

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

可重启的随机游走是一种对于两个抽象点联系的一种“距离”定义，它是通过某点随机遍历相邻节点来得到关于此点的落点在图中相对概率分布，从而确定与其他点的概念远近。它广泛的用于可抽象于图的问题结构中。

实体链接，主要指给定一个知识库，对文章中的文本进行去歧义的过程。它在生活中具有相当广泛的运用，例如扩充知识库，搜索时的去歧义等。本文的实体链接将建立在《Robust Entity Linking via Random Walks》的基础上，实现以可重启随机游走为算法核心的程序。本文的实验表明，实体链接算法在一般情况下（测试文章选自New York Times）具有较好的（70%左右）结果。

<图像匹配，为什么要研究以及算法概述>

本文的主要贡献在于，对于在线实体链接算法的实现，对于其的功能性测试，以及对于其的优化算法的提出。《TODO》

关键词：随机游走，实体链接，图像匹配

目 录

[THESIS OF BACHELOR 1](#_Toc453529889)

[第一章 技术背景综述 4](#_Toc453529890)

[1.1 随机游走阐述 4](#_Toc453529891)

[1.2 随机游走的计算 4](#_Toc453529892)

[1.3 随机游走在本文中的运用 5](#_Toc453529893)

[1.3.1 随机游走在实体链接(Entity linking)的运用 5](#_Toc453529894)

[1.3.2 随机游走在图像匹配的运用 5](#_Toc453529895)

[第二章 实体链接简介 6](#_Toc453529896)

[2.1研究对象简介 6](#_Toc453529897)

[2.2 相关研究方法 6](#_Toc453529898)

[2.2.1 局部评价方法 6](#_Toc453529899)

[2.2.2 全局评价方法 6](#_Toc453529900)

[2.3 论文研究方法 7](#_Toc453529901)

[2.3.1论文研究方法概述 7](#_Toc453529902)

[2.3.1论文研究方法讨论 7](#_Toc453529903)

[2.5 本文与论文算法实现差异讨论 7](#_Toc453529904)

[2.5 本文主要贡献 7](#_Toc453529905)

[第三章 本文实体链接算法实现 8](#_Toc453529906)

[3.1 问题建模 8](#_Toc453529907)

[3.2 算法介绍 8](#_Toc453529908)

[3.2.1在线知识库的建立 8](#_Toc453529909)

[3.2.2指代和实体的选取 9](#_Toc453529910)

[3.2.3算法流程 9](#_Toc453529911)

[3.2.4局部评价方法选取 10](#_Toc453529912)

[3.2.5全局评价方法选取 10](#_Toc453529913)

[3.3 算法局限性讨论 11](#_Toc453529914)

[第四章 实体链接实验评估 12](#_Toc453529915)

[4.1 实验搭建 12](#_Toc453529916)

[4.1.1测试环境说明 12](#_Toc453529917)

[4.1.2测试数据选取 12](#_Toc453529918)

[4.1.3对比算法设置 12](#_Toc453529919)

[4.2 数据分析 12](#_Toc453529920)

[4.2.1算法的离线基础效果检验 12](#_Toc453529921)

[4.2.2在线测试集效果检验 13](#_Toc453529922)

[4.2.3实体选取深度与正确度的关系 14](#_Toc453529923)

[4.2.4实体选取深度与时间的关系 14](#_Toc453529924)

[4.3 参数影响 14](#_Toc453529925)

[4.3.1 KL算法中 l参数设置 14](#_Toc453529926)

[4.3.2 not restart probability参数设置 15](#_Toc453529927)

[4.4 实验误差来源分析 15](#_Toc453529928)

[第五章 实体链接算法优化 16](#_Toc453529929)

[5.1 存在问题与解决算法 16](#_Toc453529930)

[5.2 优化算法概述 16](#_Toc453529931)

[第六章 图像匹配综述 17](#_Toc453529932)

[6.1 研究对象 17](#_Toc453529933)

[6.2 研究方法 17](#_Toc453529934)

[6.3 主要成果 17](#_Toc453529935)

[第七章 图像匹配算法实现 18](#_Toc453529936)

[7.1 原理简介 18](#_Toc453529937)

[7.2 算法实现 18](#_Toc453529938)

[第八章 图像匹配实验评估 19](#_Toc453529939)

[8.1 实验搭建 19](#_Toc453529940)

[8.2 数据分析 19](#_Toc453529941)

[8.3 参数影响 19](#_Toc453529942)

[8.4 本章小结 19](#_Toc453529943)

[参考文献 20](#_Toc453529944)

# 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：



其中，是随机独立变量，平均值为0，方差为1。是随机游走进行过程的时间，是在坐标平面上的位置。

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：



其中， 是n维矢量，W为概率转移矩阵(n\*n)， 为初始向量，p为重启概率。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，某些点会聚成团，而与其他点并无太多的连线，我们可以运用以下的方法，去很好的近似，达到缩减计算时间，缩减存储空间的目的。











