

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，将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 Qiang Yang

Supervised by Zhixu Li

目 录

第一章 绪论.....	1
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
第二章 相关理论及方法.....	6
2.1 数据质量	6
2.1.1 同一数据源内的质量问题	6
2.1.2 数据源之间的质量问题	9
2.2 实体匹配概念以及现有的实体匹配方法.....	10
2.2.1 基于内容的匹配方法	11
2.2.2 基于结构的匹配方法	13
2.2.3 混合方式匹配方法	14
2.3 本章小结	15
第三章 基于结构化非主属性数据的实体匹配	16
3.1 问题定义	16
3.2 预备工作	17
3.3 基于非主属性区分度的实体匹配算法	18
3.4 基于规则的概率决策树的实体匹配算法.....	18
3.4.1 算法基本思路	19
3.4.2 基于规则的概率决策树的构建	20
3.4.3 实体匹配过程	24
3.4.4 基于PRTree 的NokeaEM 算法复杂度分析.....	28
3.4.5 基于PRTree 的NokeaEM 算法的扩充算法.....	28
3.5 实验结果及分析.....	29
3.5.1 参数设定	30
3.5.2 与以往方法在匹配效果方面的比较	31

3.5.3	与以往方法在匹配效率方面的比较	33
3.5.4	算法的扩展性评估	34
3.6	本章小结	34
第四章	基于文本类型非主属性数据的实体匹配	36
4.1	问题定义	36
4.2	传统的文本数据处理方法	38
4.3	数据分块算法	39
4.4	基于IDF 的迭代式实体匹配算法	40
4.4.1	Baseline算法的工作流程	40
4.4.2	短语IDF 得分的迭代式更新	41
4.5	基于短语共现的实体匹配算法	41
4.5.1	共现图的构建	42
4.5.2	PC-Graph的分割	43
4.5.3	子主题及其权重的获取	45
4.5.4	实体匹配过程	46
4.6	实验结果及分析	47
4.6.1	与以往方法在匹配效果方面的比较	47
4.6.2	算法提取结果的评估	49
4.6.3	算法的扩展性评估	50
4.7	本章小结	51
第五章	总结与展望	52
5.1	全文总结	52
5.2	工作展望	53
	参考文献	55
	攻读硕士学位期间发表的论文	63
	致谢	65

第一章 绪论

1.1 课题研究背景

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，但是数据质量却表现出明显的下降趋势，导致可用的数据并未同比例增加。各种数据质量问题也随之产生，如数据失真、数据过期、数据缺失、数据表达不一致等问题，这些问题随着数据量的增加日益突出^[1]。在这种情况下，如何获取高质量的数据成为政府和企业关注的焦点，通过高质量数据政府可以对经济的发展趋势作出合理的分析和预测，同时企业利用高质量数据可以为用户提供更好的个性化服务，提升用户的满意度。

在海量数据中存在着各种类型的数据，如字符串型数据、数值型数据和文本数据等。但是在这些数据中存在大量的表达不一致数据，严重阻碍了数据的高效使用。实体匹配是指识别同一数据源中或不同数据源间指向同一实体的数据源记录^[2,3]。通过实体匹配可以发现表达不一致数据，从而提升数据的准确性、一致性和确定性，甚至在一定程度上有助于提升数据的完整性和时效性，使得数据质量得到极大的提升。通过实体匹配方法和技术，政府和企业可以从海量数据中挖掘出其中蕴藏的“宝藏”、充分获取其中的普遍规律，从而作出更为更符合实际情况的决策，为社会创造更多的价值。

数据库中通常包含多种形态的数据，如结构化数据和文本类型数据，但这些数据存在表达不一致的问题，含义相同的主属性值形式上可能千差万别，而含义不同的主属性值也可能表示同一实体。传统的一些实体匹配算法^[4,5]通过字符串相似度函数计算实体的相似度，如编辑距离、Q-gram、余弦相似度等，但容易受到数据表达多样化的影响。因此，传统的实体匹配方法很难保证较高的匹配准确性和很强的数据可用性。在使用结构化数据和非结构化数据进行实体匹配时，除了数据表达形式多样的问题外，还存在一些其它的问题。例如，在基于结构化非主属性数据的实体匹配方法中，选择哪些非主属性参与实体匹配，以及如何对非主属性识别实体的能力进行评估等是该类算法的难题。而在基于文本类型非主属性数据的实体匹配方法中，如何从文本数据中获取高质量的信息也是一个很棘手的问题。因此，如何利用

结构化非主属性数据和文本类型非主属性数据获得较好的实体匹配效果成为了实体匹配研究的重点和难点问题。

1.2 课题研究现状

实体匹配方法大致可分为基于主属性数据的实体匹配方法和基于非主属性数据的匹配方法，其中基于非主属性数据的实体匹配方法又可细分为基于结构化非主属性数据的匹配方法和基于文本类型非主属性数据的匹配方法，两类方法在匹配的准确性上存在很大的差异。目前已有的实体匹配方法大都是基于主属性数据的匹配方法^[6]。这类方法通常使用编辑距离，Jaccard 距离等字符串相似度函数度量实体的相似度。主属性虽然能够唯一标识某一实体，但易受数据表达方式多样化的影响。例如主属性值为“Li Hua”和“Hua Li”的两个实体虽然在表达方式上不相同，但实际上二者表示同一实体。又或是主属性值直接无法使用，如主属性值中出现了丢失数据和噪声数据，都会使得传统的基于主属性数据的实体匹配方法难以获得较高的准确性。

基于非主属性数据的实体匹配方法则可以解决上述问题，原因在于：（1）非主属性数量相对于主属性而言在数量占据优势；（2）非主属性较主属性而言蕴含了更多的信息，对于提升匹配结果具有很大的帮助。因而，一些基于非主属性的实体匹配方法被提出，文献 [7] 提出了一种使用非主属性创建匹配树的实体匹配算法，该匹配树拥有非0 即1的两类分支，通过判断每个节点的匹配情况决定最后的匹配结果，但是该方法一方面存在匹配效率低的问题，另一方面存在健壮性的问题，容易受缺失值的影响。

在有些情况下，由于结构化非主属性数据的缺乏导致没有足够的信息可以判断实体是否匹配，使得实体虽然具有相同或相似的结构化非主属性数据，但是仍然可能是不匹配的。注意到文本数据中可能隐藏了实体的某些特性，这些特性对于提升匹配的准确性起着决定性作用。传统的一些处理文本数据的主题模型^[8-10]，如LDA、LSA、PLSA 等，能够从自由文本数据中识别一些诸如“education”、“financial”、“sports”等主题，取得了较好的提取效果。但是实体的文本描述信息中的主题则被认为是一般主题的子主题，因为它们共享许多主题词而且

这些主题之间并没有明显的界限。以前的主题模型在处理实体的描述信息时效果会受到极大的影响。此外,还有一些其它类型的文本数据处理方法被提出,并将之应用到实体匹配中。文献 [11] 提出了一种使用字符串得分和语义相似性得分结合的匹配方法,但这种方法依赖于WordNet,在某些特定领域的数据集中能够获得较好的匹配效果,对于其它数据集的匹配结果则难以保证。文献 [12] 提出了一种基于语义特征的方法,但受限于特定的语义模型,导致其适用性不强。

上述方法虽然可以解决部分实体匹配问题,但仍旧存在一定的缺陷与不足。因此,如何有效地利用多形态非主属性数据进行实体匹配成为了本文的研究重点。

1.3 课题研究内容

本文研究了基于多形态非主属性数据的实体匹配问题,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,提高了实体匹配的准确率和召回率。此外,为了解决实体匹配效率低的问题,本文提出了一种数据分块算法将可能匹配的实体尽可能的放在同一个块中。本文主要解决了以下三个问题:(1)如何使用结构化非主属性数据进行实体匹配;(2)如何使用文本类型非主属性数据进行实体匹配;(3)如何提高实体匹配的准确性和效率。本文的主要内容如下:

(1) 本文着重研究了实体匹配问题,介绍了已有的实体匹配方法,描述了具有代表性的各类算法的优势,并分析了其中存在的问题。

(2) 提出了基于结构化非主属性数据的实体匹配算法。传统的基于主属性数据的方法容易受到数据表达方式不一致的影响,使得匹配准确性难以得到保障。而数据集中存在许多结构化的非主属性数据,它们刻画了实体的特点。因此,本文提出了基于结构化非主属性的实体匹配算法,通过对非主属性的筛选,计算每个非主属性识别实体的能力,通过寻找非主属性的组合进行实体匹配,而不是使用所有的非主属性进行匹配。本文提出的基于结构化非主属性的实体匹配算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。使用结构化非主属性数据虽然能够发现匹配的实体,但是有些情况下我们发现实体虽然具有相同或相似

的结构化非主属性数据，它们仍有可能是不匹配的。而数据集中存在一些实体的文本描述信息，其中虽然含有一些噪数据，但是可能包含了一些结构化数据中所没有的信息。因此，本文提出的基于文本类型非主属性数据的实体匹配算法，通过对文本数据的挖掘，可以找出实体的特征，利用该特征并结合结构化数据进行实体匹配，可以提高实体匹配的准确率和召回率。此外，实体匹配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。通过该算法将可能匹配的实体分配到同一个数据块中，使得实体比较次数大大降低。

