theta$ , 并且实体对  $(r_i, r_j)$  之前在同一个数据块中, 它们将被合并为一个实体; 否则, 如果  $sim(r_i, r_j) > \theta$ , 但是实体对  $(r_i, r_j)$  之前隶属于不同的数据块, 在  $r_j$  的数据块编码存在缺失值的前提下, 将实体  $r_j$  从原来的数据块中移除放到实体  $r_i$  所在的数据块中, 并将之与  $r_i$  合并。

#### 4.4.2 短语IDF得分的迭代式更新

值得注意的是我们计算每个短语的IDF得分时, 是根据该短语的CText数据所在的数据块计算得到的, 随着匹配实体的发现, 实体所依赖的数据块也应随之该改变。因此, 我们需要对短语的IDF得分更新。迭代式更新短语IDF得分的想法是基于如下事实: (1) 随着更多匹配实体的发现, 更多相关的CText可以被用于计算或更新短语的IDF得分; (2) 随着相同数据块中相关CText数据的增加, 更多匹配的实体可以被发现。因此, 我们迭代地更新所有短语的IDF得分, 然后重复上述三个步骤直至短语的IDF得分趋于稳定。

### 4.5 基于短语共现的实体匹配算法

基于IDF得分的baseline算法只是在一个维度上度量CText之间的相似度。然而, 作为包含实体各方面信息的CText数据, 实际上可以通过方法分析出短语之间的关联关系, 利用关联关系提取CText的主要信息。在本章中将描述实体某些方面特性的关键信息称之为子主题。这里所说的子主题不同于诸如“体育”、“经济”、“文化”等主题, 而是同一主题的各个方面。例如, 在房源数据表中的CText数据通过诸如

“direction”，“greening”，“property”，“traffic”等方面进行描述。

鉴于baseline 算法的不足，我们提出了一种新颖的基于短语共现的实体匹配算法，用于挖掘CText 的子主题，然后计算CText 数据在所有子主题维度上的相似性。该算法的思路是如果两个短语总是同时出现在一个句子中，那么很有可能这两个短语之间存在一定的关联性，通过对这种关联性的挖掘找出CText 中的关键信息，然后通过归一化处理获取关键信息的向量表示，最后在结构化非主属性的帮助下计算CText 对应的向量之间的相似性表示实体间的相似度。基于上述事实，通过对CText 中短语的统计分析方法建立短语共现图（Phrase Cooccurrence Graph，简称PC-Graph）用以表示CText 数据之间所有短语的共现关系。值得注意的是有些短语虽然与其它短语共现但是这种共现关系并不强烈，又或是某些短语经常与其它多个短语共现，为了解决这个问题我们提出了分割算法将PC-Graph 分割为多个划分。然后利用短语关联度（Phrase Association Degree，简称PAD）度量实体在相应子主题上的相似度。

在执行图划分时面临一个挑战就是如何使得每一个划分中的短语都尽可能地对应到一个子主题上。为此，我们首先将此问题构建成一个优化问题，然后通过理论上分析证明其是一个NP-hard 问题。通过短语的PAD 度量短语之间的关联程度的大小，并提出了一个贪心算法用于选择具有最小PAD 得分的边作为划分子点。

#### 4.5.1 共现图的构建

对于给定的一个CText 数据 $ct$ ，根据诸如“，”，“.”，“?”等分隔符和停顿词将CText 分割为多个片段 $t_1, t_2, \dots, t_n$ ，然后在给定的已经过滤过停顿词的词典上，利用Longest-Cover 方法<sup>[2]</sup>对每个片段分割获得最长短语。如果两个短语同时出现在一个片段中就在两个短语之间添加一条边并给其一定的权重，其中短语 $p_i$ 和 $p_j$ 所在边的权重可根据公式 4.3得到：

$$freq(ct, p_i, p_j) = e^{-gap_{ct}(p_i, p_j)} \cdot bool(p_i, p_j) \quad (4.3)$$

其中 $gap_{ct}(p_i, p_j)$ 表示短语 $p_i$ 和 $p_j$ 在其CText 数据中的距离， $e^{-gap_{ct}(p_i, p_j)}$ 是为了惩罚具有较长距离的短语， $bool(p_i, p_j)$ 是为了减少相似短语在同一CText 数据中的影响，具

体可由公式 4.4 计算得到:

$$bool(p_i, p_j) = \begin{cases} 1, & \text{if } sim(p_i, p_j) \leq \theta \\ 0, & \text{otherwise} \end{cases} \quad (4.4)$$

其中函数  $sim(\cdot, \cdot)$  用于计算短语之间的字符串相似度 (如编辑距离),  $\theta$  是字符串相似度阈值。

对于给定的训练集合  $T$ , 使用公式 4.5 统计短语对  $(p_i, p_j)$  在所有 CText 数据中共同出现的总频率,

$$Freq(p_i, p_j) = \sum_{ct \in T} freq(ct, p_i, p_j) \quad (4.5)$$

进而, 使用公式 4.3 结合  $Freq(p_i, p_j)$  计算连接顶点  $p_i$  和  $p_j$  所在边的 PAD 得分,

$$PAD(p_i, p_j) = \frac{Freq(p_i, p_j)}{\sum_{p \in \vec{P}_g} Freq(p_i, p)} \cdot \log \frac{|\vec{P}_g|}{|Adj(p_j)| - 1} \quad (4.6)$$

