

SHANGHAI JIAO TONG UNIVERSITY

**学士学位论文**

## THESIS OF BACHELOR



论文题目：随机游走在实体链接以及图像匹配的应用

学生姓名: 张晟，姜德志

学生学号: 115037910055，X

专 业: 软件工程

指导教师: 沙朝峰

学院(系): 电子信息与电气工程

随机游走在实体链接以及图像匹配的应用

摘要

<随机游走概述>

<实体链接，为什么要研究以及算法概述>

<图像匹配，为什么要研究以及算法概述>

<本文贡献概述>

关键词：随机游走，实体链接，图像匹配

目 录

[THESIS OF BACHELOR 1](#_Toc453244603)

[第一章 技术背景综述 4](#_Toc453244604)

[1.1 随机游走阐述 4](#_Toc453244605)

[1.2 随机游走的计算 4](#_Toc453244606)

[1.3 随机游走在本文中的运用 4](#_Toc453244607)

[1.3.1 随机游走在实体链接的运用 4](#_Toc453244608)

[1.3.2 随机游走在图像匹配的运用 5](#_Toc453244609)

[第二章 实体链接简介 6](#_Toc453244610)

[2.1研究对象简介 6](#_Toc453244611)

[2.2 相关研究方法 6](#_Toc453244612)

[2.2.1 局部评价方法 6](#_Toc453244613)

[2.2.2 全局评价方法 6](#_Toc453244614)

[2.3 研究方法 6](#_Toc453244615)

[2.3.1研究方法概述 6](#_Toc453244616)

[2.3.1研究方法讨论 6](#_Toc453244617)

[2.4 主要贡献 6](#_Toc453244618)

[第三章 实体链接算法实现 7](#_Toc453244619)

[3.1 问题建模 7](#_Toc453244620)

[3.2 算法介绍 7](#_Toc453244621)

[3.2.1在线知识库的建立 7](#_Toc453244622)

[3.2.2具有歧义文字和候选实体的选取 7](#_Toc453244623)

[3.2.3算法流程 7](#_Toc453244624)

[3.2.4局部评价方法选取 7](#_Toc453244625)

[3.2.5全局评价方法选取 7](#_Toc453244626)

[3.3 算法讨论 7](#_Toc453244627)

[第四章 实体链接实验评估 8](#_Toc453244628)

[4.1 实验搭建 8](#_Toc453244629)

[4.1.1测试环境说明 8](#_Toc453244630)

[4.1.2测试数据选取 8](#_Toc453244631)

[4.1.3对比算法设置 8](#_Toc453244632)

[4.2 数据分析 8](#_Toc453244633)

[4.2.1算法的正确性检验 8](#_Toc453244634)

[4.2.2指代数量和时间的关系 8](#_Toc453244635)

[4.2.3实体选取深度和广度与时间的关系 8](#_Toc453244636)

[4.2.4实体选取深度和广度与正确度的关系 8](#_Toc453244637)

[4.2.5指代密度和正确度的关系 8](#_Toc453244638)

[4.2.6和其他算法的对比 8](#_Toc453244639)

[4.3 参数影响 8](#_Toc453244640)

[4.3.1 kl和dot算法对比 8](#_Toc453244641)

[4.3.2 lambda参数设置 8](#_Toc453244642)

[4.3.3 restart probability参数设置 8](#_Toc453244643)

[4.4 本章讨论 8](#_Toc453244644)

[第五章 实体链接算法讨论与优化 9](#_Toc453244645)

[5.1 存在问题与解决算法 9](#_Toc453244646)

[5.2 优化算法 9](#_Toc453244647)

[第六章 图像匹配综述 10](#_Toc453244648)

[6.1 研究对象 10](#_Toc453244649)

[6.2 研究方法 10](#_Toc453244650)

[6.3 主要成果 10](#_Toc453244651)

[第七章 图像匹配算法实现 11](#_Toc453244652)

[7.1 原理简介 11](#_Toc453244653)

[7.2 算法实现 11](#_Toc453244654)

[第八章 图像匹配实验评估 12](#_Toc453244655)

[8.1 实验搭建 12](#_Toc453244656)

[8.2 数据分析 12](#_Toc453244657)

[8.3 参数影响 12](#_Toc453244658)

[8.4 本章小结 12](#_Toc453244659)

[参考文献 13](#_Toc453244660)

