東南大學

毕业设计（论文）开题报告

院（系）计算机科学与工程学院

专　业　计算机科学与技术

设计（论文）题目　基于多知识库的表格实体链接系统

学 生 姓 名　严晟嘉　学号　09013119

开 题 日 期　2017 年 3 月

开 题 地 点　东南大学九龙湖校区计算机楼

指 导 教 师　李慧颖

毕业设计（论文）开题报告

论文题目	基于多知识库的表格实体链接系统

一、选题背景和意义：

1991 年 8 月 6 日 Tim Berners-Lee 向世人公布了万维网（World Wide Web，WWW）项目，这是因特网上的万维网公共服务首次亮相。在此之后的 20 多年里，万维网相关技术取得了飞速的发展。万维网的出现革命性地改变了人们的生活方式和习惯，它将信息和知识的检索从图书馆搬到了互联网上。如今，万维网已经成为各类信息资源的聚集地和仓库。然而，随着万维网的发展和壮大，越来越多的信息充斥在网络上。据 Google 统计，早在 2008 年，万维网上就存在超过 10000 亿个 Web 文档，而且这个数字还以每天几十亿的速度持续增长。如此庞大的 Web 资源，使得用户难以在短时间内获得自己所需的信息。与此同时，Web 上充斥着各种无结构化的，冗余的，甚至是错误的信息。而且，采用 HTML 或 XML 格式的无结构化、半结构化以及结构化的数据是计算机不能直接理解和处理的，它适合人阅读，但不适合计算机。

为此，万维网的创始人 Tim Berners-Lee 于 1998 年提出语义网 (Semantic Web) 的概念，旨在通过给万维网上的文档添加能够被计算机所理解的语义元数据（Meta Data），来使整个互联网成为一个通用的信息交换媒介。从万维网到语义网的转变，关键在于给现有万维网上的文档数据进行元数据标注。所谓的元数据指的是描述数据的数据，通常用来描述数据的属性信息。实体链接的目的在于将表格单元格中的字符串指称链接到知识库中它的参考实体。实体链接（Entity Linking，EL）的工作可以看作是对 Web 上的数据进行元数据标注。因此，实体链接是语义网技术必不可少的研究内容。

有数据指出，在 2008 年仅仅英文版的维基百科就有 141 亿张 HTML 表格，其中 1.54 亿张表格包含了高质量的关系数据。因此，对 Web 上的表格进行实体链接，并挖掘出表格中的高质量关系数据可以在很大程度上丰富现有的语义数据，具有很大的研究意义和实用价值。目前，已有不少有关表格实体链接的研究，但大部分的表格实体链接都是在单个知识库中进行的。已经有研究者尝试进行多知识库的实体链接，但是，作者仅仅是用多个知识库来扩充候选实体集合，以在一定程度上解决单个知识库实体缺失的问题。而实际上，利用不同知识库的链接结果，以及知识库之间的信息可以提高单个知识库的实体链接质量，修正错误的实体链接结果。

此外，实体链接在计算机自然语言处理（Natural Language Processing，NLP）领域中也发挥着重要作用。它既是自然语言处理的关键所在，也是瓶颈所在。比如，语义搜索系统，自然语言问答系统，情感分析系统，推荐系统等都直接依赖于实体链接的效果。而且，在自然语言处理领域，有一个和实体链接非常相似的概念，叫词义消歧（Word Sense Disambiguation，WSD），它是自然语言处理相关技术的基础，同时也是难点。

因此，无论是在语义网技术还是在自然语言处理领域，也不论是在研究领域还是在工程领域，表格实体链接都起着重要的作用，具有极高的研究意义和实用价值。

二、课题关键问题及难点：

关键问题：

问题一：为什么我们的实体链接系统基于表格数据来做？
因为 Web 表格中是关系型数据的一个很有价值的来源。在 2008 年，Web 上就已经包含大约 1.54 亿张关系型数据的 HTML 表格，仅 Wikipedia 就包含 160 万高质量的表格。对 Web 上的表格进行实体链接，并挖掘出表格中的高质量关系数据可以在很大程度上丰富现有的语义数据，具有很大的研究意义和实用价值。抽取 Web 表格中的语义以生产机器可以理解的知识已经成为了一个热门的研究领域。

问题二：为什么利用了多知识库？
目前，已有不少有关表格实体链接的研究，但大部分的表格实体链接都是在单个知识库中进行的。但是单知识库存在很多实体缺失的情况，对实体的覆盖范围有限。实际上，利用不同知识库的链接结果，以及知识库之间的信息可以提高单个知识库的实体链接质量，修正错误的实体链接结果。

难点：

* 难点一：一词多义
因为自然语言的歧义性，使得同一字符串名称在不同的上下文中可能代表知识库中不同的实体。比如，对于字符串名称“苹果”，如果出现在上下文“相比之下，更多人喜欢吃苹果，而不是梨。他们认为苹果的营养价值大于梨”，那么它表示的是水果类实体苹果。如果它出现在“中关村的苹果产品比华强北的便宜”，那么，它代表是电子产品类实体苹果。实体链接需要提取出能够代表上下文的特征信息，为这些字符串名称进行消除歧义操作。即，需要确定某一个字符串名称具体指代了知识库中的哪一个实体。

* 难点二：实体缺失
由于知识库不可能覆盖现实生活中所有的实体，所以，不可能为每一个字符串名称都在知识库中为它找到对应的实体。实体链接技术需要解决这种实体缺失的问题。在现有的研究中，研究者直接将没有找到知识库实体的字符串名称标记一个 NIL 标签，表示该字符串名称不代表任何的实体。往往单知识库存在很多实体缺失的情况，对实体的覆盖范围有限，这也是我们采用多知识库来做实体链接的原因。一方面，利用多知识库中的丰富的实体来扩充候选实体集合；另一方面，利用多知识库之间的 sameAs 关系来提升实体链接的质量。