其中  $\frac{Freq(p_i, p_j)}{\sum_{p \in \vec{P}_g} Freq(p_i, p)}$  计算  $p_i$  和  $p_j$  的 PAD 得分与所有连接  $p_i$  的短语的 PAD 得分之和的比例。  
 $\log \frac{|\vec{P}_g|}{|Adj(p_j)| - 1}$  为了惩罚总是和其它短语一起出现的其它短语,  $Adj(p_j)$  表示和  $p_j$  一起出现的短语集合,  $|\cdot|$  获取集合的大小。

图 4-2 给出了部分建立在租房信息数据集的 PC-Graph, 从图中可以看出经常一起出现的短语具有较高的 PAD 得分, 如 “convenience”, “ease” 等, 而有些很少在一起出现的短语具有较低的 PAD 得分, 如 “good” 和 “traffic” 等。

#### 4.5.2 PC-Graph 的分割

如图 4-2 所示, 有些短语节点中存在很弱的关联关系, 也即是一些短语之间具有很低的 PAD 得分, 阻碍了我们从 PC-Graph 识别 CText 的子主题。因此, 本章考虑将 PC-Graph 分割为多个图划分, 其中的每一个划分紧密的对应着一个子主题。受文献 [?] 所提模型的启发, 我们的 PC-Graph 分割问题可转化为如下的优化问题: (1) 最大化每个图划分内的 PAD 得分之和; (2) 减少图划分之间的 PAD 得分。此问题可形

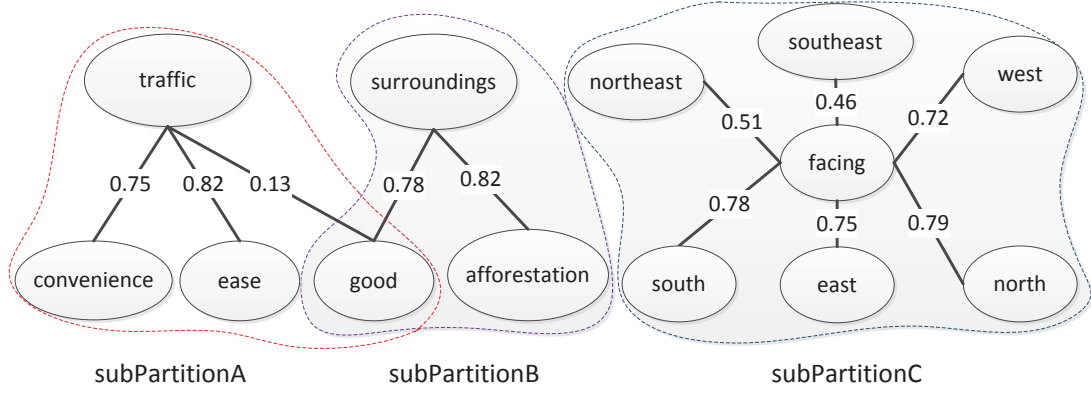


图 4-2 例：具有三个划分的PC-Graph

式化的表示为公式 4.7 所示的最大化问题：

$$\text{Maximize} \sum_{p_1 \in \vec{P}_g, p_2 \in \vec{P}_g, p_1 \neq p_2} \frac{PAD(p_1, p_2)}{dis(p_1) + dis(p_2) + \alpha} \quad (4.7)$$

$$\text{where} \begin{cases} dis(p_1) = \text{Max}_{p \in \text{Adj}(p_1)} PAD(p_1, p) - \text{Min}_{p \in \text{Adj}(p_1)} PAD(p_1, p) \\ dis(p_2) = \text{Max}_{p \in \text{Adj}(p_2)} PAD(p_2, p) - \text{Min}_{p \in \text{Adj}(p_2)} PAD(p_2, p) \end{cases} \quad (4.8)$$

其中 $\alpha$ 是一个平衡因子，用于防止分母为零。

### 定理1.

发现公式 4.7 的最优解决方法是一个 *NP-hard* 问题。

### 证明1.

我们证明即便在子主题数目已知的情况下，上述优化问题是 *NP-hard* 问题。通过证明该问题可还原为平衡 *max-skip* 划分问题，从而使原命题得证<sup>[2]</sup>。对于给定一个二元向量集合  $V$ ，通过  $V$  发现一个划分  $\mathcal{P}$  使得公式 4.9 的总开销  $C(\mathcal{P})$  最大化：

$$C(\mathcal{P}) = \sum_{p_i \in \mathcal{P}} C(P_i) \quad (4.9)$$

其中  $C(P_i) = |P_i|$  是一个图划分  $P_i$  的开销。在我们的问题中，使用公式 4.10 定义一个



图划分 $P_i$ 的开销:

$$C(P_i) = \sum_{p_1 \in \vec{P}_g, p_2 \in \vec{P}_g, p_1 \neq p_2} \frac{PAD(p_1, p_2)}{dis(p_1) + dis(p_2) + \alpha} = \sum_{p_1 \in \vec{P}_g, p_2 \in \vec{P}_g, p_1 \neq p_2} 1 - \Delta(P_i) \quad (4.10)$$

