

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](#_Toc453019084)

[第一章 技术背景综述 3](#_Toc453019085)

[1.1 随机游走阐述 3](#_Toc453019086)

[1.2 随机游走的计算 4](#_Toc453019087)

[1.3 随机游走在本文中的运用 4](#_Toc453019088)

[第二章 实体链接简介 5](#_Toc453019089)

[2.1研究对象简介 5](#_Toc453019090)

[2.2 相关研究方法 5](#_Toc453019091)

[2.2.1 局部评价方法 5](#_Toc453019092)

[2.2.2 全局评价方法 5](#_Toc453019093)

[2.3 研究方法 5](#_Toc453019094)

[2.3.1研究方法概述 5](#_Toc453019095)

[2.3.1研究方法讨论 5](#_Toc453019096)

[2.4 主要贡献 5](#_Toc453019097)

[第三章 实体链接算法实现 6](#_Toc453019098)

[3.1 问题建模 6](#_Toc453019099)

[3.2 算法介绍 6](#_Toc453019100)

[3.2.1在线知识库的建立 6](#_Toc453019101)

[3.2.2具有歧义文字和候选实体的选取 6](#_Toc453019102)

[3.2.3算法流程 6](#_Toc453019103)

[3.2.4局部评价方法选取 6](#_Toc453019104)

[3.2.5全局评价方法选取 6](#_Toc453019105)

[3.3 算法讨论 6](#_Toc453019106)

[第四章 实体链接实验评估 7](#_Toc453019107)

[4.1 实验搭建 7](#_Toc453019108)

[4.1.1测试环境说明 7](#_Toc453019109)

[4.1.2测试数据选取 7](#_Toc453019110)

[4.1.3对比算法设置 7](#_Toc453019111)

[4.2 数据分析 7](#_Toc453019112)

[4.2.1算法的正确性检验 7](#_Toc453019113)

[4.2.2文本数量和时间的关系 7](#_Toc453019114)

[4.2.3实体选取深度和广度与时间的关系 7](#_Toc453019115)

[4.2.4实体选取深度和广度与正确度的关系 7](#_Toc453019116)

[4.2.5文本密度和正确度的关系 7](#_Toc453019117)

[4.2.6和其他算法的对比 7](#_Toc453019118)

[4.3 参数影响 7](#_Toc453019119)

[4.3.1 kl和dot算法对比 7](#_Toc453019120)

[4.3.2 lambda参数设置 7](#_Toc453019121)

[4.3.3 restart probability参数设置 7](#_Toc453019122)

[4.4 本章讨论 7](#_Toc453019123)

[第五章 实体链接算法讨论与优化 8](#_Toc453019124)

[5.1 存在问题与解决算法 8](#_Toc453019125)

[5.2 优化算法 8](#_Toc453019126)

[第六章 图像匹配综述 9](#_Toc453019127)

[6.1 研究对象 9](#_Toc453019128)

[6.2 研究方法 9](#_Toc453019129)

[6.3 主要成果 9](#_Toc453019130)

[第七章 图像匹配算法实现 10](#_Toc453019131)

[7.1 原理简介 10](#_Toc453019132)

[7.2 算法实现 10](#_Toc453019133)

[第八章 图像匹配实验评估 11](#_Toc453019134)

[8.1 实验搭建 11](#_Toc453019135)

[8.2 数据分析 11](#_Toc453019136)

[8.3 参数影响 11](#_Toc453019137)

[8.4 本章小结 11](#_Toc453019138)

[参考文献 12](#_Toc453019139)

# 

# 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，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

另外，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，其中的值，代表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

在本文中，我们将用这种抽象距离去表达，两个实体概念的联系程度。我们《测试了两种不同分解方法的耗时，和准确程度》

# 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，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文字所包含的意思。它具有一些十分典型的运用，比如通过去处歧义，去扩充知识库，还有在进行网络搜索时给出更准确的建议。一般来说，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.

《》

## 2.2 相关研究方法

### 2.2.1 局部评价方法

《》

### 2.2.2 全局评价方法

《》

## 2.3 研究方法

### 2.3.1研究方法概述

### 2.3.1研究方法讨论

## 2.4 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)

# 第三章 实体链接算法实现

## 3.1 问题建模

(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))

## 3.2 算法介绍

### 3.2.1在线知识库的建立

### 3.2.2具有歧义文字和候选实体的选取

### 3.2.3算法流程

### 3.2.4局部评价方法选取

### 3.2.5全局评价方法选取

## 3.3 算法讨论

# 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文本数量和时间的关系

### 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文本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文本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文本。直到合并成完整的文本。如果还剩余未复制的文本，去除所有以复制的文本，形成新的文本，按照刚刚步骤再来一次，直到声誉为复制文本与上次一样。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.