* 难点三：实体消岐
实体链接任务通常分为三步：指称识别，候选实体生成，实体消岐。其中实体消岐是最为关键的一步，因为它直接影响了链接的准确率。实体消岐就是对于一个潜在的指称，基于其上下文在它的候选实体集合中挑选出一个实体作为指称的参考实体。如何快速且准确的挑选出参考实体是我们必须思考的问题。

注：开题报告可单独装订，但在院（系）范围内，封面和装订格式必须统一。

三、文献综述（或调研报告）：（2000 字以上）

在哲学范畴内，实体指的是能够独立存在的、作为一切属性的基础和万物本源的事物。在语义万维网领域中可以将实体理解为相互区别的且独立存在的某种事物，它并不一定是物质上的存在。比如，“乔治•华盛顿”，“三星集团”，“中国北京”，“水果”都是语义万维网中的实体。随着自然语言处理技术的发展，以及语义万维网的兴起，实体这一概念变得越来越重要。相应的，作为自然语言处理，语义万维网知识库构建中的关键技术，实体链接也受到越来越多的关注和研究。

然而，自然语言在各个层次（比如短语，句子，语法等层次）上广泛存在的歧义性，给实体链接带来了一些困难和挑战。比如，在自然语言中，“苹果”一词既可以表示一种水果（苹果食物），也可表示一个公司（苹果公司），甚至一个电影（苹果电影）。同样对于知识库中的实体上海交通大学，在自然语言中可以被表示成“上海交通大学”，“上海交大”，“交大”，“SJTU”等字符串。此外，任何一个知识库都不能够覆盖世界上所有的实体。所以，实体链接还面临的一个挑战——实体的缺失。即，对于现实世界中的一些实体，在知识库中并不能找到与其相对应的实体。比如，对于一些新生事物，由于知识库的更新滞后，使得这些新生事物并不能及时的包含在现有的知识库中。

实体链接技术就是为了解决这样的问题而产生的。它根据待链接字符串的特征信息，从知识库中选择一个最合适的实体分配给该字符串。在实际的应用中，实体链接只是作为一个核心的子模块来为应用提供支持。所以，不论在工业界还是学术界，很少有专门的实体链接系统，甚至工具，更多的是作为系统的一个子模块出现。但是，对实体链接的研究却有很多。大致可分为非结构化数据的实体链接和结构化数据的实体链接两种。

文献综述首先介绍实体链接的基本定义，实体链接的过程，实体链接所需要解决的问题。然后介绍非结构化数据的实体链接方案，以及结构化数据的实体链接方案。最后，对这些实体链接方案进行对比和分析。

1 实体链接

1.1 实体链接的定义

实体链接就是将来自自然语言文本中的可能代表某一实体的字符串链接到知识库中与之相对应的实体上。

下面给出实体链接中的 3 个主要的概念：

* 概念 1：名称（Name Mention）表示来自自然语言文本的可能代表知识库中某一实体的字符串。这里的自然语言文本包括非结构化的自由文本，半结构化的列表内容，以及结构化的表格内容。

* 概念 2：实体（Entity）表示来自知识库中的一个实体。知识库中的实体是相互区别的，向下，实体拥有自己的属性，向上，实体属于某一类别。

* 概念 3：链接（Linking）表示自然语言字符串到知识库实体的一个映射过程。也被称之为实体消歧。用 f(n, e)表示，其中 f 是映射函数。n 是名称，即来自自然语言中可能代表知识库

某一实体的字符串，n 作为函数的输入。e 是知识库中与名称 n 相匹配的实体，是函数的输出。

1.2 实体链接的过程

由于自然语言的歧义性，使得一个字符串可表示多个知识库实体（比如字符串“苹果”）。同样，一个知识库实体可以被多个字符串表示（比如知识库实体上海交通大学）。因此，实体链接并不仅仅是一个检索的过程[1]。更精确的说，实体链接是一个消歧的过程。整个实体链接过程可分为如下 3 个主要的子模块。

* 模块 1：文本预处理
目的是为了从文本中得到所有可能代表知识库实体的字符串名称。对于非结构化数据，首先进行分词处理，然后进行词性标注，最后提取出所有的名词作为字符串名称。对于半结构化，以及结构化的数据，直接将属性值或单元格内容抽取出来作为字符串名称。在抽取的同时，过滤掉长段文本内容。

* 模块 2：候选实体生成
为每一个字符串名称，检索知识库。比如，将知识库中和字符串名称完全匹配的实体，被字符串名称包含的实体，包含字符串名称的实体，以及和字符串名称满足一定相似度值的实体全部抽取出来，形成候选实体集合。

* 模块 3：实体消歧
从候选实体集合中选择一个最合适的知识库实体，分配给字符串名称。目前主要的消歧方法有基于概率统计的方法，基于机器学习的方法，以及基于上下文语义相似度联合消歧方法。