其中 $\Delta(P_i)$ 类似于在平衡 $max-skip$ 划分问题中的 $\bar{v}(P_i)$ 。因此,公式4.7等价于最大化总开销 $\mathcal{P}$ ,也即是说,发现公式4.7的最优解问题是 $NP-hard$ 问题。因此,定理1得证。

根据定理1可知解决这个非线性优化问题并不容易。为此,我们提出了一个贪心算法,每次贪心地选择具有最小PAD得分的边执行图划分。为了表示每个划分内节点的紧密程度,我们为每个划分 $G_{par}$ 定义了一个一致性得分(cohesion score,简称CScore),可由如下公式4.11计算得到:

$$CScore(G_{par}) = \frac{\sum_{(p_1, p_2) \in \vec{P}_{G_{par}}} PAD(p_1, p_2)}{\max_{(p_1, p_2) \in \vec{P}_{G_{par}}} PAD(p_1, p_2) - \min_{(p_1, p_2) \in \vec{P}_{G_{par}}} PAD(p_1, p_2) + \alpha} \quad (4.11)$$

其中 $\alpha$ 是一个平衡因子,用于防止分母为零, $\vec{P}_{G_{par}}$ 表示划分 $G_{par}$ 的短语集合。假定图划分 $G_{par}$ 可以在具有最小PAD得分的边上被分割为两个子划分 $G_{par1}$ 和 $G_{par2}$ 。如果这个分割操作满足如下公式4.12中的限制条件,我们将执行分割操作。

$$\left\{ \begin{array}{l} CScore(G_{par}) \leq CScore(G_{par1}) + CScore(G_{par2}) \\ |CScore(G_{par1}) - CScore(G_{par2})| \leq \min_{(p_1, p_2) \in \vec{P}_{G_{par}}} PAD(p_1, p_2) \\ |G_{par1}| > 1, |G_{par2}| > 1 \end{array} \right. \quad (4.12)$$

对于每一个图划分,我们迭代的选择具有最小PAD得分的边执行分割操作,直至没有更多的边满足公式4.12中的限制条件。

#### 4.5.3 子主题及其权重的获取

对于分割后的每个图划分,我们从中获取图划分的子主题。具体来说,计算图划分中每个节点的平均PAD得分,选择具有最大平均PAD得分的短语作为该图划分的子主题,而其它短语作为该子主题下的子主题值。假定从PC-Graph中我们得到了 $K$ 个子主题可形式化表示为向量 $\langle subT_1, subT_2, \dots, subT_K \rangle$ ,其中每个 $subT_i$

$(1 \leq i \leq K)$  表示一个子主题。对于子主题向量的每一维使用领域知识设置用于实体匹配的权重，可形式化表示为如下的权重向量  $\langle w_1, w_2, \dots, w_K \rangle$ 。初始阶段，我们设定  $w_k = 1 (1 \leq k \leq K)$ ，但随着匹配结果的改变权重向量将被迭代式的更新。在一次迭代完成后根据实体匹配结果，使用公式 4.13 更新权重  $w_k$ 。我们迭代式的执行更新操作直至它趋于稳定。

$$w_k = \frac{Pos_{subT}(k)}{Pos_{subT}(k) + Neg_{subT}(k)} \quad (4.13)$$

其中  $Pos_{subT}(k)$  表示满足如下要求的所有实体对  $(r_i, r_j)$  的个数：如果  $r_i[k] = r_j[k]$ ，也即是说实例对  $(r_i, r_j)$  在子主题  $subT_k$  上具有相同的子主题值，那么实体对  $(r_i, r_j)$  在当前迭代中是匹配的实体； $Neg_{subT}(k)$  表示满足如下要求的所有实体对  $(r_i, r_j)$  的个数：如果  $r_i[k] \neq r_j[k]$ ，那么实体对  $(r_i, r_j)$  在当前迭代中不是匹配的实体。

#### 4.5.4 实体匹配过程

首先，我们识别每个实体的每个 CText 片段的子主题，然后使用 Adjusted Cosine Similarity 函数<sup>[2]</sup>计算实体对的相似度：

$$Sim(r_i, r_j) = \frac{\sum_{k=1}^K w_k^2 \cdot sim(r_i[k], r_j[k])}{\sum_{k=1}^K [w_k \cdot sim(r_i[k], r_j[k])]^2} \quad (4.14)$$

