



中国科学院大学
University of Chinese Academy of Sciences

硕士学位论文

面向微博文本的实体链接方法研究

作者姓名: 李禹恒

指导教师: 吴一戎 研究员

付琨 研究员

中国科学院电子学研究所

学位类别: 工学硕士

学科专业: 信号与信息处理

研究所: 中国科学院电子学研究所

2015 年 5 月

Study on Entity Linking Method for Microblogs

By

LI Yuheng

A Thesis Submitted to

The University of Chinese Academy of Sciences

In partial fulfillment of the requirement

For the degree of

Master of Signal and Information Processing

Institute of Electronics

Chinese Academy of Sciences

April, 2015

研究成果声明

本人郑重声明：所提交的学位论文是我本人在指导教师的指导下进行的研究工作获得的研究成果。尽我所知，文中除特别标注和致谢的地方外，学位论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得中国科学院电子学研究所或其它教育机构的学位或证书所使用过的材料。与我一同工作的合作者对此研究工作所做的任何贡献均已在学位论文中作了明确的说明并表示了谢意。

特此申明。

签名: 李和生 日期: 2015.5.18

关于学位论文使用权的说明

本人完全了解中国科学院电子学研究所有关保留、使用学位论文的规定，其中包括：①电子所有权保管、并向有关部门送交学位论文的原件与复印件；②电子所可以采用影印、缩印或其他复制手段复制并保存学位论文；③电子所可允许学位论文被查阅或借阅；④电子所可以学术交流为目的，复制赠送和交换学位论文；⑤电子所可以公布学位论文的全部或部分内容（保密学位论文在解密后遵守此规定）。

签名: 李和生 日期: 2015.5.18

导师签名: 吴成伟 日期: 2015.5.18
付琨

摘要

微博（Twitter、Tumblr...）作为一种新兴媒体，日趋发展成为人类活动数字化、文本化的表现形式。然而人类自然语言的多样性和歧义性为计算机理解微博文本带来极大障碍。实体链接技术通过将实体别名链接到知识库中确定词条，可以消除语言歧义，具有广泛的应用价值。由于受到微博文本长度等特性的限制，传统面向长文本的实体链接方法对微博短文本链接准确率较低。为此本文对算法预处理和算法模型这两个实体链接核心环节进行创新性研究，并结合算法设计并实现实体链接原型系统。

本文主要工作概括如下：

1. 传统实体链接预处理方法需要充足上下文特征，而微博文本由于长度受限，无法满足这一要求。本文针对该问题对传统预处理方法进行优化改进。首先引入临近上下文相似度特征，捕捉实体类别信息，从而在语义相似或缺少语义信息的情况下，为消歧算法提供分辨依据；进一步利用了基于朴素贝叶斯的文本相似度特征，避免了传统方法中易于出现相似度为零的情况；最后提出改进的基于均分词频的实体流行度解决了传统词频统计方法对重定向页面统计缺失或重复统计的问题。实验表明，本文改进的预处理方法在提高了实体流行度特征的可靠性的基础上，降低了传统方法对上下文的依赖，为下一步面向微博文本的实体消歧算法提供了更具分辨力的特征，确保了消歧算法在微博文本集上的消歧能力。

2. 针对协同消歧算法模型训练复杂度高、对于微博文本链接准确率低的问题，提出层次化实体消歧算法（Hierarchical Entity Linking Algorithm, HEL）。该算法将同一用户下的不同消歧任务根据其指称项的模糊程度逐层消歧，并利用信息函数对指称项模糊性进行定量计算。同时利用确认实体池记录历史消歧结果，表征用户偏好，进一步指导下一层实体消歧任务。实验结果表明，本文算法在降低了模型训练复杂度的同时，提高了消歧算法在微博文本集上的链接准确率，也为其他基于微博文本的实体链接研究及应用提供了可借鉴的思路。