实体链接的主要过程如图 1 所示：

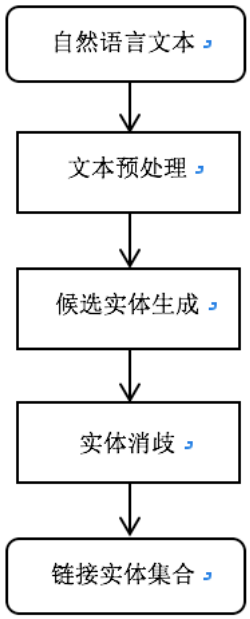

图 1 实体链接过程

1.3 实体链接需要解决的问题

实体链接有 2 个主要的问题需要解决。

问题 1：一词多义

因为自然语言的歧义性，使得同一字符串名称在不同的上下文中可能代表知识库中不同的实体。比如，对于字符串名称“苹果”，如果出现在上下文“相比之下，更多人喜欢吃苹果，而不是梨。他们认为苹果的营养价值大于梨”，那么它表示的是水果类实体苹果。如果它出现在“中关村的苹果产品比华强北的便宜”，那么，它代表是电子产品类实体苹果。实体链接需要提取出能够代表上下文的特征信息，为这些字符串名称进行去歧义操作。即，需要确定某一个字符串名称具体指代了知识库中的哪一个实体。

问题 2：实体缺失

由于知识库不可能覆盖现实生活中所有的实体，所以，不可能为每一个字符串名称都在知识库中为它找到对应的实体。实体链接技术需要解决这种实体缺失的问题。在现有的研究中，研究者直接将没有找到知识库实体的字符串名称标记一个 NIL 标签，表示该字符串名称不代表任何的实体。比如文献[2]，文献[3]，文献[4]。本文并不对实体缺失作相关介绍。

2 实体链接方案

2.1 非结构化数据的实体链接方案

非结构化的数据指的是来自自然语言的自由文本段。非结构化数据的实体链接主要有基于概率统计和基于上下文语义相似度两种。

2.1.1 基于概率统计的实体链接

基于概率统计的实体链接，指的是根据已有的样本训练集统计出某一字符串名称在特定的上下文中指向知识库中某一特定实体的概率。文献[1]提出了一种基于概率的实体链接模型，它先计算出候选实体 e 出现在某一页面中的概率，实体 e 被某一特定的字符串名称表示的概率，以及实体 e 出现在特定的上下文文本中的概率。然后将这 3 者相乘，得到实体 e 和字符串名称 n 之间的相似度评分值。选择相似度评分值最大的实体 e 作为字符串名称 n 的链接实体。

样本训练集是从维基百科站点提供的 dump 语料库中抽取得来。将 dump 中超链接的锚文本，超链接所指页面的标题，以及超链接前后各 50 个字符抽取出来，就形成了样本空间。其中锚文本就是知识库实体对应的字符串名称，页面标题就是知识库（特指 DBpedia 知识库）实体，而超链接前后 50 个字符组成的文本就是该实体的上下文。

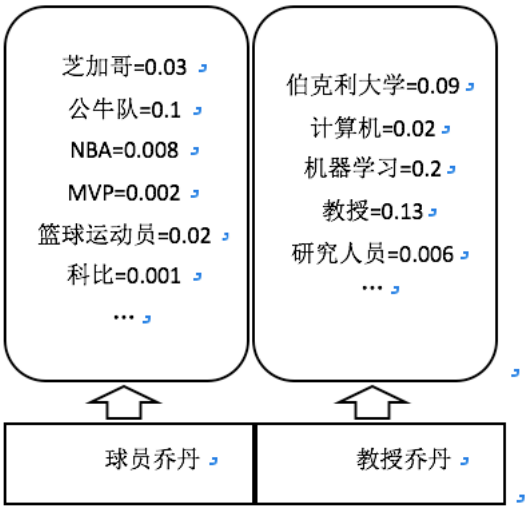

图 2 篮球运动员迈克尔·乔丹实体以及计算机教授迈克尔·乔丹实体的上下文

实体的上下文经过分词之后，表示成词向量。比如，对于知识库实体球员迈克尔•乔丹，它有一个上下文“芝加哥公牛队的迈克尔•乔丹再一次获得了 NBA 最有价值球员称号”，经过分词处理后，被表示成（芝加哥，公牛队，的，再，一次，获得，了，NBA，最，有价值，球员，称号）。然后，统计 dump 中所有出现球员迈克尔•乔丹的上下文，就可以得到球员迈克尔•乔丹的上下文词向量以及词频。因此，可以得到每一个实体的在维基百科中的上下文。比如对于篮球运动员“迈克尔•乔丹 ”以及伯克利大学教授“迈克尔•乔丹 ”，我们可以得到图 2 所示的上下文信息。

2.1.2 基于上下文语义相似度的实体链接

文献[5]给出了另外一种类型的文本实体链接。它在文献[1]的基础上考虑了不同实体的语义相似度。基于语义相似度的实体链接方法基于这样的一个观测：出现在同一上下文或者同一页面的命名实体所代表的知识库实体往往存在语义层面的相关性。而这些实体间语义相关性可以辅助实体链接。 文献[5]充分考虑不同实体对链接的相互影响，它将这些来自文本段的字符串名称以及来自知识库中的候选实体一同构建出一张称为 Referent Graph 的图。然后将实体链接的过程模拟为图的随机游走模型。通过协同推演算法（Collective Inference Algorithm）将实体间的相互影响传递给彼此。最后，保留那些彼此语义最相似的实体集作为字符串名称集合的链接实体。实体间的语义相似度由谷歌公式计算得到。

比如对于非结构化数据“在公牛期间，乔丹出演了电影大灌篮”，构建出来的 Referent Graph 如图 3 所示。

在 Referent Graph 之上，使用协同推演算法进行消歧。最终可得到的结果是，字符串名称“公牛”，“乔丹”，“大灌篮”分别指代知识库实体公牛队，乔丹（球星），宇宙大灌篮。