上述方法虽然可以很好的发现 CText 片段中的子主题，但是有可能在 CText 片段中没有明显的子主题，这将直接导致片段子主题识别的失败。为此，我们使用了一个概率模型去推断其所隶属的子主题。假定  $\vec{P}(t)$  表示片段  $t$  中识别的短语集合，我们根据全概率公式使用公式 4.13 计算  $t$  属于子主题  $subT$  的概率：

$$Pr(subT|\vec{P}(t)) = \sum_{p \in \vec{P}(t)} \frac{Pr(p|subT) \cdot Pr(subT)}{\sum_{subT} Pr(p|subT) \cdot Pr(subT)} \quad (4.15)$$

其中  $subT$  是一个子主题， $Pr(p|subT)$  表示在满足子主题  $subT$  的条件下出现短语  $p$  的概率，其值可通过先验知识计算可得。在计算了  $t$  属于每个不同子主题的概率之后，选择具有最大概率值的子主题作为该片段  $t$  的子主题，然后利用公式 4.14 计算实体的相似度。

## 4.6 实验结果及分析

我们从Web上爬取了二手商品数据，其中既包含结构化数据又包含文本数据，在真实数据集上验证所提出算法的有效性和可用性。

(1) 二手房数据集 (*House*): 这个数据集是关于租房信息的，我们收集了来自三个租房网站 (“Ganji”, “Anjuke”, “58tongcheng”) 的五个大中型城市 (“Beijing”, “Shenzhen”, “Tianjing”, “Chendu” 和 “Suzhou”) 的二手房数据，其中既有结构化的数据也有非结构化的CText数据。数据集的性质如表4-2所示。

(2) 二手车数据集 (*Car*): 这个数据集是关于二手车信息的，我们搜集了来自于 “Ganji” 和 “168che” 的五个品牌 (“Toyota”, “Audi”, “BMW”, “Honda” 和 “Buick”) 的二手车数据，其中包含了结构化数据和CText信息。数据集的性质如表4-3所示。

表 4-2 二手房数据信息表

	House				
	beijing	chengdu	suzhou	shenzhen	tianjin
#Attribute	22	22	22	22	22
#Record	5.6k	8.6k	10.8k	17.1k	13.5k

表 4-3 二手车数据信息表

	Car				
	Toyota	Audi	BMW	Honda	Buick
#Attribute	12	12	12	12	12
#Record	5.6k	5.2k	6.0k	5.5k	5.8k

为了评估方法的效果我们使用了3个度量标准：准确率 (Precision)，是指通过实验获得的所有匹配的实体中正确匹配的实体所占的比例；召回率 (Recall)，是指所有应该匹配的实体中正确匹配的实体所占的比例；F1 Score，是指对准确率和召回率的综合考虑，计算方法为： $F1 = (2 * precision * recall) / (precision + recall)$ 。我们使用算法的时间开销对方法的效率进行评估。

### 4.6.1 与以往方法在匹配效果方面的比较

我们将本章中提出的两种CTextEM算法，基于IDF的实体匹配算法（称为baseline算法）和基于短语共现的实体匹配算法（称为Sub-topic算法），与一

些最先进的EM（实体匹配）算法和使用了其他经典主题模型的CTextEM 算法进行比较，并对它们的效果进行评估。

（1）*Key-based*：该方法集成了一些基于主属性值的先进技术用于减少匹配的开销，如Q-gram<sup>[2]</sup>和Inverted Indices<sup>[2]</sup>。

（2）*Blocking-based EM*<sup>[2]</sup>：该算法选择具有好识别度的属性创建Hash Buckets 进行实体匹配。在相同的Hash Buckets 中的实体有可能是同一实体，而具有不同Hash 编码的实体则不可能是匹配的实体。

（3）*PRTree-based EM*<sup>[2]</sup>：该算法使用具有结构化数据的属性（主属性和非主属性）建立一颗类决策树，利用该类决策树进行实体匹配。

（4）*LDA-based EM*：该算法依赖LDA主题模型<sup>[2]</sup>从CText 数据中挖掘隐藏的各种主题，进而建立主题向量，利用主题向量计算实体间的相似度。

（5）*GLC-based EM*：该算法依赖GLC 主题模型<sup>[2]</sup>理解CText 数据中的信息，然后建立主题向量，利用主题向量计算实体间的相似度。

如图 4-3（a）所示，只依赖主属性的Key-based EM 方法具有最低的F1 score，由于结构化数据中具有丢失数据使得Blocking-Based EM 方法的效果大打折扣，导致了匹配假阳性的出现。PRTree EM 方法于Key-based EM 方法相比具有较高的F1 score，而与baseline 算法相比其结果低于其F1 Score，因为PRTree EM 方法只使用了结构化数据而没有使用CText 数据，导致一些本应该匹配的实体却没有被发现。baseline 算法从CText 数据中提取信息，并通过与结构化数据的结合，获得了较高的F1 score。LDA-based EM 算法的准确性低于baseline 算法，因为它并不擅长从CText 数据中学习子主题。baseline 算法和GLC-based EM 方法的匹配准确性不相上下，但都低于Sub-topic 算法，因为它可以以更高级的方法从CText 中抽取关键信息，进而获得更好的匹配效果。我们同时给出了这些方法在二手车数据集上的实验结果，如图 4-3（b）所示。