1.4 课题研究意义

实体匹配可以找出单个数据源内或多个数据源之间指向同一实体的数据库记录，进而减少数据表达不一致问题的出现，并在一定程度上提升了数据的完整性和确定性。通常情况下可以利用主属性数据对实体是否匹配进行判别，因为主属性可以唯一标识实体，但是主属性数据存在表达方式多样化的问题，使得难以获得较高的准确率和召回率。而本文提出的基于结构化非主属性的实体匹配方法则可以很好的解决主属性方法中存在的问题。使用结构化非主属性数据进行实体匹配并非易事，因为不同的非主属性在识别实体的能力方面各不相同，选择哪些非主属性参与实体匹配是一个难题。本文提出了基于非主属性数据的实体匹配方法通过对非主属性识别实体能力的分析与考量，选择识别力较大的非主属性，构建非主属性组合序列，利用生成的序列进行实体匹配。本文所提出的方法不仅克服以往方法中的缺点，提高匹配的准确率和召回率，还能在一定程度上降低匹配的比较开销，提升了匹配的速率。

在有些情况下，结构化非主属性并不能反映真实的匹配情况。实体即使具有相同结构化非主属性值，仍有可能是不匹配。但数据集中关于实体的文本描述信息可能包含了实体的一些特征信息，通过对文本信息的挖掘可以找出实体的特性。但处理文本数据并非易事，实体的文本描述信息通常描述了实体的一个主题的多个方面，使得传统的主题模型难以适用。而且文本描述信息中含有大量的噪数据。本文提出的基于文本类型的非主属性数据的实体匹配算法，通过对文本数据特征的提取，可以将具有相同或相似结构化非主属性数据的实体识别出来，极大的提高了实体匹配

的准确性。在实体匹配过程中还存在匹配效率低的问题，本文提出了数据分块算法，将具有相似的结构化非主属性数据的实体放到同一个数据块中，减少了实体间的比较次数。

在众多含有噪数据和表达不一致数据的数据集中，通过基于多形态非主属性数据的实体匹配算法发现匹配的实体，除了能够极大地提升实体匹配的准确率和召回率以及效率，还具有很大的现实意义。一方面政府可以更方便的使用一致的、正确的、无冗余的高质量数据，企业可以从高质量数据中挖掘用户的喜好，为用户提供更为个性化的定制服务，另一方面实体匹配工作对数据挖掘、知识发现、数据分析等工作也具有十分重要的意义。

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，给出了数据质量的相关概念，并结合例子阐述了常见的数据质量问题。此外，介绍了实体匹配的概念，通过例子说明了实体匹配过程中遇到的主要问题、面临的挑战。最后介绍了一些常见的实体匹配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结构化非主属性数据的实体匹配算法，首先给出问题定义，介绍了基于非主属性数据的baseline 匹配算法，然后介绍了本文提出的基于规则的概率决策树的提升算法，并通过实验证明所提出算法的效果。

第四章介绍了本文提出的基于文本类型非主属性数据的实体匹配算法，首先给出问题定义，描述了传统的文本数据处理方法，以及这些方法在实体匹配中存在的问题。然后介绍了基于文本类型非主属性数据的实体匹配算法的工作流程，主要介绍了其中的baseline 算法和基于短语共现模型的匹配算法。最后通过实验证明了所提出算法的效果。

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

本章首先介绍了数据质量的相关概念及其定义，并描述了常见的数据质量问题。然后给出了实体匹配的概念，并对实体匹配方法进行了分类。最后详细介绍了常见的实体匹配方法，并分析了各类方法的优缺点。

2.1 数据质量

前人给出了数据质量的各种定义，文献 [13]指出数据质量是数据适合于使用的程度，数据质量越高，适用的程度也随之越高；文献 [14]指出数据质量是数据满足用户某种特定需求的期望程度；文献 [15]认为数据质量是数据在多大程度上满足了模式层和实例层的一致性、正确性、完整性和最小性的要求。文献 [16, 17]认为数据质量可精简地由以下六个方面组成（如图 2-1 所示）：准确性（Accuracy）、一致性（Consistency）、完整性（Integrity）、时效性（Timeliness）、有效性（Validity）和完备性（Completeness）。数据质量问题具有多种分类方式，本文则从数据源的角度对其进行分类，包括同一数据源内的质量问题和不同数据源之间的质量问题，其分类结果如图 2-2 所示。下面对这两类数据质量问题进行简要的描述。

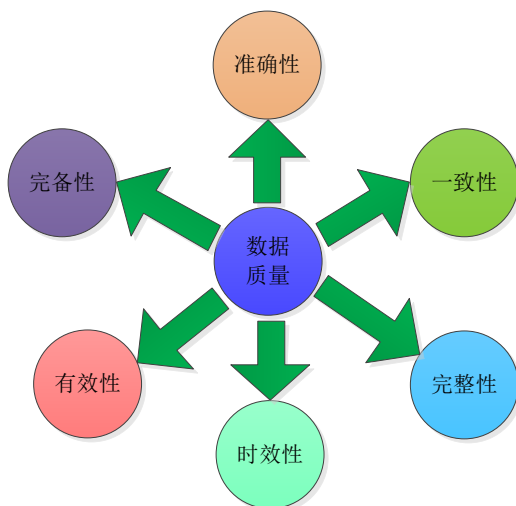


图 2-1 数据质量的六个维度

2.1.1 同一数据源内的质量问题

该类问题主要集中在模式层和实例层。常见的模式层问题^[18]如下所示：

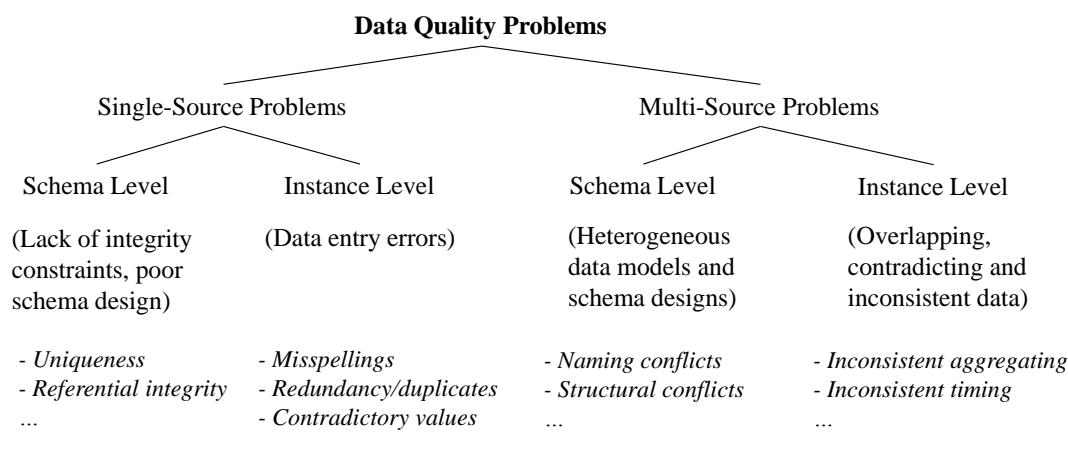


图 2-2 数据质量问题分类

(1) 唯一性约束问题 (Uniqueness Violation Problem)。如学生信息表 2-1 中学生的 ID 为主属性，可以唯一标识一名学生，因而属性值必须唯一，不能重复，但表中 “Wei Wang” 和 “Hua Li” 的 “ID” 却相同，解决此类问题可借助外部资源。

(2) 参照完整性问题 (Referential Integrity Violation Problem)。如表 2-1 中的 “Department ID” 需要参照院系信息表 2-2，而 “Qiang Wang” 的 “Department ID” 属性值 “005” 在表 2-2 中并不存在，此时便产生了参照完整性问题，同样可借助外部资源解决此问题。

(3) 非法值问题 (Illegal Values Problem)。如表 2-1 中学生 “Wei Wang” 的 “Age” 值为 “1000”，学生 “Hua Li” 的 “Birthday” 值为 “1993.13”，然而正常人的年龄值不可能为 1000，任何一个人也不可能出生在 13 月，这些值都是非法的，不符合实际情况，可通过约束方法找出此问题，并使用参照数据集解决。

(4) 属性依赖约束问题 (Violated Attribute Dependencies Problem)。如表 2-1 中 “Zhao Li” 的 “Age=24” 而 “Birthday=1991.05”，我们可以发现 “Age” 的值并不等于当前的年份减去属性 “Birthday” 值，此时产生了依赖约束问题，可使用参照数据集解决。

对于上述模式层的问题，目前较为常见的处理方式是改进模式设计、增加完整性约束等^[19]。相比而言，实体层上的数据问题比模式层上的问题更为复杂，文献 [18] 给出了实体层的主要质量问题及解决方案：

(1) 缺失值问题 (Missing Values Problem)。如表 2-2 中院系编号 “002” 的职工数 “# Faculty” 为空, 该问题是信息系统和数据库中普遍存在的一个问题^[20], 对于缺失数据的填补工作前人已经做出了大量研究^[21,22]。解决数据丢失的常见方法包括: 1) 根据属性的依赖关系 (如 Function Dependency, Conditional Function, Integrity Dependency 等), 借助数据集中的其它实体进行填补; 2) 基于模型的数据填补方法, 通过利用已有的数据建立预测模型对缺失值进行预测; 3) 利用外部资源进行填补。