关键词：微博实体链接；实体消歧；自然语言处理；知识图谱；信息系统

Abstract

The microblog (Twitter, Tumblr...), as a form of new media, has become a digital and textualized manifestation of human activities. However the variety and ambiguity of human nature language set tremendous barriers for computers in comprehending microblog texts. Applying Entity Linking technology to microblogs could directly link mentions in microblogs to knowledgebase entity items, thus diminish such language ambiguity. The results of Entity Linking are of great practical value. Previous Entity Linking methods focus on long text corpus. However, performance of such methods may be limited by features like the maximum length restriction of microblog text. Bearing this problem, this dissertation undertakes a serious of creative and innovative study on two of the core stages of entity linking algorithm—feature selection and extraction. As a capstone project of this study, a Entity Linking prototype System based on proposed algorithm has been designed. Work of this dissertation has been summarized as follows:

1.Traditional methods rely on considerable quantity of context, which is deficient in microblog context due to word limitation. Regarding to this problem, an improved entity linking preprocessing method has been proposed. This method firstly introduces an innovative adjacent context feature to capture category information using the scarce context besides the entity mentions, and provide evidence for disambiguating algorithm under little semantic knowledge. Then a text similarity metric based on Naïve Bayes has been applied to avert void similarity problem in traditional metrics because of insufficient context information. Finally, an equipartition-frequency based entity popularity has been proposed to tackle the omittance and redundancy in traditional entity-mention pair counting in redirect pages. Experiment results show that based on the reliability enhancement of entity popularity, our improved preprocessing method weakens the reliance on the context from traditional methods, and provides a feature with greater discriminability to the following entity disambiguating algorithm, which guarantees a better disambiguating proformance on microblog corpus.

2.A Hierarchical Entity Linking algorithm (HEL) has been proposed to deal with high complexity and low accuracy problem in collaborative disambiguating methods. The algorithm conducts the disambiguating work by users hierarchically based on the ambiguity of their surface name, which could be calculated quantitatively using a proposed info-function. A concept of Certain Entity Pool has been proposed to represent user preferences, which subsequently be used in computing entity coherence with candidate entities. The HEL algorithm has been proved to be more efficient and accurate on microblog text than previous endeavors. It improves the existing collaborative disambiguating methods, and inspires other entity linking study on microblog text.

Keyword: Microblog Entity Linking; Entity Disambiguation; Nature Language Processing; Knowledge Graph; Information System

目录

摘要	I
Abstract	III
目录	V
第一章 绪论	1
1.1 课题研究背景及意义	1
1.2 国内外研究现状与进展	3
1.2.1. 长文本实体链接研究现状	3
1.2.2. 微博文本实体链接研究现状	6
1.3 本文的研究问题	7
1.4 论文主要工作与创新点	8
1.5 本文的内容安排	9
第二章 微博实体链接相关理论与方法	13
2.1 引言	13
2.2 实体链接任务定义	13
2.3 微博文本定义	15
2.4 知识库概述	16
2.4.1. 主流知识库综述	16
2.4.2. 维基百科页面分类	18
2.5 候选实体词典构建方法	20
2.6 实体链接文本处理方法	21
2.6.1. 字符串比较	21
2.6.2. 向量空间模型	22
2.6.3. 余弦相似度	23
2.6.4. TF-IDF 权重	24
2.6.5. 朴素贝叶斯分类器	24
2.7 实体链接效果评价	25

2.7.1. 实体链接的评测方法	25
2.7.2. 实体链接相关评测竞赛	27
2.8 本章小结	29
第三章 面向微博文本的预处理方法.....	31
3.1 引言	31
3.2 候选实体词典构建	32
3.3 传统实体链接预处理方法	33
3.3.1. 实体流行度提取	34
3.3.2. 文本上下文相似度提取	34
3.3.3. 实体相关性提取	35
3.4 面向微博文本的预处理方法	37
3.4.1. 基于均分词频的实体流行度提取	37
3.4.2. 基于朴素贝叶斯的文本相似度提取	38
3.4.3. 临近上下文文本特征提取	41
3.4.4. 面向层次消歧的实体相关性提取	43
3.5 实验结果与分析	44
3.5.1. 候选实体词典分析	44
3.5.2. 实体流行度分析	45
3.5.3. 临近上下文分析	45
3.6 本章小结	46
第四章 基于信息函数的层次化实体消歧方法.....	47
4.1 引言	47
4.2 传统实体消歧相关方法	47
4.2.1. 独立消歧方法	48
4.2.2. 协同消歧方法	49
4.3 层次化实体消歧算法	50
4.3.1. 方法框架	50
4.3.2. 信息函数	53
4.3.3. 层次消歧算法	54