为了获取更为全面的比较效果，我们比较了这些方法在房源数据集上的准确率和召回率。由于页面原因只选取前四个城市的实验结果进行展示。由表 4-4可以看出，sub-topic EM 算法较其他的匹配算法而言也取得了最高的准确率和召回率，而GLC-based EM 方法获得了第二高的匹配准确率和召回率。baseline 算法的结果近

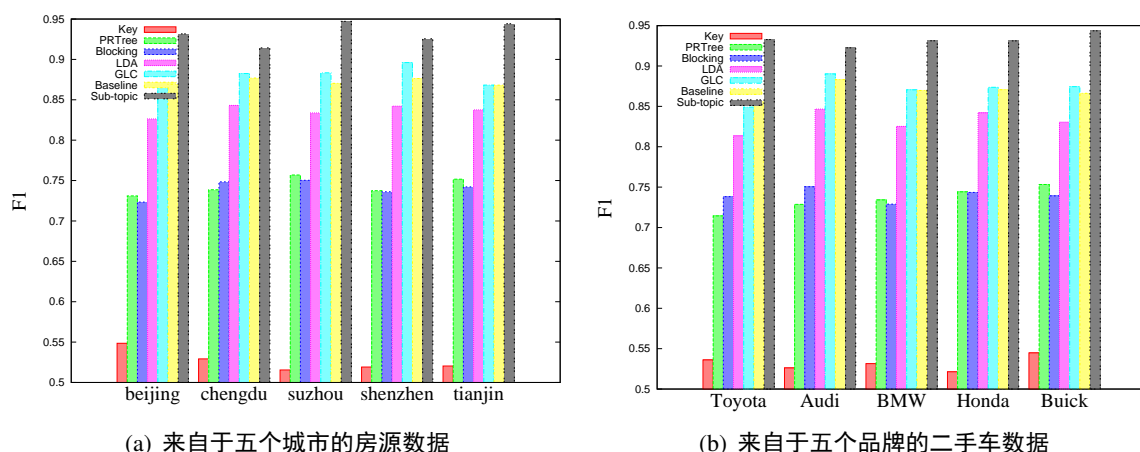


图 4-3 与以往方法在F1 score上的比较

似于GLC-based EM 方法，但LDA-based EM 方法是这四个使用CText 数据中最差的。

表 4-4 在房源数据集中与以往方法在准确率和召回率上的比较（前四个城市）

Methods	beijing		chengdu		suzhou		shenzhen	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
<b>Key</b>	0.6994	0.4512	0.7116	0.4212	0.7254	0.3998	0.7059	0.4105
<b>PRTree</b>	0.7504	0.7125	0.7542	0.7239	0.7556	0.7582	0.7694	0.7081
<b>Blocking</b>	0.7452	0.7028	0.7645	0.7332	0.7583	0.7425	0.7467	0.7259
<b>LDA</b>	0.8472	0.8066	0.8616	0.8253	0.8438	0.8241	0.8527	0.8320
<b>GLC</b>	0.8801	0.8625	0.8964	0.8693	0.9045	0.8632	0.9366	0.8590
<b>baseline</b>	0.8966	0.8437	0.9059	0.8498	0.8891	0.8524	0.9105	0.8447
<b>Sub-topic</b>	<b>0.9688</b>	<b>0.8974</b>	<b>0.9472</b>	<b>0.8836</b>	<b>0.9802</b>	<b>0.9163</b>	<b>0.9650</b>	<b>0.8892</b>

#### 4.6.2 算法提取结果的评估

我们将本章提出的两种信息提取方法与不同的主题模型提取方法进行比较，如表 4-5所示，Sub-topic EM 算法较其他方法能够获得更为准确的信息，因为该方法通过产生的子主题很好的抽取了CText 数据中的有效信息。从表中可以看出LDA 模型只获取了粗糙的信息，这些信息不能很好地支撑实体匹配。此外，还可以发现一些重要的短语如“Community Planning”被分为了两个短语。GLC 模型有时能获取较好的子主题而有时却不能，例如，短语“floor”和“twenty”被混合在一起，这说明了该方法对数据集具有依赖性。baseline 算法的结果近似于GCL 模型的提取结果。总结来说，Sub-topic 方法较其它方法在理解CText 数据方面更为合适。