(2) 缩写问题 (Abbreviations Problem)。如表 2-2 中院系编号 “004” 的院系名称 “BSME” 为缩写, 其完整值为 “Biological Science and Medical Engineering”。解决缩写类问题的常见方法是借助外部知识建立全称和缩写简称的对应表。

(3) 拼写错误问题 (Misspelling Problem)。如表 2-2 中院系编号为 “002” 的实体其 “Name = Computer Sience”, 而实际的值应该为 “Computer Science”。解决拼写错误问题的常见方法也是借助外部知识建立相应的知识库, 在本例中可借助互联网建立院系名称一览表, 将拼写错误信息加以修正。

(4) 矛盾记录问题 (Contradicting Records Problem)。如表 2-1 中学生编号为 “0004” 的 “Qiang Wang” 的两条记录表示的是同一个实体。此外对矛盾的记录进行统一是一个非常棘手的问题, 文献 [23] 提出了通过 Web 搜索引擎获取信息从而决定哪个值更为可靠, 而文献 [24] 则通过 Crowdsourcing 指导进行数据修复。

(5) 表达不一致记录问题 (Duplicated Records Problem), 该类问题又被称为实体匹配问题。如判断表 2-1 中的两个人 (实体) “Qiang Wang” 是否是同一个人, 目前已有大量的研究成果^[25,26], 通常需要选用合适的相似度算法判断实体是否匹配, 但这些方法要么存在匹配准确性的问题要么存在适用范围的问题。

表 2-1 学生信息表

Stud. ID	Name	Sex	Age	Birthday	Department ID
0001	Wei Wang	Male	1000	1990.07	001
0001	Hua Li	Male	24	1993.13	002
0003	Lei Liu	-	26	1991.05	003
0004	Qiang Wang	Male	23	1994.05	005
0005	Zhao Li	Female	24	1991.05	004
0004	Qiang Wang	Male	23	1994.05	002

表 2-2 院系信息表

Dept. ID	Name	Website	#Faculty
001	Computer Science	nd.cs.edu.cn	230
002	Materials Science and Engineering	nd.is.edu.cn	-
003	Automation Science and Electrical Engineering	自动化科学与电气工程	500
004	BSME	nd.cad.edu.cn	380

2.1.2 数据源之间的质量问题

数据源之间的质量问题较数据源内的质量问题更加难以解决，因为不同数据源之间的不一致问题更为突出。常见质量问题如下所示：

(1) 命名冲突问题 (Naming Conflicts Problem)。如表 2-3 中实体 r_1 和表 2-4 中实体 s_1 的产品名称出现冲突，二者表示的都是同一款手机。解决该类问题可通过实体匹配算法进行识别，并统一起来。

(2) 结构冲突问题 (Structural Conflicts Problem)，该类问题是指数据集具有不同的模式层属性，可通过模式识别算法解决该问题。

(3) 数据重复问题 (Entity Overlapping Problem)。如表 2-3 中实体 r_5 和表 2-4 中实体 s_5 就是重复的数据，可通过实体匹配算法解决。

(4) 不一致问题 (Inconsistent Problem)。如表 2-3 中时间的格式为 “YY.MM”，其中月份用的数字，但表 2-4 中时间的格式却为 “MM, YY”，并且月份是英文表示，另外还存在缩写问题。该类问题可通过制定统一的规则进行处理。

(5) 数据丢失问题 (Missing Values Problem)。如表 2-3 中实体 “r2” 在属性 “size” 下出现了缺失值，实体 “s3” 在属性 “Manufacture” 下的值为空。该类问题可通过数据填补解决。

表 2-3 从天猫网站收集的在售手机信息

	Product	Manufacture	Size	RAM	Release
r1	w2013	SAMSUNG	3.7 inches	1GB	2013.04
r2	8295	Coolpad	-	1GB	2013.01
r3	MX II	-	4.4 inches	2GB	2012.12
r4	IPhone 4s	Apple	3.5 inches	512MB	2011.01
r5	G9098	SAMSUNG	3.67 inches	2GB	2014.09
r6	4s	XiaoMi	5.0 inches	3GB	2016.02

表 2-4 从中关村在线网站收集的在售手机信息

	Product	Manufacture	Size	RAM	Release
s1	Galaxy w2013	SAMSUNG	3.7 ”	1G	April, 2013
s2	8295	Coolpad	4.7 ”	1G	Jan., 2013
s3	MeiZu X2	MeiZu	4.4 ”	2G	2012, 12
s4	4s	Apple	3.5 ”	-	January, 2011
s5	G9098	SAMSUNG	-	2G	Sept. 2014

2.2 实体匹配概念以及现有的实体匹配方法

实体匹配是指找出同一数据源内或不同数据源间指代同一实体的数据库记录^[27,28]，这些实体具有不同的表达形式。以手机实体为例，实体匹配就是指找出表 2-3 和表 2-4 中表示同一款手机的数据库记录。通过观察，发现实体对 $\langle r_1, s_1 \rangle$ ， $\langle r_2, s_2 \rangle$ ， $\langle r_3, s_3 \rangle$ ， $\langle r_4, s_4 \rangle$ ， $\langle r_5, s_5 \rangle$ 分别表示同一实体，尽管它们的某些属性在表达形式上不同，但在现实世界中它们是同一实体，而实体对 $\langle r_6, s_4 \rangle$ 虽然具有相同的产品名称“4s”，但是本质上它们是不同的实体。实体匹配近年来一直被广泛地研究，其应用领域涉及医疗卫生^[29]、信息检索^[30]、商业数据管理^[31] 等。

实体匹配方法有多种分类方式，本文主要从数据的角度进行分类，大致可分为三类：基于内容的匹配方法、基于结构的匹配方法和基于混合模式的匹配方法，图 2-3 给出了实体匹配方法分类树^[32]，下面对已有的实体匹配方法做简单的介绍。

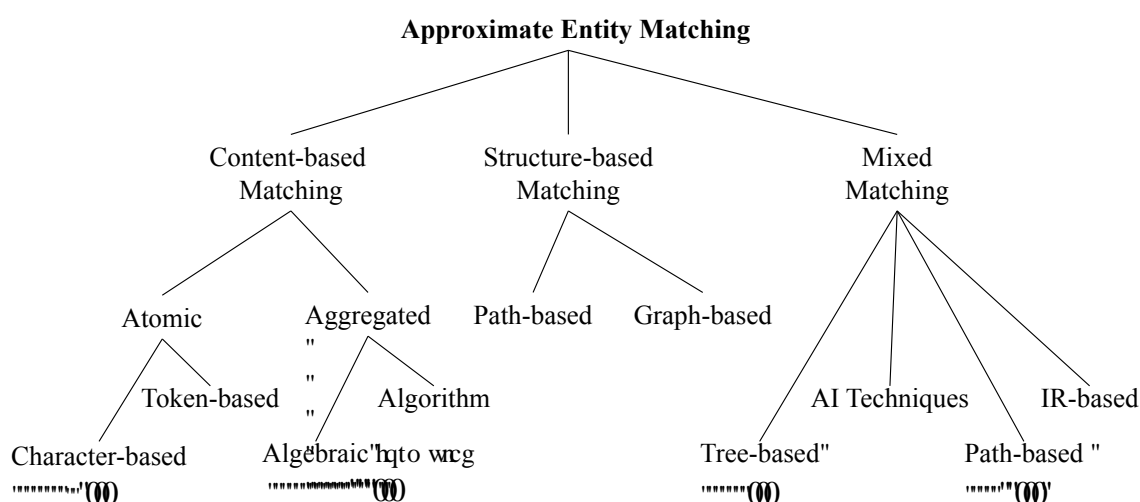


图 2-3 实体匹配方法分类

2.2.1 基于内容的匹配方法

基于内容的匹配方法（Content-based Matching Methods）主要通过一个相似度函数（如edit distance similarity, Q-gram similarity 等）度量数据值上的相似性，进而比较两个实体是否匹配。这些数据值可以是数据表中的属性值，XML 文档中标签值等。此类方法大致可分为两大类：基于Atomic values 的方法和基于Aggregated values的方法。

（1）基于Atomic values 的方法通过相似度函数比较原子值的相似性得分，又可细分为基于字符的方法^[33]（character-based）和基于符号的方法^[34]（token-based），前者进行字符间的比较，比如Levenshtein 方法，Jaro 方法和Q-grams 方法，后者则是进行符号之间的比较，这些符号可以是单词，或者子串，比如SoftTFIDF 方法，MongeElkan 方法和Jaccard 方法。此外，还有一些扩展方法利用机器学习技术构建相似度函数进行实体匹配的^[35]。

（2）基于Aggregated values 的方法考虑如何对数据集中单个属性的相似度进行组合而匹配的，可细分为基于代数的方法^[36]（Algebraic formula-based）和基于AI 技术的方法^[37]。前者使用一些度量方式结合属性集合中单个属性值的相似性进行匹配，进而得到实体的相似度，如Euclidean distance, Vectorial model 等，后者则是利用一些AI 技术和关联规则、聚类方法等进行匹配的。此外，还有一些基于有序邻居的方法（Sorted Neighborhood Method, 简称SNM）^[38]，该方法利用主属性值发现比较靠近的实体。如表 2-3 中的“G9098”和表 2-4 中的“G9098”可通过主属性判断是否为同一实体。但是这类方法容易受到表达方式多样的影响，导致匹配的准确性不高。