# 第一章 技术背景综述

## 1.1 随机游走阐述

随机游走是一种数学上的随机过程，在二维的情况下它等同于维纳过程，可以用如下的数学等式去描述：

《<https://en.wikipedia.org/wiki/Wiener_process>》

其中，《》

随机游走的应用十分广泛，较为典型的应用是金融学中对于股票市场的模拟。金融学把股价的波动看作是一种随机游走，并在此基础上发展出了著名的Black-Scholes Formula。在物理上，对于分子热运动，布朗运动看作是随机游走的一种形式，同样的，它具有十分重要的运用。

在关于图的算法研究上，随机游走可以抽象为，按照一定概率去访问邻边点。根据它随机的特性，我们可以把稳态的概率分布看作是图结构抽象表示，它可以相对确定各点之间边的权重大小。

在图上的重启随机游走(Random Walk with Restart)，是一种特殊的随机游走，每一步，它都具有一定的概率返回的出发点，再进行后续随机游走。在数学上用矩阵表达为：

《FAST》

其中，《》。具有重启的随机游走增加了对于初始点的访问概率，从而也增加了对于初始点相连接节点的访问概率。如果说，我们可以把从某个节点出发到达另外一个点的概率，看做是这两个点的一种距离。从这种角度出发，就很好的引申出了，两个抽象点之间的距离的一种定义。这种定义极大的方便人们去表达抽象联系，page rank就是随机游走的一种很好的运用。

## 1.2 随机游走的计算

对于重启随机游走的计算，一般的我们运用矩阵运算。初始化矩阵每行每列代表对应点的标识，其中的值，代表对应标识的两点之间的边的权重，通过归一化（Laplace或者行列归一）我们可以方便的把它具有概率意义的矩阵。通过求逆矩阵，可以方便的得到所求的概率分布(r)。但是，当矩阵内所含点的数量过大，计算逆矩阵的耗时也就越大(呈现三次方关系)，存储也就越大(呈现二次方关系)，注意到，在一般的图中，所化成的矩阵为稀疏矩阵，我们可以运用以下的方法，去很好的近似求逆，达到缩减计算时间，缩减存储空间的目的。

《TODO 分割方法》

## 1.3 随机游走在本文中的运用

### 1.3.1 随机游走在实体链接(Entity linking)的运用

在本文中，我们将用这种抽象距离去表达，实体(entity)与文档(document)之间的联系程度。具体的，我们运用重启的随机游走方法，对一个文档中的实体集合进行签名，方法是以文档中不同实体所对应的重要程度对文档中的实体进行一个概率测度的赋值，形成一个代表不同文档中实体重要程度的向量，如上述所言，把此向量作为随机游走的初始向量，经过概率矩阵的转化，形成对此文档的签名。可以看到，这种以随机游走方法的形成的签名，能以一定的形式保存出文档实体之间关系。

### 1.3.2 随机游走在图像匹配的运用

# 第二章 实体链接简介

## 2.1研究对象简介

实体链接是一种通过搜索知识库(Knowledge base)，把文章中的文字映射到实体的方法，它主要用于确定文中具有歧义的指代（mention）所包含的意思。它具有一些十分典型的运用，比如通过去歧义，扩充知识库，还有在进行网络搜索时给出更准确的建议。实体链接可以被以下例子说明。

Example 1. Saban, previously a head coach of NFL’s Miami, is now coaching Crimson Tide. His achievements include leading LSU to the BCS National Championship once and Alabama three times.

Example 2. After his departure from Bualo, Saban returned to coach college football teams including Miami, Army and UCF.

在Example 1，2中，Crimson Tide和Alabama同样的指代一个足球队，即是the Alabama Crimson Tide of the University of Alabama。在另一个方面，Example 1中的Saban 和Example 2中的Saban所指代的对象不同，分别是Nick Saban在Example 1里和Lou Saban在Example 2里。论文希望所提出的算法能识别出这种文本中的指代，通过对知识库里的实体含义比较，把每个指代都映射到真正的实体含义上。

## 2.2 相关研究方法

实体链接的方法是要找出每个指代所对应的实体，现行的一般算法为，先对每一个指代找到可能的实体称为歧义候选（disambiguation candidates），主要是通过同义词词典，例如在Example中，Saban的歧义候选包括Nick Saban和Lou Saban。然后再对每一个歧义候选进行筛选。主要是通过以下两种方式，局部评价方法和全局评价方法。