如表 4-6所示，我们根据子主题获取模型及权重分配模型将房源数据集中部分子主题及其权重列举出来。从该表可以观察到子主题“floor”较其它子主题具有较高

表 4-5 与以往的方法在抽取结果上的比较

Methods	Example	
	1. Community Planning well, unique warmth, flowers and trees patchwork, like a garden, furniture and appliances e-quipped well, refined decoration, facing south right, twenty floor, tenant types limit for family ...	1. south facing, good lighting, two air conditioning, water heaters and washing machines equipped, free of property charges. ...
LDA	<b>Community, Planning, warmth,</b> flowers, trees, garden, <b>furniture, appliances, decoration, south, floor,</b> tenant types, family, ...	<b>south, facing, lighting,</b> air, conditioning, water, heaters, washing, machines, <b>property, charges,</b> ...
GLC	<b>Community Planning, warmth,</b> flowers and trees, garden, <b>furniture and appliances,</b> refined, <b>decoration,</b> south, <b>twenty, floor,</b> tenant types, family, ...	<b>south, lighting, two,</b> air conditioning, water heaters, washing machines, <b>free, property charges,</b> ...
baseline	<b>Community Planning,</b> well, warmth, flowers and trees, garden, <b>furniture and appliances,</b> refined, <b>decoration, south, facing, floor,</b> tenant types, family, ...	<b>facing, south,</b> lighting, air conditioning, water heaters and washing machines, <b>property charges,</b> ...
Sub-topic	<b>Community Planning,</b> warmth, flowers, trees, furniture, <b>appliances and decoration,</b> well-groomed, <b>facing, floor,</b> tenant types, family, ...	<b>facing, lighting,</b> air conditioning, water heaters, washing machines, <b>property charges,</b> ...

的权重，因为该子主题能够更好的决定匹配的结果，如果两个房源实体具有不同的楼层号，则二者不可能为同一实体。这也和我们的期望一致，具有较高识别度的子主题较其它子主题具有更高的权重。

表 4-6 例：房源数据集中不同子主题的权重

CText	1. Community Planning well, unique warmth, flowers and trees patchwork, like a garden, furniture and appliances equipped well. 2. Hardcover house, well-groomed room very much, matching color, facing south right, twenty floor, free of property charges, tenant type limits for single, free of parking, bag check, ...								
Phrases	Community Planning	warmth	flowers and trees patchwork	furniture and appliances	decoration	color	facing	floor	...
Weight	0.56	0.31	0.43	0.75	0.69	0.44	0.85	0.89	...

4.6.3 算法的扩展性评估

我们比较了本章中提出的两个算法（baseline EM 算法和Sub-topic EM 算法）与基于主题模型的算法如LDA-based EM 方法和GLC-based EM 方法的可扩展性，即随着数据量的增加算法在F1 score 和时间开销方面的变化趋势。如图 4-4（a）所示，随

随着实体数目从100增加至10000，Sub-topic EM 算法的F1 score 变化非常稳定，而且较其它方法具有更高的F1 Score。如图 4-4 (b) 所示，随着实体数目的增加，Sub-topic EM 算法的时间开销较其它算法更少，原因在于该算法通过所产生的子主题能够更好的减少比较的维度，从而减少实体匹配的时间。

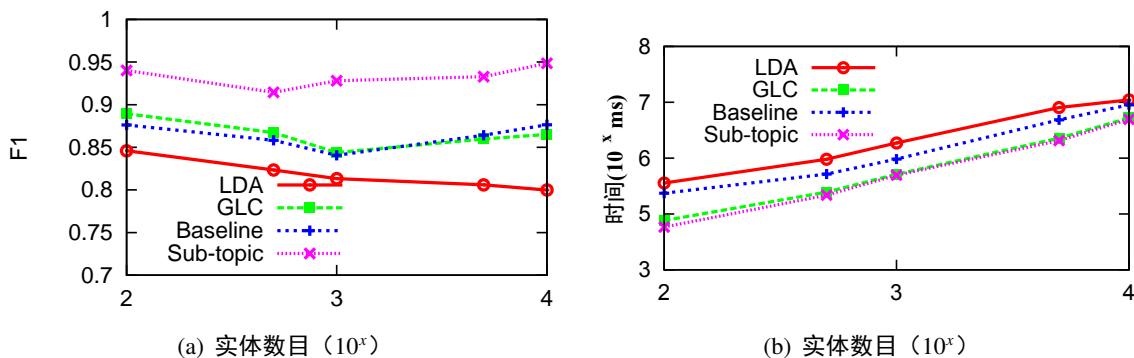


图 4-4 扩展性评估：在F1 score和时间开销方面与一些主题模型的比较

## 4.7 本章小结

本章主要介绍了如何利用非结构化的文本数据进行实体匹配。本章首先给出了基于文本类型非主属性实体匹配的问题定义，简单介绍了已有的文本处理方法并描述了其优缺点。然后提出了一种提高匹配效率的数据分块算法，并在此基础上提出了两种基于文本类型非主属性数据的实体匹配算法，最后在实验部分通过与以往实体匹配方法的比较，验证所提出算法的效果。