基于内容的匹配方法又可根据数据集中属性的类型进行分类，可细分为基于主属性数据的匹配方法和基于非主属性数据的实体匹配方法，其中基于非主属性数据的实体匹配方法又可细分为基于结构化非主属性数据的匹配方法和基于文本类型非主属性数据的匹配方法。

（1）基于主属性数据的匹配方法大都是通过各种字符串相似性度量方法进行实体匹配,如基于字符的算法（如Edit distance^[33], Q-gram^[39]）和基于符号的算法（如atomic string^[40], WHIRL^[41]），也有一些混合的方法被提出^[42]，这些方法或是直接用于实体匹配或是间接辅助进行实体匹配，有的方法着重于匹配效率的提升，

如Q-gram算法,而有的方法着重于提高匹配的质量,如WHIRL。文献[43]提出了两种可学习的文本字符串相似度度量方法,包括可学习的字符串编辑距离算法和使用SVM的向量空间度量算法,针对数据表的不同的字段使用不同的方法,在实体匹配过程中该方法具有较好的匹配准确性。近几十年来,一些基于分类^[44]和语义网络^[45]的方法也有被提出,但是这类方法存在着灵活性差,时间开销大等问题。基于主属性的匹配方法容易受到数据表达方式多样的影响,含义相同的主属性值形式上可能千差万别,而含义不同的主属性值也可能表示同一实体。因此,没有任何一种相似度度量方法可以非常准确地度量所有主属性值之间的相似度。此外,这些实体匹配方法的准确率和召回率对相似度阈值也很敏感,阈值的设定会极大地影响匹配的结果。

(2) 鉴于主属性值在表达方式多样化和匹配准确性低等方面的劣势以及结构化非主属性在数目上的优势,一些基于结构化非主属性数据的实体匹配算法被提出,通常是根据结构化数据的相似性^[46]或关联性^[47]进行实体匹配,常见的结构化数据如数值型数据、日期数据和短字符串数据等。文献[48]提出了一种使用正反例的方式并结合相似度连接(similarity join)和相似度结合(similarity unions)以及用户指定的相似度函数构建SJU操作树,通过该结构查询某种适用于待匹配实体属性的相似性度量算法,使得每个属性以最合适的相似性算法进行实体匹配,该方法不仅在小中型数据集上表现不错,而且在大型数据库上也适用。此外,也有一些方法^[49]将结构化非主属性值映射到多维欧几里得空间中,结合给定的合并规则利用属性的相似性选择一个最优的属性集合,应用多维相似性连接判断实体对的相似度。文献[50]中提出了一种基于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,该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。

(3) 虽然基于结构化非主属性数据的实体匹配算法可以发现匹配的实体,但是当没有足够的可用的结构化数据足以反映实体匹配的结果时,也就是说即使实体具有相同或相似的结构化数据,仍有可能不是匹配的实体,在该情形下基于结构化数

据的实体匹配算法很难达到较好的匹配效果，甚至产生更多的误匹配。作为结构化数据的补充，非结构化的文本数据经常出现在各种类型的数据集中。由于每个文本数据中包含了十几个句子甚至更多，而且伴有一些噪数据的出现，使得传统的字符串相似度算法并不能直接的应用。使用文本数据进行实体匹配的关键一步是如何从包含噪数据的文本数据中获取重要信息。近来有一些使用文本数据进行实体匹配的方法被提出。文献 [11]提出了一种混合算法，该算法同时计算文本之间的字符串相似性得分和语义相似性得分，但是字符串相似性只是简单应用了Jaccard 算法，而语义相似度算法也是简单定义了一些字段（如“Address”，“City”，“Phone”，“Type”等），且是在WordNet的辅助下获取相应信息的，因此该方法只使用一些特定的数据集，当换成其他数据集时效果会大大降低。文献 [12]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，然后训练一个分类器可以根据实体的特征向量用以发现重复的实体，但是这种方法也受限于特征向量的维度，因为有些文本不包含这些特征向量中的信息。因此，此方法很难应用于其它数据集。近来，有一些利用主题模型的方法进行实体匹配，但有些情况下自由文本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，并没有明显的不同，因此直接使用这些主题模型很难获得较好的匹配效果。此外，文献 [51]根据单词的依赖关系提出了基于单词共现的方法，可以借助网页中的锚节点文本挖掘隐藏的子主题，但这种方法不仅受限于查询query的质量，而且还需要外部资源的支撑，使得该方法的健壮性受到影响。

2.2.2 基于结构的匹配方法

基于结构的匹配方法（Structure-based Matching Methods）主要通过比较数据值的结构，把值和结构作为唯一的比较单元，并不考虑数据的语义信息，这种方法被大量用于半结构化数据，如XML文档。此类方法大致可分为两类：基于路径的匹配方法（Path-based）^[52]和基于图结构的匹配方法（Graph-based）^[53]。

（1）基于路径的匹配方法是通过比较树结构中一条或多条路径进行匹配的。有的方法通过建立shingles，然后计算待比较实体对的shingles 的交集的集合大小与并集的集合大小的比例作为它们的相似度，而有的方法则使用Temporal Series 比较XML 文档结构的相似性，还有一些方法则是利用离散傅里叶变换（Discrete fourier

transform) 将时间序列从时间域转换为频率域进行比较的。

(2) 基于图结构的匹配方法是通过计算实体间基于图的数据结构的相似度进行匹配的, 有些方法利用基于树的编辑距离 (tree-edit-distance) 来表示XML 文档 (实体) 的相似性。而有的方法则使用边覆盖 (Edge cover) 的原理在创建的二分图上进行实体之间的匹配从而计算匹配相似性。还有一些方法则利用similarity flooding 计算带有标记边的图结构中节点的相似性, 通过对节点相似度的组合获得匹配的相似性。但是, 这些方法对实体的类型要求比较严格, 实体必须是具有一定结构的该方法才能有效。如果实体是关系表中的记录, 该类方法就失效了, 因此存在适用范围小的问题。

2.2.3 混合方式匹配方法

混合方式匹配方法 (Mixed Matching Methods) 同时使用实体的值和结构, 其中结构用于决定哪些值应该用于比较, 此类方法大致可分为如下四类: (1) 基于树的方法 (tree-based) ^[54], 通过结合扩展版的基于树的编辑距离方法和度量算法在树结构上进行相似性得分的计算; (2) 基于AI 技术的方法 ^[55], 该方法考虑如何将聚类算法和逆文档频率相结合应用于图结构中, 利用迭代学习方法计算相似性; (3) 基于路径的方法 ^[56], 同时考虑数据值和与其相关的路径, 并对这两个方面赋予不同的权重, 然后计算总的相似性; (4) 基于IR 的方法 ^[57], 通常使用一些IR技术 (如Vectorial model) 实现匹配, 通过分析将待比较实体的结构和数据值转化为向量形式, 进而比较向量的相似度。

此外, 随着实体数量的增加, 实体匹配的效率会急剧下降, 如何在获得较高的实体匹配准确性的同时, 又能保证较高的匹配效率也是当前的研究热点。为了减少实体对的比较次数, 近年来有许多高效的方法被提出。例如, 一些工作将Q-gram 算法和倒排索引算法 (Inverted Indices) ^[58] 结合, 通过这两种方法可以提高匹配效率, 以尽可能少的时间开销为目的进行实体匹配, 但此类方法的适用性存在一定的问题, 如当参与匹配的实体存在缺失值, 此类方法将不再适用。此外, 还有一些基于前缀修剪 (pre-fix based pruning) 的方法 ^[59]、基于top-k model 的方法 ^[59] 以及基于批量匹配 (batch-based) 方法 ^[60], 通过这些方法过滤那些“明显”不匹配的实体, 可以减少匹配过程中的比较次数。

2.3 本章小结

本章首先介绍了数据质量问题的概念，介绍了同一数据源内的质量问题和多个数据源之间的质量问题，并给出了各类问题的解决方法。然后详细介绍了实体匹配的概念，并对实体匹配方法进行分类，分析了已有的各类方法的优缺点。

第三章 基于结构化非主属性数据的实体匹配

本章首先给出了基于结构化非主属性数据实体匹配的问题定义。然后介绍了与本章算法相关的一些预备工作。接下来提出了两个基于结构化非主属性数据的实体匹配算法，并通过实验验证所提出算法的效果。

3.1 问题定义

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 θ ，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。

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1.