4.3.4. 空实体预测	58
4.4 实验结果与分析	59
4.4.1. 实验数据	59
4.4.2. 评测指标	60
4.4.3. 实验结果	61
4.5 本章小结	65
第五章 实体链接原型系统设计.....	67
5.1 引言	67
5.2 系统总体设计	67
5.2.1. 软件开发平台	68
5.2.2. 功能模块设计	69
5.3 系统实现	71
5.3.1. 软件主界面	71
5.3.2. 链接信息窗	72
5.3.3. 算法设置面板	74
5.3.4. 算法评估报告	74
5.3.5. 增值应用扩展	74
5.4 本章小结	75
第六章 总结与展望.....	77
6.1 全文内容总结	77
6.2 下一步工作展望	78
参考文献.....	79
攻读硕士学位期间发表论文情况.....	83
致谢	85

第一章 绪论

1.1 课题研究背景及意义

信息是现代社会的重要组成部分，互联网的兴起为人类产生信息提供极大便利。21世纪初，随着以Twitter、Tumblr等微博服务类网站为代表的自媒体行业的飞速发展，互联网的信息结构已经从“少数人发布，多数人浏览”逐步转型成人人都是信息的发布者。随之引发的信息过载问题令人们不得不花费更多的时间和精力在海量的数据中搜寻想要的信息，过剩的信息与落后的信息检索技术已经一跃成为阻碍大数据时代技术进步的首要矛盾。

传统的基于关键词匹配的信息检索技术难以满足人们对精确搜索的需求，其主要原因在于自然语言具有歧义性，即存在一词多义、同义复指的问题。例如“布什”既可以代表美国前总统“乔治·W·布什”，也可以指代他的父亲，那么在文本“布什预计本月20日携夫人访问中国。”中的“布什”一词就存在指代不清的问题。在缺少更多先验知识的前提下，单纯基于关键词的匹配几乎无法从根本上解决自然语言模糊带来的问题；此外，搜索引擎缺乏对于网页中人、事、物之间语义关系的理解，字面层次的查询匹配丢失掉大量不包含查询词但内容相关的信息，同时引入了语义无关的噪声。综上所述，我们需要一种技术，通过对自然语言中名词表达与客观世界实体的准确对应，消除语言歧义。

2009年NIST在其主办的文本分析会议(Text Analysis Conference，简称TAC)上提出**实体链接（Entity Linking）**评测任务。实体链接利用现有的百科知识库（如Wikipedia、百度百科...）作为实体知识来源，将文本中出现的实体别名与知识库中条目进行映射，即实现了从文本中无意义的字符串到客观世界准确实体的转化，解决了上文所述问题。

实体链接研究主要包含以下三方面意义：

首先，实体链接可以增强用户文本浏览体验。在用户浏览微博的过程中，实体链接系统可以自动将文本中的指称项转化成锚文本，并指向知识库实体词条，从而满足用户扩展阅读的需求；进一步，通过对实体链接结果进行倒排索引，可以实现

如图 1-1 的基于知识图谱的搜索（实体检索），得到查询实体多源信息（微博、社交网络、知识库、媒体...）的聚合报告；此外，利用实体与不同语言别名的映射关系，可以实现高精度的机器翻译，从而进一步降低用户阅读障碍；

其次，实体链接可以实现更准确的机器理解与预测分析。例如，利用实体链接结果可以提高自动问答系统中语义解析器的解析能力，使之准确把握用户查询中的实体名词；基于实体链接的推荐系统消除了关键词模糊问题，能够准确识别用户感兴趣的对像，从而实现精准推荐，既节约广告投放者的开支，也方便真正有需求的用户快速定位产品，实现双赢；