## 第五章 总结与展望

### 5.1 全文总结

随着信息技术的高速发展，各领域数据的量级呈现爆炸式增长，伴随而来的数据质量问题，如数据失真、数据过期、数据缺失、数据表达不一致等问题，也日益凸显。其中数据表达不一致问题尤为突出，该问题不仅影响了数据的高效使用，而且还容易导致其它数据质量问题的产生。本文主要研究数据质量中的一个重要课题-实体匹配问题，即识别单源或多源数据中指向同一实体的数据库记录。本文提出了两类实体匹配算法，基于结构化非主属性数据的实体匹配算法和基于文本类型非结构化数据的实体匹配算法。此外，实体匹配的另外一个问题是匹配效率问题，传统的一对一比较方法匹配效率很低，产生了巨大的时间开销，为此本文还提出了一个数据分块算法。本文的具体研究内容如下：

(1) 对数据质量进行了讨论和分析，列举了常见的数据质量问题。介绍了实体匹配的概念，深入研究了已有的实体匹配相关理论和方法，并对其优缺点进行了分析，说明了实体匹配的现实意义。

(2) 提出了基于结构化非主属性数据的实体匹配算法。通过对非主属性的筛选，寻找非主属性的组合进行实体匹配。为了较好地利用结构化非主属性，本文提出了两种基于结构化非主属性的实体匹配算法，其中的baseline 算法通过考虑非主属性区别一实体不同于其它实体的能力筛选非主属性，结合相似度函数计算实体相似度，另外一种基于PRTree 的提升算法首先利用非主属性建立基于规则的概率决策树，然后利用该树结构进行实体匹配计算相似度，最后结合相似度阈值决定匹配结果。通过该类方法一方面可以提高匹配的准确率和召回率，另一方面可以提高匹配的效率。

(3) 提出了基于文本类型非主属性数据的实体匹配算法。文本数据中虽然含有一些噪数据影响了匹配结果，但这些数据中可能包含了一些结构化数据中所没有的信息，通过对这些信息的使用能够有效地提高实体匹配的效果。但从文本数据中挖掘有效信息并非易事。为了能够通过文本数据提升实体匹配的效果，本文提出了基于文本类型非主属性数据的实体匹配算法，其中的baseline 算法通过考量短语的IDF



得分从文本数据中发现重要的信息，然后构建实体的比较向量计算相似度，另外一种基于短语共现的实体匹配方法则是通过发现短语之间的共现关系建立短语共现图。考虑到共现图中一些弱关联关系的存在，阻碍了子主题的获取，需对其进行分割，本文提出了一个贪心式的分割算法，进而获取分割后的每个划分的子主题及其权重，然后利用子主题计算实体的相似度，最后结合相似度阈值决定匹配结果。此外，在进行实体匹配的过程中还面临着严重的匹配效率问题，为了解决该问题本文提出了数据分块算法，将可能匹配的实体分配到同一个块中，极大的减少了不必要的比较，使得实体比较次数大大降低。

## 5.2 工作展望

本文对多形态非主属性数据的实体匹配算法进行了研究，提出了基于结构化非主属性数据的实体匹配算法和基于文本类型非主属性数据的实体匹配算法，通过与以往方法对比，在真实数据集上验证了本文提出算法的效果，实验结果表明所提出算法取得了较好的匹配效果，在效率方面也优于其它算法。然而，其中仍有可以改进的地方：

(1) 基于PRTree的方法在匹配准确性方面还有待提升。在未来的工作中，我们会考虑如何更好地解决错误属性值对匹配的影响。我们还考虑能否将数据纠错和实体匹配相结合，实现二者之间的交互，使二者相互促进。一方面更多匹配的实体有利于纠正更多的错误值，而更多错误值的纠正有利于发现更多匹配的实体。此外，我们还考虑将众包技术（Crowdsourcing）融入到我们的方法之中，将一些机器无法做出的判断交给众包处理，从而进一步提升实体匹配的效果。

(2) 基于短语共现的实体匹配算法也有一些需要改善的地方，在使用提取的sub-topic进行实体匹配时，我们并没有使用所有的子主题，忽略了一些权重较小的子主题，但这种方式存在一些问题。比如有些子主题的权重虽然不高，但对实体匹配的结果会产生很大的影响。例如，在房源数据的CText中可能存在一些对招租对象的限制，可能只针对单身也可能只租给有家庭的人。如果房源的招租对象不同，即使具有相同或相似的结构化数据和其他非结构化数据，仍有可能不是同一房源。众包作为一种非常重要的技术，能够做一些计算机做不了或者做不好的事情。因此，

我们可以考虑使用众包对那些具有较低权重的子主题进行判断，确定子主题是否应该被忽略，如果众包认为不应该忽略，则让众包提供合适的权重分配给该子主题，进而结合新发现的子主题和原来的子主题进行实体匹配。

（3）相似度阈值的设定。本文中提出的算法所使用的相似度阈值是通过大量实验获得的，如果在实际应用中将会影响算法的效果。对此，可以使用一些机器学习的方法或众包的方式去自动调整阈值。通过这些方法可以使算法更好的适应其它数据集，进而获得更好的匹配效果。