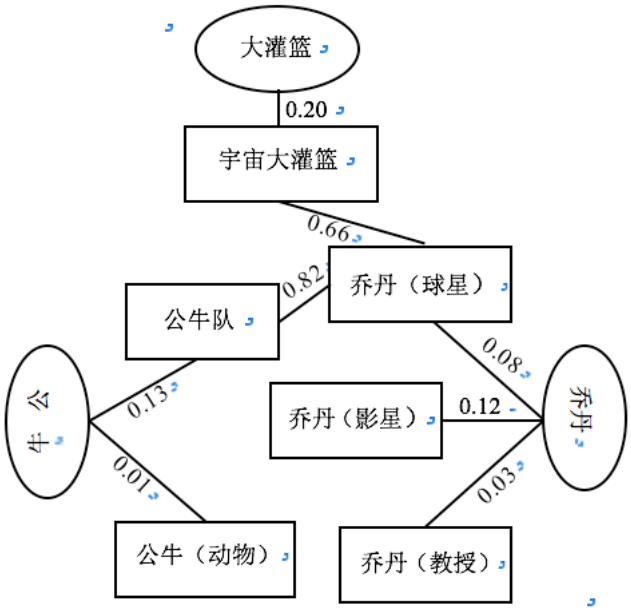

图 3 Referent Graph

2.2 结构化数据的实体链接方案

结构化数据（半结构化数据和结构化数据类似，这里统一用结构化数据表示）指的是诸如网页列表和表格数据。文献[4]指出，在 2008 年 Google 抓取的页面中就含有 141 亿个 HTML 表格，其中有 1.54 亿个表格含有高质量的关系数据。但是这些表格中的数据并没有被充分的利用起来。而挖掘出这些潜在的关系，需要有一个高质量的实体链接作为基础。现有的结构化数据的实体链接主要有基于上下文语义相似度以及基于机器学习的方法。

2.2.1 基于上下文语义相似度的实体链接

文献[6]将表格的列标注，表格实体链接，列之间的语义关系抽取联合起来以挖掘出表格中潜在的各种语义关系。在实体链接过程中，根据字符串相似度，从知识库中为表格中的表头以及表格单元格内容生成候选实体集合。接着，对每一个表头以及表格单元格内容从候选实体中分配一个初始的实体。然后计算这些初始的实体之间的语义相似度，如果存在语义不一致，则更改初始分配的实体。迭代多次，直到整个实体分配结果的语义信息收敛为止。

比如，对于一个 3 行 3 列的表格，可以将消歧过程模拟为图 4 所示模型， 其中椭圆节点代表的是为表头以及表格单元格内容分配的实体。而方框节点则负责计算与之相连接的实体的语义一致性。

2.2.2 基于机器学习的实体链接

文献[4]也是将表格的列类型标注，表格实体链接，列之间的语义关系抽取联合起来以挖掘出表格中潜在的各种语义关系。在做实体链接时，它根据表格实体所在的列的类型限定了候选实体集的范围，从而控制候选集合的大小。然后将表格实体和候选实体的编辑距离，Dice 分值，以及实体页面的 Page Rank 值，实体页面的长度构成一个特征向量。最后利用 SVM 学习算法为表格实体选择一个最合适的实体。

文献[7]研究的内容也是通过表格列标注，表格实体链接和列之间的语义关系抽取来挖掘出表格中潜在的语义关系。相比于文献[4]，这里的表格实体链接相对简单，它简单地计算了表格实体和实体的相似度，然后综合了不同的相似度计算公式，比如经过 TF-IDF 化的余弦公式，Jaccard 公式，再通过机器学习的方法设置各个相似度公式的权值。

除了上面两种常用的方法（基于语义相似度以及机器学习方法）之外，文献[8]提出了一种很简单的实体连接方案。文献[8]也是对表格关系抽取的一个研究，其中涉及到表格的实体链接。它将表格实体的链接直接转化为 url 的映射。因为文献[]处理的对象是表格中单元格内容为超链接的文本，并且这些表格都是来自维基百科站点的。能这么处理的原因是：维基百科站点的每一个超链接文本都对应于一个页面，也就是一个实体。而这些实体在 DBpedia 中都有唯一的 uri 标识。DBpedia 中 uri 和文本超链接的 url 只是前缀不同而已。所以简单的 url 转换就可以完成表格实体到知识库实体的一个链接。

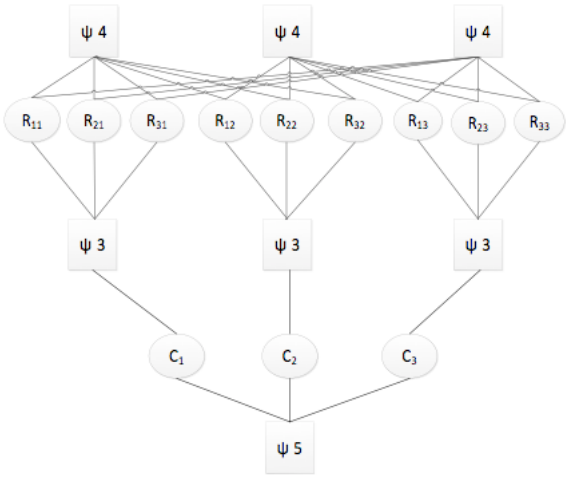

图 4　表格实体消歧模型图

3 实体链接方案对比分析

3.1 非结构化与结构化实体链接分析比较

相同点 1：过程相同
无论是非结构化数据，还是结构化数据，其实体链接的过程都是文本预处理，候选实体生成，实体消歧。

相同点 2：方法相似
实体链接的核心都是实体消歧，都是借助可用的上下文特征信息来进行消歧。

相异点 1：预处理不一样
对于非结构化的数据，预处理需要完成分词，词性标注操作。而对于结构化的数据，预处理操作只需简单的提取单元格内的内容，然后过滤掉类似于“解释”，“说明”性的长文本就行。