### 2.2.1 局部评价方法

实体链接局部的评价方法一般是考虑每个指代周围的实体分布，做一个统计，当询问某个指代的意思时候，则挑选数据库中此时与它周围最匹配的分布所对应的实体意思。可以看出，这种方法需要大量的训练集合，当某个指代的歧义候选分布差距很大的时候，可能找不到在知识库中对应的数据。

### 2.2.2 全局评价方法

全局评价方法考虑的是，每个指代和实体的含义。含义可以表达为某个实体相对于其他实体的关系，可以抽象称为图，每个节点就是实体，图中的边代表的就是实体和实体之间有引用。通过以确定文章中的实体在图中的分布，就可以进一步选取某一个指代的所有歧义候选中，与图连接性最强的候选为此指代的真实实体。同样的，这种建立图的方式，计算量十分巨大，所以需要通过一定的技巧去缩减不相关实体信息。

## 2.3 论文研究方法

### 2.3.1论文研究方法概述

论文结合了两种方法，做出了综合评判。对于设定全局评价函数，文章《Robust Entity Linking via Random Walks》首先建立了实体签名的概念，通过重启的随机游走，确定实体在知识库图中的概率分布，为此实体对于所有实体的概率向量，再建立文档签名，即是对于重要实体在知识库中的概率分布，通过迭代的方式不断引入新的歧义候选，判别每个歧义候选的实体签名（向量）和文档签名（向量）的差别，选取较小的，作为指代的真实实体，并引入文档，扩充文档中的实体。而对于局部的评价函数，可以使用先验概率或者其他上下文敏感的测量方法。

### 2.3.1论文研究方法讨论

论文引入的随机游走算法，有如下优点，其一相比于局部评价，只观察实体与指代是否一起出现的0/1关系，概率分布式一种更细粒度的评价联系的方法，其二，随机游走能处理语义上的相关关系而不是结构上同时出现的关系，其三，那些与总体比较疏远的歧义候选，可以同样得到很高的评分如果它和已经确定的实体具有较高的联系。

## 2.5 本文与论文算法实现差异讨论

本文大部分采用论文《Robust Entity Linking via Random Walks》所提供的办法。但是在以下方面与论文有所差别。

1. 本文的实现语言是Python，与原论文不一致，可能导致算法在性能上有所差异。
2. 本文并没有采取离线的知识库，一是离线的知识库的节点数量过大，不减枝时进行矩阵运算耗时太长，载入速度十分漫长，二是本文针对在线数据设计算法，所以采取了网络Wikipedia作为知识库，在线数据库具有针对性强，实时性强，覆盖面强等特征，但相应的缺少实体联系信息。本文并在实验部分对知识库建立时的对于实体的取样进行了讨论，实验表明，减少节点能一定程度上减少不相干节点对算法的影响。
3. 论文中没有明确给出一些特殊的参数设置，如局部评价函数和全局评价函数的权重，和一些阈值设定，本文按照实验所测试结果比较优异的参数进行设计。

## 2.5 本文主要贡献

1）根据《Robust Entity Linking via Random Walks》论文内容实现了在线的实体链接功能模块，该模块寄存在github上。

2）对于实现的模块进行了效果测试，给出测试样例，给出数据结果与分析。

3）针对论文特定不足，给出了优化的建议。并提出了改进算法，YAEL(Yet Another Entity Linking)算法设计。

# 第三章 实体链接算法实现

## 3.1 问题建模

我们首先给出，实体链接的定义。

(Entity Linking). Given a set of mentions M = {m1,...,mn} in a document d, and a knowledge base KB whose entity set is E, the problem of entity linking is to find an assignment : M ! E [{NIL}.]

(m) = arg max ( (m, ei) + (ei, d))

其中，<>

## 3.2 算法介绍

### 3.2.1在线知识库的建立

<图片>

本文的知识库是建立在Wikipedia之上。采取了实时获取数据的网络爬虫方式。

根据Wikipedia词条结构的特征，本文设计其获取的方式。以下为爬虫算法简介。

Wikipedia对于大部分词条具有去歧义页(Disambiguation page)，每条歧义页中会有列出一些引用指向词条的文章(article)，并且在去歧义页中会含有主要词条，主要词条是在普通情况下的搜索词条所对应的文章。

另外，Wikipedia对于一些词条会进行跳转，这种跳转一般发生只有唯一确定词条解释时，我们把它当作无歧义词条(在算法中这些无歧义词条的确定是十分重要的，用以确定其他词条) 。