其中 是把原图分割k份，每份份内边形成的对角矩阵，是原图剩余份与份之间边形成的稀疏矩阵。是对原矩阵做的近似分解。



## 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》所提供的方法。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，覆盖面广等特征，但相应的缺少全面的实体联系信息。
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](https://github.com/PublicShawn/RWR_FOR_ALGO)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。

# 第三章 本文实体链接算法实现

## 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

其中T的目标函数为：



L是局部评价函数，G是全局评价函数。

## 3.2 算法介绍

### 3.2.1在线知识库的建立

Wikipedia

**图片 3.1 WIKI链接**

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Cashew Apple(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。通过文章中的向外链接，实体与实体之间便链接起来，我们可以得到了一张关于指代的实体图（Entity graph）。

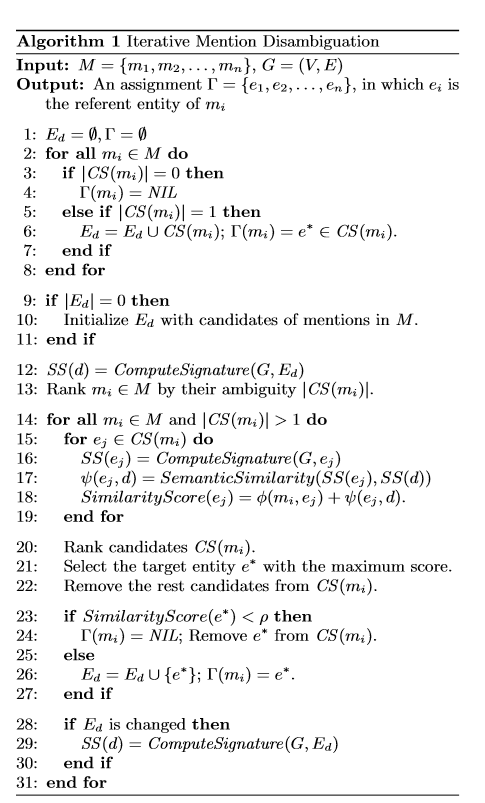
### 3.2.2指代和实体的选取

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。

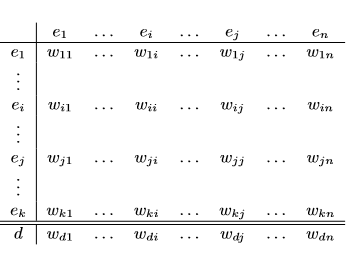
歧义候选的选取，本文所采用的方法是设定深度，和广度的阈值。深度代表着，从某个指代开始最大跳转数，广度代表着某个实体所能获取的最大相关的引用数。如上文所提，Apple->Apple Inc->iPhone->Phone是一个深度为4的设定，而Apple:[Apple Inc. , Apple Tree, Apple Corps]是一个广度为三的设定。

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，对算法的结果产生影响。本文选取的广度为8，一般来说它包含了大部分有用的信息，值得注意的是我们仅仅限制实体与实体之间的广度，对于候选我们的设定是20，超过了大部分词语所拥有的歧义条目。

### 3.2.3算法流程



**图片 3.2 算法**



**图片 3.3 签名算法**

1. 首先对于给定所有的指代，我们从Wikipedia上获取其的实体图。算法的目的是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重要程度联系强度作为加权（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



其中C是参数在计算文档中一起出现的次数。是指代， 是实体。

### 3.2.5全局评价方法选取

重启的随机游走的初始向量，有以下数学公式确定:



I是标准tfidf算法用来确定指代m的重要程度，其思路是某一个指代在本文档中出现的次数越多，越重要，在其他文档中出现的次数越多越不重要（说明此指代不容易区分文章的差别），P是先验概率，即。

算法中所用计算两个分布差距的函数，Zero-KL，由以下公式确定：





SS函数是以实体(e)为初始向量所进行随机游走的函数，如图3.3所示。同样的，SS(d)为文档(d)的初始向量所进行的随机游走函数，它们将会得到一个概率分布，3.3图中的d。文中的l取20.