相异点 2：上下文特征不一样
非结构化数据的上下文特征信息主要有，字符串名称所在文本的前后若干字符，所在文本的

主题，以及同一文本中的其他字符串名称等。而结构化数据的上下文特征主要有表名，表说明，表头，表格中同一行文本，表格中同一列文本。

相对于结构化数据，非结构化数据的上下文特征更丰富，易寻找。但是，非结构化数据的上下文特征提供的信息一般比非结构化数据有价值。因为，表格中的同一行一般用来描述同一个事物。而且，同一列的数据一般都来自知识库同一个类别，属于同一个知识库概念。

3.2 不同特征对实体链接的作用分析与比较

在文献[1]中，实体链接用到了 3 个特征：实体流行度（实体出现在维基百科站点的概率），实体表示（实体被哪些字符串名称表示，以及被这些字符串名称表示的概率），上下文相似度（字符串名称的上下文和实体的上下文的相似度）。文献通过融合 3 个特征得到字符串名称所对应的知识库实体。文献[]的实验结果显示，不同的特征对实体链接的贡献大小不一。其中，上下文相似度对实体链接的影响最大，其次是实体流行度，最后是实体表示。

与此相类似，文献[4]，文献[7]指出，实体消歧中，其上下文语义相似性特征对实体链接的贡献最大。因此，无论是非结构化数据还是结构化数据的实体消歧，上下文信息都是非常重要的特征。

4 小结

随着语义万维网技术的兴起与发展，以及自然语言处理技术的日益成熟。语义的挖掘，知识库的构建正如火如荼的进行。实体链接作为语义挖掘与发现的一个重要步骤，正不断受到学术界和工业界的关注和研究。本文详细介绍了实体链接的基本概念，实体链接需要处理的问题，以及通用的实体链接方案，并对这些方案给出了分析和对比。

参考文献

[1] Han X, Sun L. A Generative Entity-Mention Model for Linking Entities with Knowledge Base[C]. In：Proceedings of the 49th Annual Meeting of the ACL, 2011, 945-954.

[2] Wu C, Wang H, Qu J, Yu Y. ZhishiLink: Entity Linking on Zhishi.me[C]. In: Proceeding of the 7th CSWS, 2013, 161-174.

[3] Dredze M, McNamee P, Rao D, Gerber A, Finin T. Entity Disambiguation for Knowledge Base Population[C]. In：Proceedings of the 23rd International Conference on Computational Linguistics, 2010, 277-285.

[4] Mulwad V, Finin T, Syed Z,Joshi A. Using Linked Data to Interpret Tables[C]. In: Proceeding of the 1th Inernational Workshop COLD, 2010.

[5] Han X, Sun L, Zhao J. Collective Entity Linking in Web Text: A Graph-Based Method[C]. In：Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2011, 765-774.

[6] Mulwad V, Finin T, & Joshi A. Semantic Message Passing for Generating Linked Data from Tables[M]. The Semantic Web–ISWC 2013. Springer Berlin Heidelberg, 2013: 363-378.

[7] Limaye G, Sarawagi S, Chakrabarti S. Annotating and Searching Web Tables Using Entities, Types and Relationships[C]. In：Proceedings of the VLDB Endowment, 2010, 3(1-2): 1338-1347.

[8] Muñoz E, Hogan A, Mileo A. Using Linked Data to Mine RDF from Wikipedia's Tables[C]. In：Proceedings of the 7th ACM International Conference on Web Search and Data Mining, 2014, 533-542.

[9] 1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a web of open data. In: ISWC, pp. 722–735 (2007)

[10] 2. Bhagavatula, C.S., Noraset, T., Downey, D.: Tabel: Entity linking in web tables. In: ISWC, pp. 425–441 (2015)

[11] Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hell- mann, S.: Dbpedia-a crystallization point for the web of data. Web Semantics: science, services and agents on the world wide web 7(3), 154–165 (2009)

[12] Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collabo- ratively created graph database for structuring human knowledge. In: SIGMOD, pp. 1247–1250 (2008)

[13] Brin,S.,Page,L.:Reprintof:Theanatomyofalarge-scalehypertextualwebsearch engine. Computer networks 56(18), 3825–3833 (2012)

[14] Cafarella, M.J., Halevy, A., Wang, D.Z., Wu, E., Zhang, Y.: Webtables: exploring the power of tables on the web. PVLDB 1(1), 538–549 (2008)

[15] Craswell, N.: Mean reciprocal rank. In: Encyclopedia of Database Systems, pp. 1703–1703. Springer (2009)

[16] Ferna´ndez-Delgado,M.,Cernadas,E.,Barro,S.,Amorim,D.:Doweneedhundreds of classifiers to solve real world classification problems? The Journal of Machine Learning Research 15(1), 3133–3181 (2014)

[17] Hignette, G., Buche, P., Dibie-Barth´elemy, J., Haemmerl´e, O.: Fuzzy annotation of web data tables driven by a domain ontology. In: ESWC, pp. 638–653 (2009)

[18] Limaye, G., Sarawagi, S., Chakrabarti, S.: Annotating and searching web tables using entities, types and relationships. PVLDB 3(1-2), 1338–1347 (2010)

[19] Mulwad, V., Finin, T., Joshi, A.: Semantic message passing for generating linked data from tables. In: ISWC, pp. 363–378 (2013)

[20] Mun˜oz, E., Hogan, A., Mileo, A.: Using linked data to mine rdf from wikipedia's tables. In: WSDM, pp. 533–542 (2014)

[21] Navigli, R., Ponzetto, S.P.: Babelnet: Building a very large multilingual semantic network. In: ACL, pp. 216–225 (2010)