有此，我们方便的设计，把每个指代先去在去歧义页中获取所有的链接（如果没有本文尝试搜索原词，如果再无此词条信息，本文把其指代赋值为Nil），把这些链接指向的文章，作为它的歧义候选，并在每篇文章中获取其相关的链接，作为与其相关的实体。例如，文中的Apple可能指向苹果公司，指向一种水果，指向某个电影，我们把这些作为Apple指代的歧义候选，苹果公司又有许多引申出的相关概念链接，如iPhone，加州，科技公司等，我们把其作为与之相关的实体。这样我们就得到了一张关于所有指代的实体图。

### 3.2.2指代和实体的选取

本文所采用的摘取文章中指代策略十分简单，一是采用黑名单，去除一些常用词，A the等，二是设定白名单对于一些专有名词进行摘取，三是设定频率，每个多少字摘取一个。

歧义候选的选取，本文所采用的方法是设定深度，和广度的阈值。深度代表着，从某个指代开始最大跳转数，广度代表着某个实体所能获取的最大相关的引用数。如上文所提，Apple->Apple Inc->iPhone->Phone是一个深度为4的设定，而Apple:[Apple Inc. Apple Tree, Apple Corps]是一个广度为三的设定。

深度和广度的设定极大的影响着所构造的实体图的连接性。深度和广度越大，则图的复杂性越大，计算时间越长，所含信息越大，值得注意的是它所含的无用信息（噪声）也就越大，可能对算法的结果产生极影响。

### 3.2.3算法流程

<算法图>

1. 首先对于给定所有的指代，我们从Wikipedia上获取其的实体图。我们想要得到对每个指代所对应的在实体图里实体。
2. 我们确定那些只含有一个歧义候选的指代，把这些指代的值赋值为歧义候选所代表的实体。并把它加入此文档所包含的实体。
3. 如果不含有只有一个候选的指代，则把所有候选加入文档实体。
4. 我们对于指代进行排序，依据是其包含的歧义候选的个数，我们先从那些较小个数的候选处理。
5. 我们对于文档进行签名，签名的方法是根据其所含有的实体和其的重要程度和联系强度（tfidf和prior probability），得到重启随机游走的初始向量，在针对图的概率转移矩阵，进行运算，得到分布作为文档的签名。
6. 对于每一个实体也进行以上的签名。
7. 全局评价分数由这两个签名所对应的概率分布差所确定，论文使用Zero-KL Divergence函数。
8. 再与局部评价分数相加（本文使用的先验概率函数），最高得分大于阈值，则把它作为这个指代的实体，并把它加入文档。
9. 如果文档所包含的实体有变化重新计算文档签名，直到处理了所有指代。

### 3.2.4局部评价方法选取

本文采取的是先验概率模型，即

<公式>

### 3.2.5全局评价方法选取

随机游走的初始向量，有以下数学公式确定:

~v i = I(m)⇤P(ei,m)