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 τ ，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：

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

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

3.2 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 [50]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。

3.3 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 2-3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：

$$difScore(A_i, T) = \frac{distinct(A_i, T)}{|T|} \quad (3.1)$$

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：

$$\mathcal{F}_{baseline}(t, s) = \frac{\sum_{A \in S_{NK}} (difScore(A) \times sim(t[A], s[A]))}{\sum_{A \in S_{NK}} difScore(A)} \quad (3.2)$$

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 $|T_1| = n$, $|T_2| = m$, $|S_{NK}| = p$ 。

3.4 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。

3.4.1 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3-1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。

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：

$$R = \begin{cases} 1, & \text{if } t_i = s_j \\ 0, & \text{otherwise} \end{cases} \quad (3.3)$$

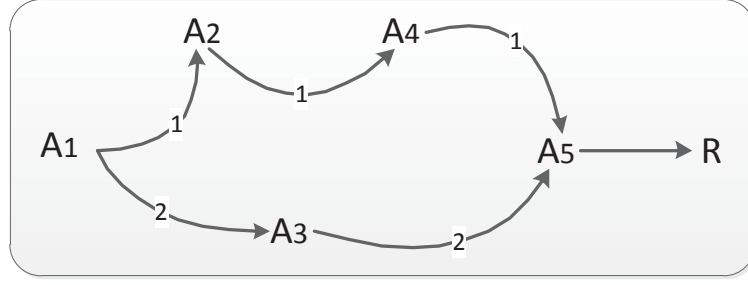


图 3-1 属性间的相关性

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：

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

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：

$$p(S_{NK}) = Pr(R = 1 | S_{NK}) = \frac{Pr(S_{NK} | R = 1) Pr(R = 1)}{Pr(S_{NK} | R = 1) Pr(R = 1) + Pr(S_{NK} | R = 0) Pr(R = 0)} \quad (3.5)$$

3.4.2 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。

基于第 3.4.1 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2.

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 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	Collpad	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	...
...

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, 能够在多大程度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 2 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：

$$suf(A_k) = \sum_{\{(t_i, s_j) | Tr_c\}} \frac{B(t_i = s_j | t_i[A] = s_j[A])}{B(t_i = s_j | t_i[A] = s_j[A]) + B(t_i \neq s_j | t_i[A] = s_j[A])} \quad (3.6)$$

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

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

$$B(bool_1|bool_2) = \begin{cases} 1, & \text{if } bool_2 = \text{TRUE and } bool_1 = \text{TRUE} \\ 1, & \text{if } bool_2 = \text{TRUE and } bool_1 = \text{FALSE} \\ 0, & \text{if } bool_2 = \text{FALSE} \end{cases} \quad (3.8)$$

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 (τ_{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：

$$P_{root} = \max_{A \in S_{NK}} (suf(A), nec(A)) \quad (3.9)$$

(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 3.6 和 3.7 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 3.10所示：

$$P_{nonleaf} = P_{A|Candit(PRTree)} \max_{A \in (S_{NK} - Candit(PRTree))} (suf(A), nec(A)) \quad (3.10)$$

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 τ_{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 3.11所示：

$$P_{leaf} = \max_{A \in (S_{NK} - Candit(PRTree), suf(A) \approx \tau_{stop}, nec(A) \approx \tau_{stop})} (suf(A), nec(A)) \quad (3.11)$$

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 3-2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。

3.4.3 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 “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

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 τ_s 和必要性阈值 τ_n 以及不再继续创建PRTree的阈值 τ_{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 3.6和公式 3.7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  $root$ ;
```

所有祖先节点的条件下来得到的。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 3.12 计算实例对之间的相似性：

$$\mathcal{F}_{PRTree}(t, s) = \prod_{1 \leq i \leq H(A)} (Conf(A_i) * sim(t[A_i], s[A_i]))^{\frac{1}{i}} \quad (3.12)$$

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

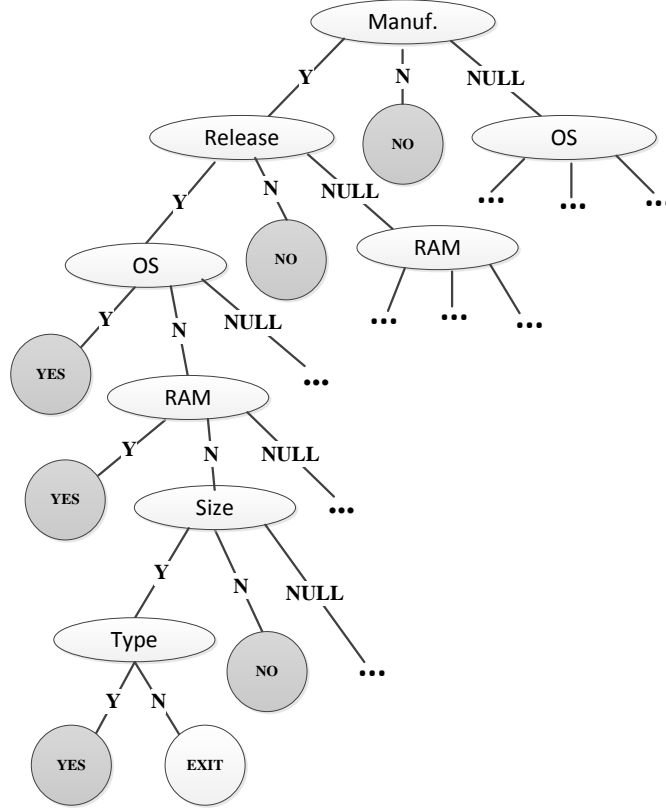


图 3-2 根据算法 1 建立的不包含主属性的 PRTree

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 3.13 所示：

$$Conf(A_i) = \begin{cases} suf(A_i), & \text{if } A_i\text{'s "Matched" branch is in passed} \\ 1 - nec(A_i), & \text{if } A_i\text{'s "Unmatched" branch is passed} \\ 1, & \text{if } A_i\text{'s "Invalid" branch is passed} \end{cases} \quad (3.13)$$

通过上述的步骤结合公式 3.6 至公式 3.13 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。

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

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  $similarity$ ;

```

实体匹配。

(1) 根据公式 3.6—公式 3.13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 3-2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 3.13, 我们得到了图 3-2 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 3-2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 3.12 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 3-2 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

3.4.4 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。

3.4.5 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 3-3 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。

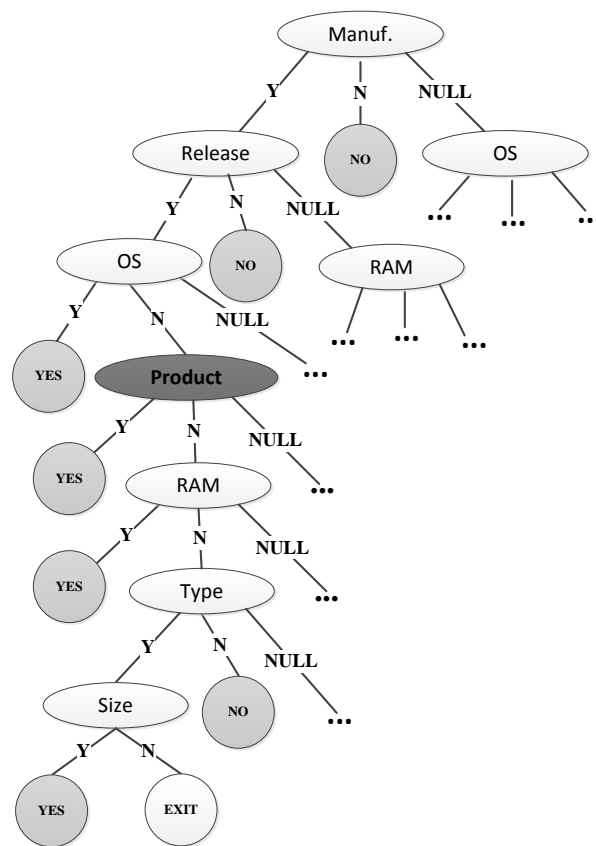


图 3-3 根据算法 1 建立的包含主属性的 PRTree

3.5 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。

3.5.1 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 3-4所示，可以发现F1 Score 随着两个阈值 τ_s 和 τ_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 τ_{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 3-4 (b) 中，由于 τ_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。

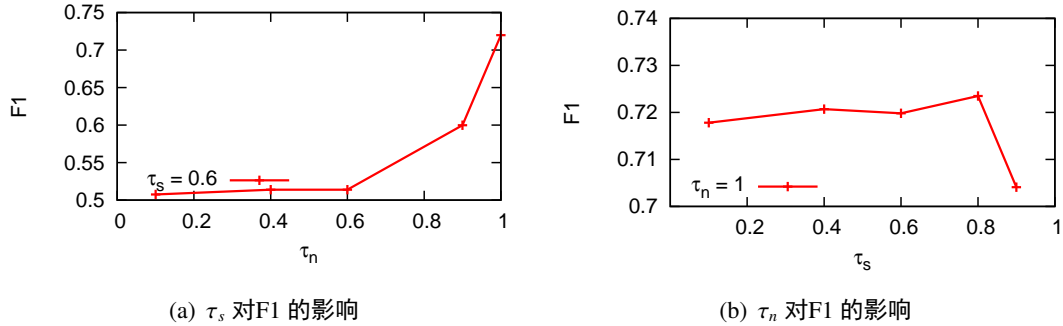


图 3-4 PRTree 的阈值对NokeaEM 的影响

3.5.2 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^[7]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^[39]。除此之外,我们还融入了诸如前缀过滤 (prefix-based filtering)^[50], 倒排索引 (Inverted Indices)^[58] 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 [7]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3-5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3-5（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3-6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。