[22] Niu, X., Sun, X., Wang, H., Rong, S., Qi, G., Yu, Y.: Zhishi. me-weaving chinese linking open data. In: ISWC, pp. 205–220 (2011)

[23] Pereira, B.: Entity linking with multiple knowledge bases: An ontology modular- ization approach. In: ISWC, pp. 513–520 (2014)

[24] Shen, W., Wang, J., Luo, P., Wang, M.: Liege:: link entities in web lists with knowledge base. In: SIGKDD, pp. 1424–1432 (2012)

[25] Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: WWW, pp. 697–706 (2007)

[26] Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: A large ontology from wikipedia and wordnet. Web Semantics: Science, Services and Agents on the World Wide Web 6(3), 203–217 (2008)

[27] Syed, Z., Finin, T., Mulwad, V., Joshi, A.: Exploiting a web of semantic data for interpreting tables. In: WebSci, vol. 5 (2010)

[28] Venetis, P., Halevy, A., Madhavan, J., Pa¸sca, M., Shen, W., Wu, F., Miao, G., Wu, C.: Recovering semantics of tables on the web. PVLDB 4(9), 528–538 (2011)

[29] Zhang, Z.: Learning with partial data for semantic table interpretation. In: EKAW, pp. 607–618 (2014)

[30] Zhang, Z.: Towards efficient and effective semantic table interpretation. In: ISWC, pp. 487–502 (2014)

三、方案（设计方案、或研究方案、研制方案）论证：（1500 字）

对于我的毕设《基于多知识库的实体链接系统》，目前我有两种设计方案。方案一是一个两步法，先用各个单知识库做实体链接，然后再用多知识库中的 sameAs 关系来提升链接质量。方案二融合了方案一中的两步，直接在一个图模型上使用随机游走算法来做实体链接，并且在实体消岐时引入了更多的特征，除此之外，系统还增加了更多有意义的功能，比如 EL 和 sameAs 关系的迭代学习，在实体链接完成之后进行表格的关系抽取。

经过多方面的考量，最终选择了方案二。接下来我来具体讲下 2 个方案。

方案一：

对于 Web 表格中的实体链接，本方案主要分为 2 步：先用各个单知识库做实体链接，然后再用多知识库中的 sameAs 关系来提升链接质量。

1. 单知识库的实体链接

1.1 候选实体生成

对于每个知识库，生成表格中的 mention 的候选实体集合。先将表格中的 mention 分词得到单词集合，如果知识库中的 entity 或其同义词包含单词集合中的至少一个，将该 entity 加入候选实体集合。

1.2 实体消岐

分为三步。

a) 构建实体消岐图 (Entity Disambiguation Graph)

使用指称和它们的候选实体来作为图的节点。

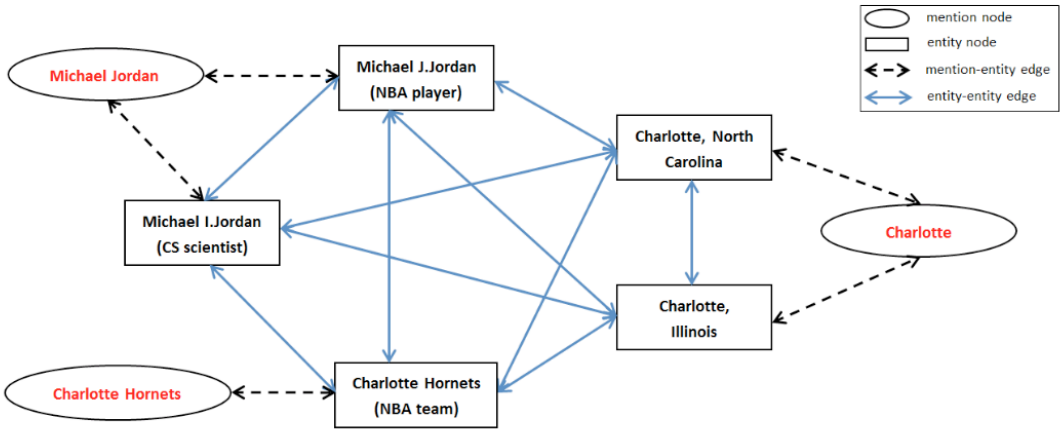

图 1. 一个构建好的实体消岐图的实例

实体消岐图包含两种类型的节点和两种类型的边：
* Mention Node: 这些节点代表表格单元格中的指称
* Entity Node: 这些节点代表在给定知识库中指称的候选实体
* Mention-Entity Edge: 指称与其候选实体之一之间的无向边
* Entity-Entity Edge: 实体之间的无向边

b) 计算 EL 影响因子 (EL Impact Factors)

在每个构建好的实体消岐图上，计算每个指称的初始重要性分值来联合消岐，然后计算不同节点之间的语义相似度作为 EL 影响因子来决定哪个候选实体是给定指称的参考实体。

对于给定 Web 表格的实体消岐图，每个节点和每条边都赋予了一个概率值。对于实体节点，其概率值代表其作为指称的参考实体的可能性。该概率初始化为 0，然后由 EL 影响因子影响。EL 影响因子，1) 指称节点的概率，可以看成指称对于联合消岐的重要性，因为我们平等地看待指称，所以在一个有 k 个指称的 Web 表格中，指称节点的概率为 1/k；2)边的概率，是节点之间的语义相似度。有 2 种边，指称-实体边和实体-实体边。

指称和实体之间的语义相似度:
* String Similarity Feature: 计算 mention 和 entity 的字符串相似度
* Mention-Entity Context Similarity Feature: 计算 mention 和 entity 的上下文相似度