## 攻读硕士学位期间发表的论文

### 期刊论文:

[1] 杨强, 李直旭, 蒋俊, 赵朋朋, 刘冠峰, 刘安, 周晓方. 基于非主属性值的实体匹配[J]. 计算机学报, 2016, 39(10): 2075-2087. (EI, 中文核心期刊)

[2] Binbin Gu, Zhixu Li, **Qiang Yang**, Qing Xie, An Liu, Guanfeng Liu, Kai Zheng and Xiangliang Zhang. Web-ADARE: A Web-Aided Data Repairing System[J]. Neurocomputing, 2016年. (SCI, CCF C类, 已录用)

### 会议论文:

[1] **Qiang Yang**, Zhixu Li, Binbin Gu, An Liu, Guanfeng Liu, Pengpeng Zhao and Lei Zhao. CTextEM: Using Consolidated Textual Data for Entity Matching [A]. International Conference on Database Systems for Advanced Applications [C]. Springer International Publishing, 2016: 117-132. (EI, CCF B 类)

[2] **Qiang Yang**, Zhixu Li, Jun Jiang, Pengpeng Zhao, Guanfeng Liu, An Liu and Zhu Jia. NokeaRM: Employing Non-Key Attributes in Record Matching [A]. International Conference on Web-Age Information Management [C]. Springer International Publishing, 2015: 438-442. (EI, CCF C 类)

[3] Jian Zhou, Zhixu Li, **Qiang Yang**, Jun Jiang, An Liu, Guanfeng Liu, Lei Zhao and Jia Zhu. HouseIn: A Housing Rental Platform with Non-Redundant Information Integrated from Multiple Sources [A]. Asia Pacific Web Conference [C]. Springer International Publishing, 2015: 859-862. (EI, CCF C 类)

[4] Jun Jiang, Zhixu Li, **Qiang Yang**, Pengpeng Zhao, Guanfeng Liu and Lei Zhao. SmartInt: A Demonstration System for the Interaction between Schema Mapping and Record Matching [A]. International Conference on Web-Age Information Management [C]. Springer International Publishing, 2015: 587-589. (EI, CCF C 类)

### 申请发明专利:

[1] 李直旭, 杨强等. “一种用于实体匹配的方法及系统”. 申请日期: 2015年07月14日, 专利申请号: 201510407893.6

[2] 杨强, 李直旭等. “一种用于实体匹配的方法及系统”. 申请日期: 2015年06月16日, 专利申请号: 201510329196.3

[3] 李直旭, 杨强等. “一种数据填充方法及系统”. 申请日期: 2015年06月08日, 专利申请号: 201510305681.7

[4] 李直旭, 杨强等. “一种基于质量控制的数据填充方法及系统”. 申请日期: 2015年06月08日, 专利申请号: 201510304683.2

**软件著作权:**

[1] 杨强, 李直旭等. “基于非主属性的概率决策树实体匹配软件”. 申请日期: 2015年01月23日, 登记号: 2015SR077638

**参与的科研项目:**

[1] 国家自然科学基金青年项目(61402313) “基于互联网海量信息的数据库文本类型数据清洗研究”, 2015/01-2017/12

## 致 谢

时光荏苒，岁月如梭，转眼间就要挥手告别研究生生活了。回顾这三年来的点点滴滴，我获得丰富的专业知识，更获得了许多宝贵的经验与回忆，同时也收获了许多感动。此时此刻，我要对这三年以来给予我帮助的亲人、老师、同学们表示衷心的感谢！

首先，我要感谢我的家人，感谢你们对我的支持与付出。是你们用辛勤的双手给了我良好的生活环境，也给了我克服困难的勇气。正是有你们的陪伴和鼓励，才使我不断突破自我，克服重重困难险阻，找到自己的人生定位。在今后的人生路上，有你们的陪伴也是我最大的幸福。

其次，我要衷心地感谢我的导师李直旭副教授。在我的研究生这三年中，李直旭老师无论在学术上还是生活中都给了我极大的帮助。在学术方面，李老师用渊博的学术知识、严谨的治学精神和正直的生活态度深深地感染了我，让我受益匪浅，在学术上对我进行悉心的教导，纠正我学术上错误；在生活方面，李老师教导我如何提升自信心，鼓励我多锻炼自己，这些方方面面对我以后的生活产生了都将产生极大的影响。此外，我能够顺利地完成学业、发表多篇科研论文都离不开李老师无微不至的关心与爱护。在此，我要再次衷心地感谢李直旭老师，您是我今后学习的榜样，您的谆谆教导让我终生难忘。

接着，我要感谢实验室的所有同学们，感谢你们在学习和生活上无微不至的关怀和帮助。在此，我要特别感谢我的师兄蒋俊和师妹何莹，正是你们的帮助才让我顺利完成学术研究。同时，感谢顾斌斌、林天巧、旷晓鹏和周剑等同学在学术方面对我的帮助。接下来，我要感谢我的舍友们，与你们的朝夕相处，使我们建立起了深厚的友谊。

最后，感谢各位老师百忙中对我的论文进行评审。

杨强

二〇一七年三月二十五日