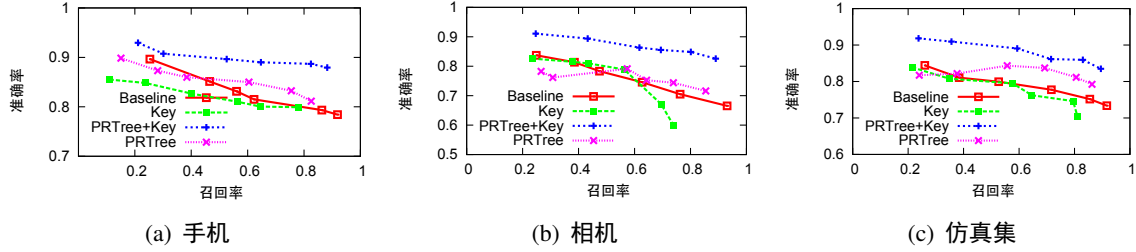


图 3-5 在准确率和召回率方面与以往方法的对比

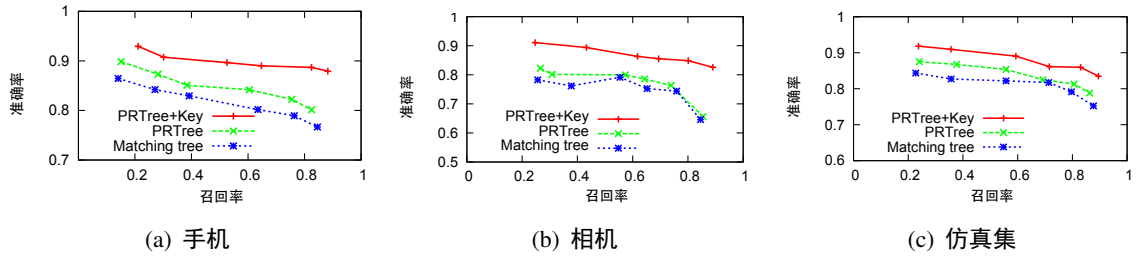


图 3-6 在准确率和召回率方面与Matching tree-based方法的对比

3.5.3 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 3-7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 3-8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。

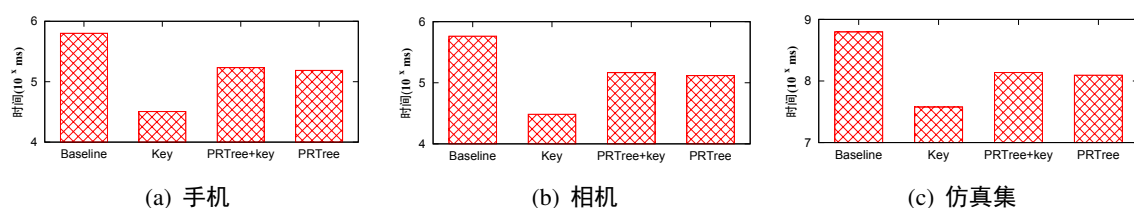


图 3-7 在效率方面与以往方法的对比

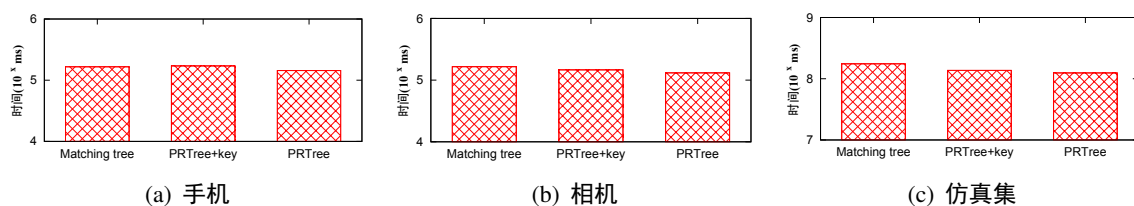


图 3-8 在效率方面与Matching tree-based方法的对比

3.5.4 算法的扩展性评估

如图 3-9 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 3-9 (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 3-9 (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。

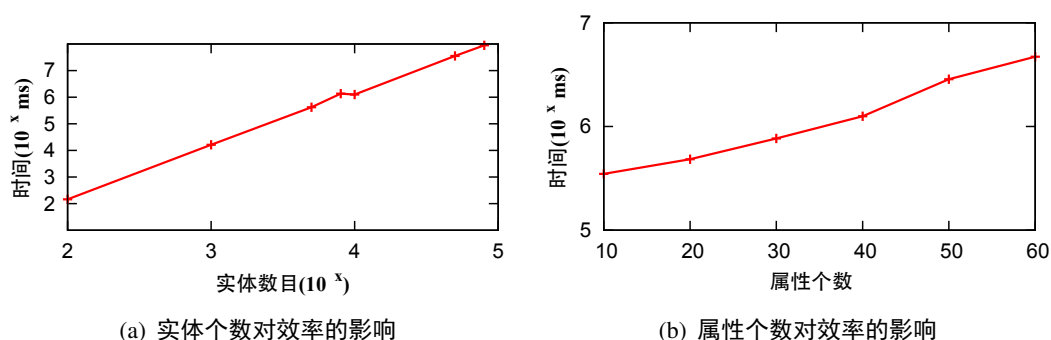


图 3-9 不同数目的记录数和属性数对PRTree在效率上的影响

3.6 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3.

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 θ ，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。

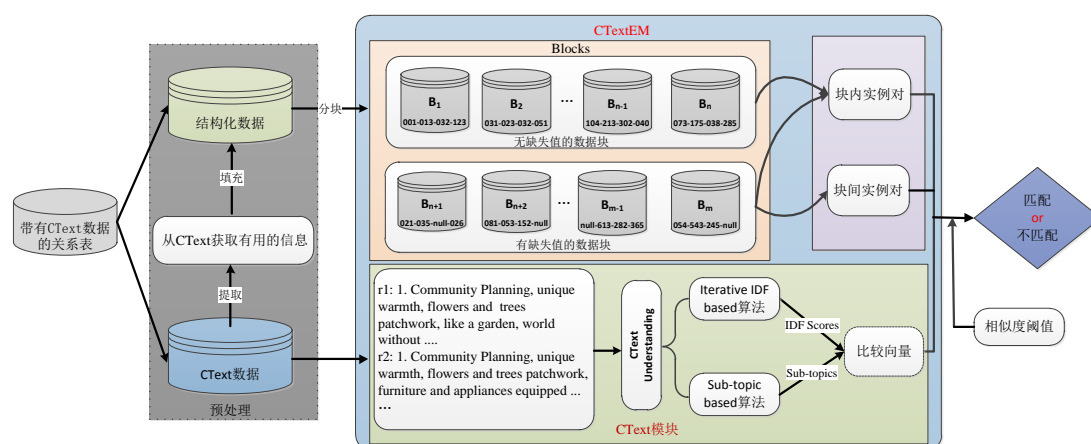


图 4-1 CTextEM 算法的工作流

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 [11] 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 [12] 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 [61] 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^[8], LSA^[10], PLSA^[9]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[51]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[62]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 θ 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^[63]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), θ 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 [64] 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

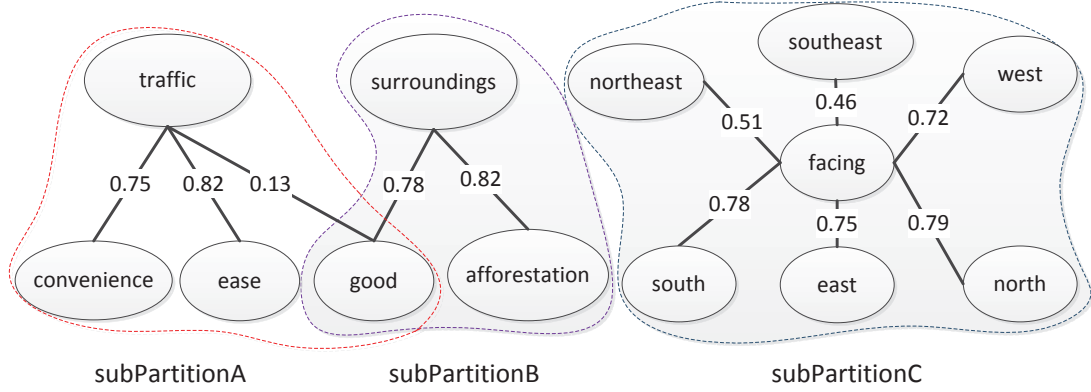


图 4-2 例：具有三个划分的PC-Graph

形式化的表示为公式 4.7 所示的最大化问题：

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

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

其中 α 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 [65]。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 α 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^[66]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^[39]和Inverted Indices^[58]。

（2）*Blocking-based EM*^[67]：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*^[68]：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^[8]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^[69]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