最后，实体链接有助于知识库的积累和更新。今天，包含 4,740,834 个实体页面的维基百科（英文版）每天会增加逾 800 条新记录，每秒钟产生约 10 次记录修改¹，完全依靠人工去发现这些新知识既费时又费力，实体链接可以识别出当前实体别名是否对应具体对象，以及该对象是谁，从而自发地实现知识库的补充及更新。

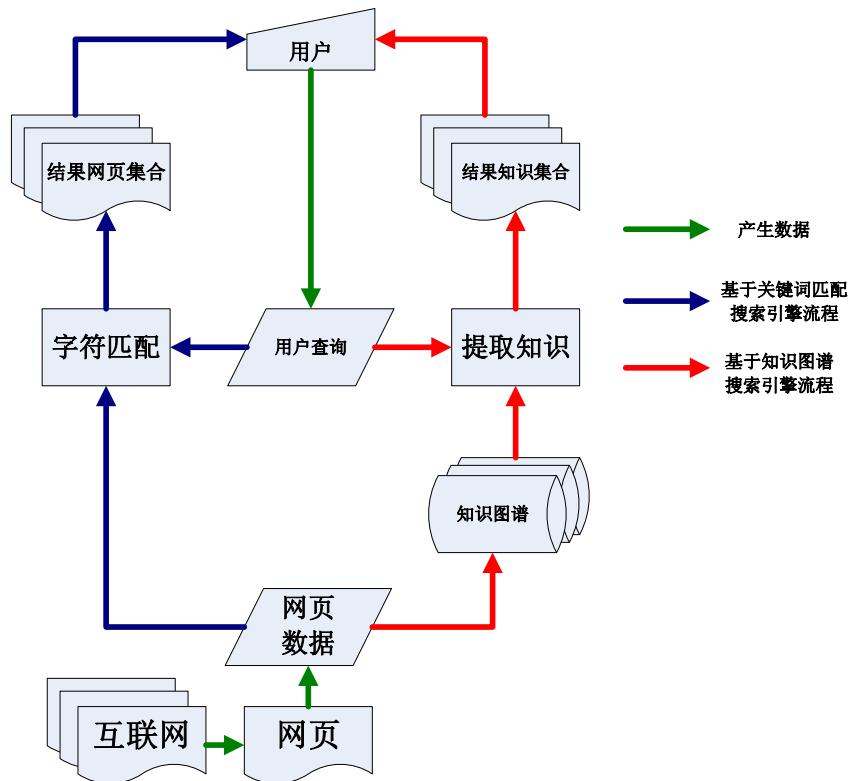


图 1-1 关键词搜索与知识图谱搜索示意图

¹ 数据来源：<http://en.m.wikipedia.org/wiki/Wikipedia:Statistics>

1.2 国内外研究现状与进展

根据研究对象的不同，实体链接任务可以分为面向网络长文本（web long document）的实体链接和面向微博短文本（microblog short document）的实体链接，目前的研究主要针对前者展开，本节将分别对这两类方法的研究现状进行介绍。

1.2.1. 长文本实体链接研究现状

2007 年，Rada Mihalcea 和 Andras Csomai 提出一套广义的实体链接系统——Wikify!，该系统利用维基百科作为知识库，首先对实体别名进行识别，进而使用基于知识库的无监督方法和基于二分类的方法进行词义消歧。

随着 2009 年 NIST 在 TAC 上的 KBP(Knowledge Base Population system)中，将实体链接列为其中一项评测任务，实体链接渐渐与实体识别任务独立出来，将研究重点落在基于实体词典的实体消歧任务上。对于实体消歧问题，目前的研究根据模型训练方法可分成两类：1) 有监督的消歧方法，利用标注训练集对候选实体 R_j^i 的排序模式进行“学习”：包括二分类方法、机器学习方法、概率方法和基于图的方法；2) 无监督的消歧方法，利用无标注语料库，利用指称项与候选实体在指定特征上的相似度，通过排序函数进行消歧：包括基于向量空间模型(vector space model, VSM)的方法以及信息抽取方法。