实体和实体之间的语义相似度：
* Triple Relation Feature: 布尔值。如果 entity1 和 entity2 在同一个 RDF 三元组中，那么值为 1；否则为 0。
* Entity-Entity Context Similarity Feature: 计算 entity1 和 entity2 的上下文相似度

c) 迭代概率繁殖 (Iterative Probability Propagation)

在实体消岐图上跑随机游走算法，使用 EL 影响因子来进行迭代概率繁殖直到每个实体的概率分数收敛。每个实体都有一个概率分数来代表其是给定指称的参考实体的概率。

2. 用多知识库提升实体链接的质量

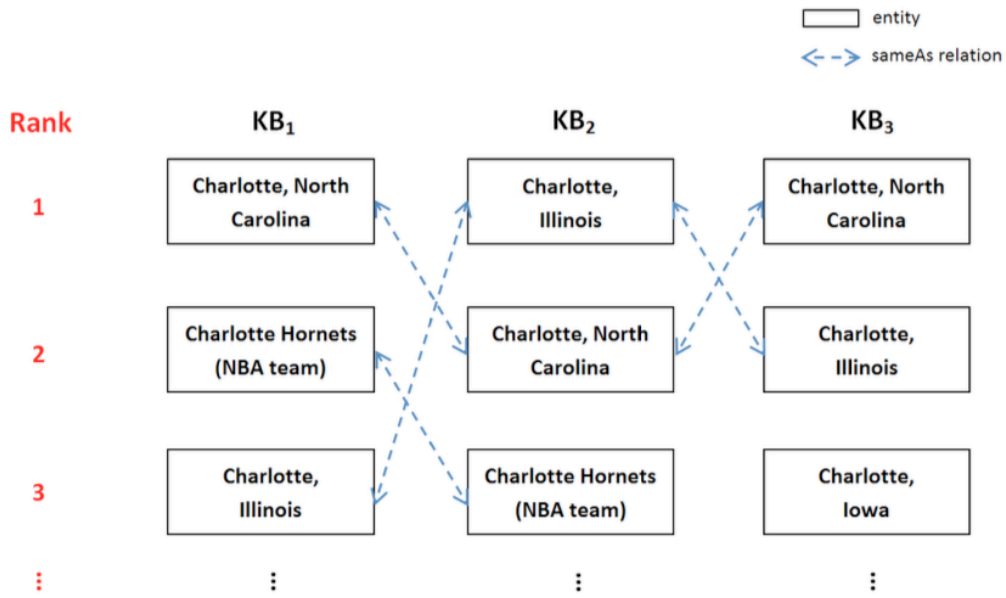

图 2. 对于指称"Charlotte"在不同知识库中的候选实体排名列表

两个实体之间的 sameAs 关系代表它们是等价的。根据对于给定指称的在不同知识库中的候选实体排名列表和实体之间的 sameAs 关系，将有 sameAs 关系的实体分组。然后通过 3 条启发式规则来重新对候选实体排名，得到指称的参考实体。

方案二：

方案二相对于方案一，具体做了以下改动：

(1) 修改了原先 EDG (Entity Disambiguation Graph) 中 entity node 的定义
将原先 EDG 中 entity node 转变为 entitySet node。原先的 entity node 只包含了一个 entity，而且 entity node 都来自同一个 KB。现在的 entitySet node 包含了一个或者多个 entity，利用多知识库间的 sameAs 关系，将有 sameAs 关系的 entity 放入同一个节点。这样我们就将多知识库的 sameAs 关系融合进 EDG 模型了。这种融合让我们有机会实现原先方法中两步的合并。

(2) EL 与 sameAs 的迭代学习
多知识库间的 sameAs 关系往往是不完备的，缺失的。我们利用多知识库间的 sameAs 来做 EL，所以不完备的 sameAs 关系会导致上面“改进一”中的 EDG 模型中很多 entitySet node 只有一个 entity，使融合步骤失去意义，变得似乎还是在和单知识库链接。而且因为缺失了 sameAs 关系，会导致链接的结果忽略了一些有价值的 entity，比如一个 entity e 在链接结果中排名靠后，但其实它和排名靠前的 entity 是等价的，然而因为缺失了它们之间的 sameAs 关系，所以最后会忽略 e。EL 又可以学习多知识库间的 sameAs 关系，如果 2 个 entity 都能被同一个 mention 链接到，那么它们就有可能是 sameAs 的。EL 和 sameAs 关系是相互促进的，因此让它们迭代学习既能补全多知识库间的 sameAs 关系，又能提升 EL 的质量。

EL 和 sameAs 迭代学习有 2 种不同的方案：
a) 我们可以将实体消岐的过程变成一个大的循环来迭代学习 EL 和 sameAs，当无法学到新的 sameAs 关系或者 EL 质量没有明显提升时终止循环。
b) 在实体消岐的最后一步“迭代概率繁殖”之后，在已得到的实体链接的结果上进行 EL 和 sameAs 的迭代学习，换句话说，就是在指称和新的实体集合之间迭代学习。当无法学到新的 sameAs 关系或者 EL 质量没有明显提升时终止循环。

(3) 在 EL 的同时进行关系抽取
关系抽取是表格 semantic interpretation 中的主要任务之一。关系抽取指的是抽取表格中列与列之间的关系。当我们拥有了表格 EL 的结果时，表格中每一列中的单元格链接的实体也就有了，识别两列中任意一行的链接实体的关系，它们的关系即为列间关系。

(4) 在计算 EL Impact Factors 时丰富了特征
原先基本上只使用了语义相似度（虽然把它细分成 2 点），现在在计算 EL Impact Factors 时加入了更多特征，来弥补原先单一特征可能导致的效果不佳的缺点，让特征的计算更加平衡。