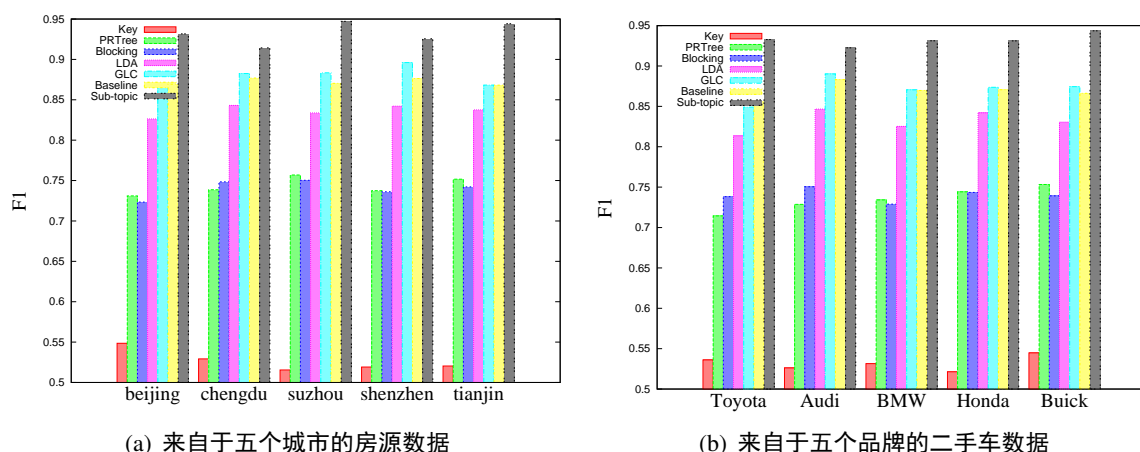


图 4-3 与以往方法在F1 score上的比较

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

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 equipped 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	Community, Planning, warmth, flowers, trees, garden, furniture, appliances, decoration, south, floor, tenant types, family, ...	south, facing, lighting, air, conditioning, water, heaters, washing, machines, property, charges, ...
GLC	Community Planning, warmth, flowers and trees, garden, furniture and appliances, refined, decoration, south, twenty, floor, tenant types, family, ...	south, lighting, two, air conditioning, water heaters, washing machines, free, property charges, ...
baseline	Community Planning, well, warmth, flowers and trees, garden, furniture and appliances, refined, decoration, south, facing, floor, tenant types, family, ...	facing, south, lighting, air conditioning, water heaters and washing machines, property charges, ...
Sub-topic	Community Planning, warmth, flowers, trees, furniture, appliances and decoration, well-groomed, facing, floor, tenant types, family, ...	facing, lighting, air conditioning, water heaters, washing machines, property charges, ...

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。

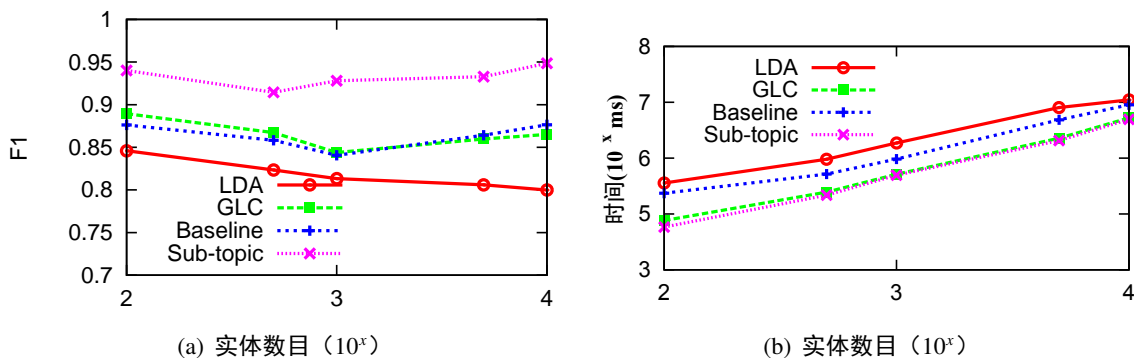


图 4-4 扩展性评估：在F1 score和时间开销方面与一些主题模型的比较

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。

参考文献

- [1] Floridi L and Illari P. The Philosophy of Information Quality [M]. Berlin: Springer International Publishing, 2014.
- [2] Ferro A, Giugno R, Puglisi P. L and Pulvirenti A. An Efficient Duplicate Record Detection Using Q-Grams Array Inverted Index [J]. Lecture Notes in Computer Science, 2010, 6263(11): 309–323.
- [3] Zheng Y, Fen X, Xie X, Peng S and Fu J. Detecting Nearly Duplicated Records in Location Datasets [A]. Agrawal D. Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems [C]. New York: ACM, 2010: 137–143.
- [4] Giunchiglia F, Yatskevich M and Shvaiko P. Semantic Matching: Algorithms and Implementation [J]. Journal on Data Semantics, 2007, 1(4): 1–38.
- [5] Comito C, Patarin S and Talia D. A Semantic Overlay Network for P2P Schema-Based Data Integration [A]. Bellavista P. Proceedings of the 11th IEEE Symposium on Computers and Communications [C]. Washington: IEEE Computer Society, 2006: 88–94.
- [6] Cheatham M and Hitzler P. String Similarity Metrics for Ontology Alignment [M]. Berlin: Springer Berlin Heidelberg, 2013.
- [7] Dey D, Mookerjee V S and Liu D. Efficient Techniques for Online Record Linkage [J]. IEEE Transactions on Knowledge & Data Engineering, 2010, 23(3): 373–387.
- [8] Blei D M, Ng A Y and Jordan M I. Latent Dirichlet Allocation [J]. Journal of Machine Learning Research, 2003, 3(11): 993–1022.
- [9] Hofmann T. Unsupervised Learning by Probabilistic Latent Semantic Analysis [J]. Machine Learning, 2001, 42(1/2): 177–196.

- [10] Landauer T K, Foltz P W and Laham D. An Introduction to Latent Semantic Analysis [J]. *Discourse Processes*, 1998, 25(2-3): 259–284.
- [11] Ektefa M, Jabar M A, Sidi F, Memar S, Ibrahim H and Ramli A. A Threshold-Based Similarity Measure for Duplicate Detection [A]. 2011 IEEE Conference on Open Systems (ICOS) [C]. IEEE Communications Society, 2011: 37–41.
- [12] Gao C, Peng H, Zhao X and Chen H. Web Race Duplication Detection Based on Context [A]. Gong Z. International Conference on Web Information Systems and Mining [C]. Berlin: Springer, 2011: 292–301.
- [13] Huang K T, Yang W L and Wang R Y. Quality Information and Knowledge [M]. Washington: Prentice Hall PTR, 1999.
- [14] Kahn B K and Strong D M. Product and Service Performance Model for Information Quality: An Update [A]. Chengalur-Smith I N. Conference on Information Quality [C]. Cambridge: MIT, 1998: 102–115.
- [15] Aebi D and Perrochon L. Towards Improving Data Quality [A]. Zwass V. International Conference on Information Systems & Management of Data [C]. New Delhi: Citeseer, 1993: 273–281.
- [16] Sadiq S, Zhou X and Deng K. Research and Practice in Data Quality [A]. Zhang Y. Asia-Pacific Web Conference on Progress in WWW Research and Development [C]. Berlin: Springer Berlin Heidelberg, 2008: 41–42.
- [17] Wand Y and Wang R Y. Anchoring Data Quality Dimensions in Ontological Foundations [J]. *Communications of the ACM*, 1996, 39(11): 86–95.
- [18] Rahm E and Do H. Data Cleaning: Problems and Current Approaches [J]. *IEEE Data Engineering Bulletin*, 2000, 23(4): 3–13.

- [19] Beneventano D, Bergamaschi S, Lodi S and Sartori C. Consistency Checking in Complex Object Database Schemata with Integrity Constraints [J]. IEEE Transactions on Knowledge and Data Engineering, 1998, 10(4): 576–598.
- [20] Vattulainen M. Improving the Predictive Power of Business Performance Measurement Systems by Constructed Data Quality Features? Five Cases [M]. Berlin: Springer International Publishing, 2015.
- [21] Liu H and Zhang S. Noisy Data Elimination Using Mutual K-Nearest Neighbor for Classification Mining [J]. Journal of Systems & Software, 2012, 85(5): 1067–1074.
- [22] Tian J, Yu B, Yu D and Ma S. Missing Data Analyses: A Hybrid Multiple Imputation Algorithm Using Gray System Theory and Entropy Based on Clustering [J]. Applied Intelligence, 2014, 40(2): 376–388.
- [23] Chu H and Rosenthal M. Search Engines for the World Wide Web: A Comparative Study and Evaluation Methodology [A]. Mosher E J. Proceedings of the Annual Meeting-American Society for Information Science [C]. New York: ACM, 1996: 127–135.
- [24] Yakout M, Elmagarmid A K, Neville J, Ouzzani M and Ilyas I F. Guided Data Repair [J]. Proceedings of the VLDB Endowment, 2011, 4(5): 279–289.
- [25] Pinheiro J C and Sun D X. Methods for Linking and Mining Massive Heterogeneous Databases [A]. Agrawal R. Knowledge Discovery and Data Mining [C]. New Delhi: Citeseer, 1998.
- [26] DeWitt D J, Naughton J F and Schneider D A. An Evaluation of Non-Equijoin Algorithms [A]. Proceedings of the 17th international conference on very large data bases [C]. Morgan Kaufmann Publishers Inc., 1991: 443–452.
- [27] Fan W, Jia X, Li J and Ma S. Reasoning about Record Matching Rules [J]. Proceedings of the Vldb Endowment, 2009, 2(1): 407–418.