此外，根据考虑的特征，也可以将消歧方法分成三类：

1. 独立消歧方法：该方法认为待消歧实体提及之间相互独立，即消歧算法仅考虑查询文档与候选实体上下文之间的关系而没有利用同一篇文档（或用户）下不同实体提及之间的关联性假说^[1-8]。

2. 协同消歧方法：该方法认为一篇文档会涉及一个或多个主题，而同一篇文档中提及的实体相互关联。该类方法利用这种“主题相关性”对同一篇文档中的提及和实体构建图模型并通过兴趣得分游走“协同的”进行链接^[9-19]。

3. 联合消歧方法：对于提及的实体链接任务，这类方法利用其他文档中和它具有相似字面特征和上下文的实体提及来进行支持，即有效的利用了跨文档特征作为当前查询实例的增强^[20, 21]。

1.2.1.1. 有监督的消歧方法

有监督的消歧方法一般利用标注训练集对模型进行训练，使其能够“学会”从候选实体中区分出真正应被链接的实体。

一些研究将实体消歧视为一个二分类问题。Zhang^[22, 23]、Chen^[24]等人使用支持向量机（Support vector machines, SVM）作为分类器，Lehmann^[6]、Monahan^[25]等人使用二分 logistic 分类器，Varma 等人^[26]使用朴素贝叶斯分类器和 K 近邻分类器。二分类法存在的一些问题，首先一个提及可能存在不止一个候选实体被标为正值，因此不可避免要对这些正值进行二次选择，方法包括基于置信度的方法^[6, 26]、基于VSM 的方法^[23]以及 SVM 排序方法^[22]。此外，模型的训练样本既不均衡，负样本的比例要远高于正样本。

与二分类方法不同，排序方法将实体消歧任务理解成根据特征向量对候选实体进行打分排序的过程，并将排位最高的候选实体作为链接结果返回。这样就有效的避免了二分类方法中二次选择的问题，同时，该方法对于每一个实体提及的候选实体集要建立一个独立的排序，有效避免了训练样本不均的问题。此外，模型训练中会同时考虑提及的全部候选实体，而不是像分类器那样将其独立计算，这也便于利用实体间的特征。此外，Shen 等人^[15]利用了实体流行度、上下文相似度、语义相似度和全局主题一致性四个特征的线性组合作为排序函数，并利用 SVM 学习权重。

基于概率的方法不利用特征向量直接计算指称项与其候选实体之间的“相似性”，而是通过对数据的观察、统计，试图找出哪个候选实体“最有可能”被链接到该指称项。韩先培等人基于这种思想提出了一种产生式的实体提及模型^[27]，该方法将三种异构的特征（包括实体流行度、实体名称和上下文）融合进一个概率产生式模型，实体流行度可以告诉我们一个实体出现在一篇文档中的概率，实体名称可以告诉我们一个实体被某个提及所表示的概率，上下文可以告诉我们一个实体出现在当前上下文中的概率。模型假设观察到的任意一个指称项是经过以上三步得到的。Demartini^[28]等人提出 ZenCrowd 系统，利用人的知识来提高链接结果的准确率。通过一个概率推理模型来动态的分析来自众包平台（Crowdsourcing Platform）的人工答案以及机器生成的答案，如果机器给出的答案不够具体，系统会将其分割成若干具体的小问题交给人工回答，人工给出的答案则作为反馈提交给概率推理系统，并有其进行综合分析，产生结果。