其中，Zero-KL，由以下公式确定：

## 3.3 算法讨论

知识库获取的其他问题。Wikipedia在线数据的禁止，大小写

# 第四章 实体链接实验评估

## 4.1 实验搭建

### 4.1.1测试环境说明

### 4.1.2测试数据选取

### 4.1.3对比算法设置

## 4.2 数据分析

### 4.2.1算法的正确性检验

### 4.2.2指代数量和时间的关系

### 4.2.3实体选取深度和广度与时间的关系

### 4.2.4实体选取深度和广度与正确度的关系

### 4.2.5指代密度和正确度的关系

### 4.2.6和其他算法的对比

## 4.3 参数影响

### 4.3.1 kl和dot算法对比

### 4.3.2 lambda参数设置

### 4.3.3 restart probability参数设置

## 4.4 本章讨论

# 第五章 实体链接算法讨论与优化

## 5.1 存在问题与解决算法

运算量极大－我们争辩，事实上，在一般情况下只需要3层深度就可以辨别

选取那些不具有歧义的实体太少－增加一些情况

缺少局部打分函数与全局打分函数的统一－统一两个函数，我们可以看作是不断缓慢引入新信息的过程。

缺少信息的局部信－我们知道用map reduce和iterative的方法不断处理相邻的数据

## 5.2 优化算法

《第一步，选取相邻的三个句子，为一个信息block，对每个block确定，不带有歧义的实体，（通过wiki的跳转，或者wiki的建议），把这个实体所具有的概念实体列入这个block概念集合，每个概念实体具有初始分数。通过给剩余带有歧义的指代每个歧义进行概念分析，选取非零的点积除以概念数最大的那个歧义实体，把它列入概念集合，更新概念集合的分数，更新方式如下，如果刚刚匹配成功的概念实体＋1分，把1/2分数留给自己，剩下的分数分给与它相邻的节点，依次下去，直到分的分数小于特定值。重复这个过程直到，所有剩余的歧义实体与它们并不相交。然后合并两个相临的block成为一个新的block，每个block中间实体的分数归一化，重复选取刚刚没有复制的指代。直到合并成完整的指代。如果还剩余未复制的指代，去除所有以复制的指代，形成新的指代，按照刚刚步骤再来一次，直到声誉为复制指代与上次一样。把它们复制为nil》

# 第六章 图像匹配综述

## 6.1 研究对象

## 6.2 研究方法

## 6.3 主要成果

## 

# 第七章 图像匹配算法实现

## 7.1 原理简介

## 7.2 算法实现

# 第八章 图像匹配实验评估

## 8.1 实验搭建

## 8.2 数据分析

## 8.3 参数影响

## 8.4 本章小结

# 参考文献

1. Buyya, Rajkumar; Chee Shin Yeo, Srikumar Venugopal. Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities. Department of Computer Science and Software Engineering, The University of Melbourne, Australia. 9. [2008-07-31]
2. White T. Hadoop: The definitive guide[M]. " O'Reilly Media, Inc.", 2012.
3. Li S, Hu S, Wang S, et al. WOHA: Deadline-aware Map-Reduce workflow scheduling framework over Hadoop clusters[C]//Distributed Computing Systems (ICDCS), 2014 IEEE 34th International Conference on. IEEE, 2014: 93-103.
4. Kc K, Anyanwu K. Scheduling hadoop jobs to meet deadlines[C]//Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on. IEEE, 2010: 388-392.
5. Tran T T, Zhang P Y, Li H, et al. Resource-Aware Scheduling for Data Centers with Heterogenous Servers[J].
6. Grandl, Robert, et al. "Multi-resource packing for cluster schedulers." Proceedings of the 2014 ACM conference on SIGCOMM. ACM, 2014.
7. Sandholm T, Lai K. Dynamic proportional share scheduling in hadoop[C]//Job scheduling strategies for parallel processing. Springer Berlin Heidelberg, 2010: 110-131.
8. Rao B T, Reddy L S S. Survey on improved scheduling in Hadoop MapReduce in cloud environments[J]. arXiv preprint arXiv:1207.0780, 2012.
9. Zaharia, Matei, et al. "Improving MapReduce Performance in Heterogeneous Environments." OSDI. Vol. 8. No. 4. 2008.
10. Vavilapalli V K, Murthy A C, Douglas C, et al. Apache hadoop yarn: Yet another resource negotiator[C]//Proceedings of the 4th annual Symposium on Cloud Computing. ACM, 2013: 5.
11. Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters[J]. Communications of the ACM, 2008, 51(1): 107-113.
12. Dean J, Ghemawat S. MapReduce: a flexible data processing tool[J]. Communications of the ACM, 2010, 53(1): 72-77.
13. Cook S A. The complexity of theorem-proving procedures[C]//Proceedings of the third annual ACM symposium on Theory of computing. ACM, 1971: 151-158.
14. LaMacchia B A. Basis reduction algorithms and subset sum problems[J]. 1991.
15. Yenisey M M, Yagmahan B. Multi-objective permutation flow shop scheduling problem: Literature review, classification and current trends[J]. Omega, 2014, 45: 119-135.
16. Nieuwenhuis R, Oliveras A, Tinelli C. Solving SAT and SAT Modulo Theories: From an abstract Davis--Putnam--Logemann--Loveland procedure to DPLL (T)[J]. Journal of the ACM (JACM), 2006, 53(6): 937-977.
17. Goldberg E, Novikov Y. BerkMin: A fast and robust SAT-solver[J]. Discrete Applied Mathematics, 2007, 155(12): 1549-1561.
18. Lai K, Goemans M. The knapsack problem and fully polynomial time approximation schemes (FPTAS)[J]. Retrieved November, 2006, 3: 2012.
19. Cheng T C E, Ding Q. The time dependent machine makespan problem is strongly NP-complete[J]. Computers & operations research, 1999, 26(8): 749-754.
20. Whitley D. A genetic algorithm tutorial[J]. Statistics and computing, 1994, 4(2): 65-85.