## 3.3 算法局限性讨论

在线搭建知识库可以获得较新较有针对性的数据，更容易针对指代建立实体图，当深度小的时候，计算量相比全部计算数据库小很多。相应的，它也会缺少一些联系信息。在下文的实验中我们发现，指代的和实体的联系大多确立在第二层联系上，（文章中某一个实体出现在指代的某一个歧义的解释中），我们因此认为在线搭建数据库也是相当有效的。

我们发现，论文算法对于一开始实体的确定十分敏感。当一篇文章内没有一个指代可以唯一确定的时候。算法的正确性就会大幅下降。另外，由于指代的选取不是文章位置相关的，它会失去局部性特征，可能的情况是前后文的时效性和空间性都不同，此算法并不能辨认这种行文所带来的差异。

我们认为，去歧义过程中，大部分文档实体的内容逐渐增加的，而且增加的有效信息量是逐渐减少的，这符合一篇文章有中心点的习惯。这样的话，算法步骤4应该改为选择那些差异最小的实体加入，而不是先判断歧义数量最少的指代。同样的， 原论文实现的Zero-KL算法在去歧义的过程中并没有cosine similarity算法合理。试验同样验证了这点。

# 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

本文的测试平台是mac pro Retina(2015版)，CPU-i5，使用的编程语言是python，网络带宽为5mb/s，Wikipedia的ping值在205ms左右。

### 4.1.2测试数据选取

本文的测试分为两部分，一部分是离线数据测试，一部分是在线数据测试，离线数据测试数据量小主要针对于算法功能性上的评测，采用testcase模式，在线测试，数据量大，主要测试其应对噪声以及与实际文本契合程度。

本文的在线测试数据来源于nytimes以及economics网站最新文章，摘取的策略是选取section底下一篇或者两篇文章，总计共有15篇文章。每篇文章字符量在400-800左右，指代的选取方法如3.2所表示的，选取白名单和黑名单，并依照一定密度去选择。测试所用指代数量为411。本文算法的测评结果，是以人工评测的方法（由于选择的是最新的数据，算法实现的也是在线摘取wiki的方式），规则如下所示，对于每一个指代去wikipedia去寻找与本文最相近的意思，如果wiki上没有且答案为Nil或者wiki上的答案与算法得出的答案一致则算相同。

### 4.1.3对比算法设置

本文设定了两个对比算法，一个是先验概率算法，另外一个是simrank算法。先验概率算法在第三章得到描述，simrank算法摘取的是[stackoverflow](http://stackoverflow.com/questions/9767773/calculating-simrank-using-networkx)实现的算法。

## 4.2 数据分析

### 4.2.1算法的离线基础效果检验

本文为了检测算法的功能性，设计了7个检测样例。分别测试算法是否能针对局部词组组合的性质得到结果（考验其局部评价算法），算法是否能通过语意联系得出结果（考验其全局评价算法），算法是否能权衡结果（在局部概率相等的情况下，是否能选择语意上更接近，在都有联系的情况下是否选择更有联系的）

**图 4.1 离线测试正确率**

如图4.1所示，only\_rwr是只采用random walk签名的方法，prior是只采用先验概率的算法，normal-dot是采用本文框架的算法，但全局函数采用点积方式衡量两个分布的差，normal-KL是本文的算法。可以看到的是在离线测试小规模数据中，normal-dot算法结果更好，这是因为相比于KL算法，它在逻辑上更清晰，更符合人们理解文章的过程，它的核心思想是，每次选择那些与已选择实体在联系上差别最小的实体，也是投影最大的实体。我们希望，在原文本实体上最被强调的概念，在引入新的实体之时，也尽可能包含这个概念。而KL不能表达这一思想。Prior和simrank算法在小数据集面前落后于其他的算法，主要由于对于小数据集，局部算法所得出的值都差不多。而only\_rwr算法，由于缺少了局部信息，我们发现在一开始的时候，对于一些指代，当它存在大概率的赋值到某个实体时，only\_rwr并不能发现这种关系，导致引入的实体错误，由于采用的是迭代方法，也大大降低了以后的赋值正确率。

### 4.2.2在线测试集效果检验

**图4.2 在线测试正确率**

在线测试实体选取的深度3广度为8，每篇文章回复的response数量约为3万， 从结果上，我们发现原论文使用的KL方法并不能很好的达到目标效果。在测试中，我们分析可能产生的原因，1）KL算法更偏好于那些出入度小的实体，它们在图中几乎与其他并不相连。而这些实体的所得到的签名与文档签名，会具有更小的KL值。2）原论文的算法设计框架是迭代的，由于KL算法偏好与离散点，导致在文档信息不充分的情况下（特别的文档没有确定的实体，也即没有一个指代有唯一的候选）选错，使后来选取的实体越发偏离原文含义。3）原文算法在步骤四中容易出错，应该选择指代中候选与文档KL值最小的，而不是按照指代数量进行排序后再选择最小的。因为当该指代与现确定实体含义偏离较大的时候，容易引入错误概念。Normal-dot算法相比于KL算法在原因1）上有所避免，一定程度上减轻了2）迭代产生的影响，它的正确率得到了很大的提升，约为10%。而prior只采用简单的先验概率也获得十分良好的正确率，这主要是由于Wiki的结构，把最常用到的候选答案放在第一个，当所有候选的概率相等时，prior会选择第一个。相比于normal-dot高出1％的正确率，我们认为是主要来自于选择文章的偏好上，prior更容易处理那些文章中含有使用概率相差很大的常用词，而normal-dot则善于分辨使用概率相同含义相差大的词汇，在我们的测试中所反映出来的结果是，在生活类词汇的得分中prior高于normal-dot，在政治经济学类文章中，则正好相反。两者成互补关系。