- [28] Arasu A and Domingoferrer J. Record Matching [J]. Encyclopedia of Database Systems, 2009, 2(1): 2354–2358.
- [29] Clark D. E. Practical Introduction to Record Linkage for Injury Research [J]. Injury Prevention Journal of the International Society for Child & Adolescent Injury Prevention, 2004, 10(3): 186.
- [30] Hajishirzi H, Yih W and Kolcz A. Adaptive Near-Duplicate Detection via Similarity Learning [A]. Crestani F. Proceedings of the 33rd International ACM SIGIR conference on Research and Development in Information Retrieval [C]. New York: ACM, 2010: 419–426.
- [31] Su W, Wang J and Lochovsky F H. Record Matching over Query Results from Multiple Web Databases [J]. IEEE Transactions on Knowledge & Data Engineering, 2010, 22(4): 578–589.
- [32] Dorneles C F, Gonçalves R and Santos R M. Approximate Data Instance Matching: A Survey [J]. Knowledge and Information Systems, 2011, 27(1): 1–21.
- [33] Cohen W W, Ravikumar P and Fienberg S E. A Comparison of String Distance Metrics for Name-Matching Tasks [A]. Kambhampati S. Proceedings of IJCAI-03 Workshop on Information Integration on the Web [C]. New York: ACM, 2003: 73–78.
- [34] Dorneles C F, Heuser C A, Lima A E, Silva A S and Moura de E S. Measuring Similarity Between Collection of Values [A]. Laender A H F. Proceedings of the 6th Annual ACM International Workshop on the Web Information and Data Management [C]. New York: ACM, 2004: 56–63.
- [35] Bilenko M, Mooney R, Cohen W, Ravikumar P and Fienberg S. Adaptive Name Matching in Information Integration [J]. IEEE Intelligent Systems, 2003, 18(5): 16–23.
- [36] Motro A. VAGUE: A User Interface to Relational Databases That Permits Vague Queries [J]. ACM Transactions on Information Systems, 1988, 6(3): 187–214.

- [37] Schallehn E, Sattler K U and Saake G. Efficient Similarity-Based Operations for Data Integration [J]. *Data & Knowledge Engineering*, 2004, 48(3): 361–387.
- [38] Hernández A M and Stolfo S J. Real-World Data Is Dirty: Data Cleansing and the Merge/Purge problem [J]. *Data Mining and Knowledge Discovery*, 1998, 2(1): 9–37.
- [39] Aizawa A and Oyama K. A Fast Linkage Detection Scheme for Multi-Source Information Integration [A]. *Proceedings of International Workshop on Challenges in Web Information Retrieval and Integration* [C]. IEEE, 2005: 30–39.
- [40] Monge A E and Elkan C. The Field Matching Problem: Algorithms and Applications [A]. Simoudis E. *KDD* [C]. Berlin: Springer Berlin Heidelberg, 1996: 267–270.
- [41] Cohen W. W. Integration of Heterogeneous Databases without Common Domains Using Queries Based on Textual Similarity [A]. Haas L M. *ACM SIGMOD Record* [C]. New York: ACM, 1998: 201–212.
- [42] Xiao C, Wang W, Lin X, Yu J X and Wang G. Efficient Similarity Joins for Near-Duplicate Detection [A]. Haas L M. *ACM Transactions on Database Systems (TODS)* [C]. New York: ACM, 2011: 1–41.
- [43] Bilenko M and Mooney R J. Adaptive Duplicate Detection Using Learnable String Similarity Measures [A]. Getoor L. *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* [C]. New York: ACM, 2003: 39–48.
- [44] Jiang F, Meng Y, Wei M and Li Q. Entity Identification in Deep Web [M]. Berlin: Springer International Publishing, 2013.
- [45] Dhamankar R, Lee Y, Doan A H, Halevy A and Domingos P. IMAP: Discovering Complex Semantic Matches between Database Schemas [A]. Weikum G. *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data* [C]. New York: ACM, 2004: 383–394.

- [46] Koudas N, Sarawagi S and Srivastava D. Record Linkage: Similarity Measures and Algorithms [A]. Gruhl D. Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data [C]. New York: ACM, 2006: 802–803.
- [47] Parkhomenko E, Tritchler D and Beyene J. Sparse Canonical Correlation Analysis with Application to Genomic Data Integration [J]. Statistical Applications in Genetics and Molecular Biology, 2009, 8(1): 1–34.
- [48] Chaudhuri S, Chen B C, Ganti V and Kaushik R. Example-Driven Design of Efficient Record Matching Queries [A]. Koch C. Proceedings of the 33rd international conference on Very large data bases [C]. London: VLDB Endowment, 2007: 327–338.
- [49] Jin L, Li C and Mehrotra S. Efficient Record Linkage in Large Data Sets [A]. Jeffrey D. Ullman. Proceedings of 8th International Conference on Database Systems for Advanced Applications [C]. Washington: IEEE, 2003: 137–146.
- [50] Wang J, Li G and Feng J. Can We Beat the Prefix Filtering?: An Adaptive Framework for Similarity Join and Search [A]. Candan K S. Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data [C]. New York: ACM, 2012: 85–96.
- [51] Kim S J and Lee J H. Method of Mining Subtopics Using Dependency Structure and Anchor Texts [A]. Calderon-Benavides L. International Symposium on String Processing and Information Retrieval [C]. Berlin: Springer, 2012: 277–283.
- [52] Vinson A R, Heuser C A, da Silva A S and Moura de E S. An Approach to XML Path Matching [A]. Fundulaki I. Proceedings of the 9th Annual ACM International Workshop on Web Information and Data Management [C]. New York: ACM, 2007: 17–24.
- [53] Kailing K, Kriegel H P, Schonauer S and Seidl T. Efficient Similarity Search for Hierarchical Data in Large Databases [M]. Berlin: Springer Berlin Heidelberg, 2004.

- [54] Diego M, Monica S and Tiziana C. Structure-Aware XML Object Identification [J]. IEEE Data Eng. Bull., 2006, 29(2): 67–74.
- [55] Yang J, Cheung W K and Chen X. Integrating Element and Term Semantics for Similarity-Based XML Document Clustering [A]. Skowron A. Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence [C]. Washington: IEEE, 2005: 222–228.
- [56] Park U and Seo Y. An Implementation of XML Documents Search System Based on Similarity in Structure and Semantics [A]. Gruhl D. Proceedings of the 2005 International Workshop on Web Information Retrieval and Integration [C]. Washington: IEEE, 2005: 97–103.
- [57] Carvalho J C and Silva da A S. Finding Similar Identities among Objects from Multiple Web Sources [A]. Chiang R H L. Proceedings of the 5th ACM International Workshop on Web Information and Data Management [C]. New York: ACM, 2003: 90–93.
- [58] Christen P. A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication [J]. IEEE Transactions on Knowledge & Data Engineering, 2011, 24(9): 1537–1555.
- [59] Gal A. Managing Uncertainty in Schema Matching with Top-K Schema Mappings [J]. Lecture Notes in Computer Science, 2006, 6(1): 90–114.
- [60] Chandel A, Nagesh P C and Sarawagi S. Efficient Batch Top-K Search for Dictionary-Based Entity Recognition [A]. Liu L. Proceedings of the 22nd International Conference on Data Engineering [C]. Washington: IEEE, 2006: 28–28.
- [61] Saranyamol C S and Sindhu L. A Survey on Automatic Text Summarization [J]. International Journal of Computer Science and Information Technology, 2014, 5(6): 7889–7893.

- [62] Wu M, Zhang D, Lan W and Wu Q. Text Topic Mining Based on LDA and Co-Occurrence Theory [A]. Gruhl D. Proceedings of the 2012 7th International Conference on Computer Science & Education [C]. Washington: IEEE, 2012: 525–528.
- [63] Kim D, Wang H and Alice O. Context-Dependent Conceptualization [A]. Rossi F. International Joint Conference on Artificial Intelligence [C]. Washington: IEEE, 2013: 2654–2661.
- [64] Guo S, Dong X L, Srivastava D and Zajac R. Record Linkage with Uniqueness Constraints and Erroneous Values [J]. Proceedings of the VLDB Endowment, 2010, 3(1-2): 417–428.
- [65] Sun L, Franklin M J, Krishnan S and Xin R S. Fine-Grained Partitioning for Aggressive Data Skipping [A]. Dyreson C E. Proceedings of the 2014 ACM SIGMOD international conference on Management of data [C]. New York: ACM, 2014: 1115–1126.
- [66] Sarwar B, Karypis G, Konstan J and Riedl J. Item-Based Collaborative Filtering Recommendation Algorithms [A]. Shen V Y. Proceedings of the 10th international conference on World Wide Web [C]. New York: ACM, 2001: 285–295.
- [67] Verykios V S, Elmagarmid A K and Houstis E N. Automating the Approximate Record-Matching Process [J]. Information sciences, 2000, 126(1): 83–98.
- [68] Yang Q, Li Z, Jiang J, Zhao P, Liu G, Liu A and Zhu J. NokeaRM: Employing Non-Key Attributes in Record Matching [A]. Li J. International Conference on Web-Age Information Management [C]. Berlin: Springer, 2015: 438–442.
- [69] Villarreal S E G and Brena R F. Topic Mining Based on Graph Local Clustering [A]. Batyrshin I Z. Mexican International Conference on Artificial Intelligence [C]. Berlin: Springer, 2011: 201–212.

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