接下来全面的讲述此时的方案二中的方法。

1. 指称识别
表格单元格中的字符串不全为指称，所以我们需要先把潜在的指称识别出来。可以采用 TabEL 中的方法，先认为单元格中最长的字符串是 tq，然后从头开始找 tq 中潜在的 mention s，s 的长度从 1 开始，每次递增 1，看 s 能不能在 KB 中找到对于的 entity，找得到的话此时的 s 就是一个 mention，然后在 tq 中去掉此时的 s，重复从头开始找的过程，循环往复，直到 tq 长度为 0。

2. 候选实体生成
对于每个知识库，生成表格中的 mention 的候选实体集合。先将表格中的 mention 分词得到单词集合，如果知识库中的 entity 或其同义词包含单词集合中的至少一个，将该 entity 加入候选实体集合。

3. 实体消岐(循环)
EL 最关键也最有挑战性的一步。为了让 EL 和 sameAs 能够迭代学习，将这步设计成了一个循环。

3.1 构建实体消岐图
根据多知识库间的 sameAs 关系，将来自不同知识库的等价的候选实体融合进同一个节点。

图的节点和边的类型：
* mention node
* entitySet node
    * 等价的 entity 组成的集合
    * 结构: {[e1, kb1], [e2, kb2], [e3, kb3], …, [en, kbn]}

* mention - entitySet edge　无向带权边
* entitySet - entitySet edge　无向带权边

潜在的问题：在缺乏大量 sameAs 关系下，链接结果可能只是一个知识库中的一个实体，跳过了多知识库的结果。

每个节点和边上都有一个概率值。
Probabilities:
* mention node probability: mention 的权重。权重均分就是 1/k
* entitySet node probability: 成为 mention 的候选实体集合的概率
* mention - entitySet edge probability: 二者之间的一系列特征计算得到
* entitySet - entitySet edge probability: 二者之间的一系列特征计算得到

3.2 计算 EL 影响因子

A set of features between mention and entitySet:
* String Similarity Feature: 计算 mention 和 entitySet 中每个 entity 的字符串相似度，最后取均值
* Mention-EntitySet Context Similarity Feature: 计算 mention 和 entitySet 中每个 entity 的上下文相似度，最后取均值
* String Equality Feature: 布尔值。如果存在 mention 和 entitySet 中的一个 entity 完全相同，那么值为 1；否则为 0
* Acronyms Feature: 布尔值。如果存在 mention 是 entitySet 中的一个 entity 的首字母缩略，那么值为 1；否则为 0
* Aliases Feature: 布尔值。如果存在 mention 和 entitySet 中的一个 entity 是别名关系，那么值为 1；否则为 0
* 同义词特征
* 列类型特征

A set of features between entitySet and entitySet:
* Triple Relation Feature: 布尔值。如果 entitySet1 中存在一个 entity 和 entitySet2 中的一个 entity 在同一个 RDF 三元组中，那么值为 1；否则为 0。
* EntitySet-EntitySet Context Similarity Feature: 计算 entitySet1 所有 entity 和 entitySet2 中所有 entity 的上下文相似度，最后取均值。O(n^2)
* String Equality Feature: 布尔值。如果存在 entitySet1 中的一个 entity 是 entitySet2 中的一个 entity 完全相同，那么值为 1；否则为 0
* Acronyms Feature: 布尔值。如果存在 entitySet1 中的一个 entity 是 entitySet2 中的一个 entity 的首字母缩略，那么值为 1；否则为 0
* Aliases Feature: 布尔值。如果存在 entitySet1 中的一个 entity 和 entitySet2 中的一个 entity 是别名关系，那么值为 1；否则为 0
* 同义词特征
* 列类型特征

3.3 迭代概率繁殖

在 EDG 上跑随机游走算法，直到收敛，得到 mention 链接到的的 entitySet node 排名。因为 entitySet node 中所有 entity 都是等价的，所以排名第一的 entitySet node 中的所有 entity 自然就是 mention 最终的链接结果。

4. 关系抽取

关系抽取指的是抽取表格中列与列之间的关系。当我们拥有了表格 EL 的结果时，表格中每一列中的单元格链接的实体也就有了，识别两列中任意一行的链接实体的关系，它们的关系即为列间关系。

相对于方案一，方案二有什么优势呢？

* EL 和 sameAs 迭代学习改善了多知识库间 sameAs 关系缺失的问题，同时也提升了链接质量，让我们的 EL 系统变成了一个可以自我学习的更智能的系统。
* 计算 EL Impact Factors 时特征增加，使得 EDG 上节点和边的概率值计算得更为合理，不会因为只使用了少量特征而使某些边上的概率偏高或偏低
* 将原来的两步方法融合为一步，简化了模型。。。
* 增加了关系抽取，增加了 EL 系统功能。

四、进度安排：

2017 年 2 月 15 日—— 2 月 27 日	阅读文献，翻译资料
2017 年 1 月 19 日—— 3 月 06 日	填写任务书，设计算法模型
2017 年 2 月 23 日—— 3 月 15 日	完成开题报告，开展实验
2017 年 4 月 20 日—— 4 月 30 日	填写中期检查表，做 EL 系统
2017 年 5 月 10 日—— 5 月 20 日	撰写论文，完善 EL 系统
2017 年 5 月 10 日—— 5 月 25 日	指导老师验收 EL 系统
2017 年 5 月 18 日—— 5 月 26 日	提交论文
2017 年 6 月	答辩

六、指导教师意见：

签名：　　年　　月　　日

七、开题审查小组意见：

签名：　　年　　月　　日