### 4.2.3实体选取深度与正确度的关系

|  |  |  |
| --- | --- | --- |
| Algo\depth | 2 | 3 |
| normal-kl | 65.83% | 67.26% |

**表4-1 深度与正确度的关系**

从表4-1我们可以看出，随着深度的增加，算法正确性得到了增加，主要原因可能是提供了更多有效概念联系，但增加的幅度非常少，主要原因是1）第二层的说明（对于歧义的解释）就足以区分大多数的歧义的不同。2）文章如果文字有歧义，所做的说明大多落在第二层的解释当中。

### 4.2.4实体选取深度与时间的关系

歧义候选选择的深度影响着网络获取的时间，和计算时间。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| algorithm\depth | 1 | 2 | 3 | 4 |
| simrank | 小于1 分钟 | 15-30分钟 | 超过2小时 | 超过2小时 |
| normal-kl | 小于1 分钟 | 小于15分钟 | 45-90分钟 | 超过2小时 |
| prior | 小于1 分钟 | 小于10分钟 | 小于20分钟 | 小于1小时 |

**表4-2 深度与时间的关系**

从表中可以看出，当深度为4的时候论文的算法在python实现上已经极其缓慢，原因在于点多，矩阵运算太费时间。而simrank方法由于迭代收敛缓慢，在depth为三的情况下已经超时。先验概率算法只需要统计，它的耗时主要来自于网络请求。

## 4.3 参数影响

### 4.3.1 KL算法中 l参数设置

**图4.3 在线测试正确率**

在测试中，我们发现，KL算法中l的参数相比于重启概率来说并不会对正确率产生太大的影响。

### 4.3.2 not restart probability参数设置

**图4.4 在线测试正确率**

图中的prob是不重启的概率。重启概率的增加，能一定程度上增加normal-kl算法的正确率。原因可能是它的概率分布跟倾向于与它更有直接联系的相邻节点，避免了对于离散点访问概率的增加。

## 4.4 实验误差来源分析

本部分试验的误差可能来自于1）网络请求连续丢失（3次），导致某一实体信息丢失2）Wiki是大小写敏感，与测试用的文章大小写不统一，导致没有查到3）Wiki链接格式变化，导致一些链接无法被获取4）选取文章覆盖面小，由于选取文章时间段是在一天以内，导致取样不均匀5）人工评判词语含义失误。

# 第五章 实体链接算法优化

## 5.1 存在问题与解决算法

我们在原论文的基础上提出改进的思路。

原论文的实现对于初始赋值敏感。事实上，我们可以做如下优化。对于每一个指代的候选，选取与它链接某些实体，这些实体不属于其他候选所链接的实体集合中，我们把它称为特征实体，如果特征实体出现在指代相邻位置，我们把它赋值为此候选。注意此算法不同于原论文的算法在于，它首先确定了特征实体，而原论文首先是进行签名，所以增加了它的效率以及准确度。它也不同于先验概率，在于它是对候选的引申为判断依据，而不是候选与指代的频率关系。

计算量过大以及指代不能辨认在文中的位置。事实上我们可以用分布式的方法分割文章。把文章分成小的信息块，每个文章块先确定其的特征实体，再对特征实体以及与它链接实体进行签名，作为这个信息块的概念集合，对于每一指代中的候选计算它与概念集合的cosine similarity（向量点积）和先验概率，如果第一候选和第二候选分数相差小于阈值，则放弃本回合赋值，等到两个信息块的合并，再作出选择。这种好处是，对于某一个指代它受它所在文章位置周围文字的影响比受更远文字的影响更大。

## 5.2 优化算法概述

第一步，选取相邻的三个句子，为一个信息块，对每个块确定特征实体，把这个实体所链接的实体列入这个信息块概念集合，每个概念实体具有初始分数（tfidf）。

第二步，通过给剩余带有歧义的指代每个歧义进行分析，选取非零的点积除以概念数与先验概率相加最大的那个歧义实体，把它以及与它链接的实体列入概念集合，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/K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。

第三步，重复这个过程直到，对于所有剩余的歧义，它们的候选分数最大和次大相差小于阈值。

第四步，然后合并两个相临的信息块成为一个新的信息块，它们的概念集合归一化，重复选取刚刚没有赋值的指代。

第五步，直到信息块合并成完整的文章。如果还剩余未复制的指代，把它们赋值为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.