基于图的方法对同一篇文档中的全部实体、提及之间的关系进行全局建模。Han

等人^[13]基于图的联合实体链接方法（graph based collective entity linking method）中利用相关图（Referent Graph）对上下文相似性和映射实体间相关性进行建模，之后利用联合推断算法在图中同时对全部指称项进行推断。Hoffart 等人^[10]则综合利用了三种特征来建图，其中提及-实体的边通过实体流行度和文本相似性加权，实体-实体的边则由实体相关性加权，利用这样一个图进行消歧的过程实际上就是在图中寻找给定提及与一个实体最稠密子图（仅包含一个提及-实体对）的过程。微博实体链接方面，Shen^[16]等人提出的 KAURI 能够将微博间用户兴趣与微博内局部知识建模到一个图模型中，其中微博内局部知识包括实体流行度、上下文文本相似性以及同篇微博中实体间的相关性，考虑到一篇微博的长度不足以提供足够的信息量，KAURI 基于同一用户关注近似主题这一假设，对跨文档的用户兴趣进行建模。

除了上述方法，一些研究利用模型组合来提高实体链接算法的效果，通过对那些使用不同特征的方法进行互补，已得到最优的解决方案。近年来，随着越来越对的实体链接平台的开放，模型组合克服了不同链接模型的问题，在工程上得到了很好的应用。Zhang 等人^[22]第一次将模型组合策略应用到实体链接任务中，其模型涵盖了包括基于信息抽取、学习排序和二分类在内的三种消歧方法，并通过利用标注数据训练一个 SVM 三分类器来判断使用哪种消歧算法更为可靠。此外 Ji 和 Crishman^[29]应用投票策略组合了九种消歧模型，实验表明其组合模型的链接准确率要高于单独使用任意一种模型，其组合模型准确率与 TAC-KBP2010 最好的系统相比，提高了 4.7%。

1.2.1.2. 非监督消歧方法

人工标注的训练数据集需要耗费大量人力，同时由于个人对实体理解的偏差，训练集的可靠性也受到人为因素的影响而不够准确。一种简单的无监督消歧方法就是基于向量空间模型（见本文 2.6.2 节）。该方法首先计算指称项和实体之间特征向量的距离，之后将相似度最高的那个指称项对应的实体作为候选实体返回。具体方法因所选择的特征和相似度度量方法而异。Chen 等人^[4]利用提及和实体上下文词组成词袋，并计算其 TF-IDF 相似度，在 TAC-KBP2010 数据集上取得 71.2 的准确率。Cucerzan^[11]提取了候选实体页面中所有的实体和分类标签作为实体特征向量，以及提及上下文所有相关实体作为提及的特征向量，并通过取向量间的最大相似度来找到候选实体。此外，Han, Zhao 等人^[30]提取提及和实体上下文的维基百科概念

(Wikipedia concept) 作为特征向量，并利用所有语义关联^[31]的加权平均来计算其相似度。

除此之外，一些研究^[22, 26, 32, 33]还将实体消歧任务转化为信息抽取任务，他们将每一个候选实体视作独立的文档，提及和其上下文文档组成一个查询，检索系统最终根据查询实例返回相关性最高的候选实体作为检索结果。

1.2.2.微博文本实体链接研究现状

目前的实体链接研究仍集中在对于长文本中指称项链接方法上，而在如何提高实体链接算法在短文本上的效果这一命题上研究较少。与长文档相比，微博文本长度受限，能够提供的上下文信息很少，且用语较不规范，常见缩写、别字、口语化的表达，这些因素都限制了基于网络长文本的实体链接方法在微博文本上的效果。

早期研究选择将整篇微博链接到一个或多个相关的主题上，属于帖级别。Meij 等人^[34]从微博中抽取 ngram 并找到其在知识库中的相关概念，并通过排序选择最相关的概念链接到整篇微博上。Guo 等人^[35]这种思想应用在微博到新闻的链接任务中。

与传统实体链接任务更相关的一类方法是词级别的链接。早起研究主要针对指称项的识别和对微博内容的处理，在语言层面上解决了错误拼写等问题。Liu 等人^[20]充分利用提到到提及，实体到实体以及提到到实体的相似度，并且针对未登录指称项，利用机器学习方法进行处理。Guo 等人^[9]使用结构化的 SVM 来协同优化提及识别和消歧任务。

Derczynski 等人^[36]通过分析传统 NLP 技术在处理微博文本上的效果时发现，传统的实体识别和词性标注在 Twitter 语料库上表现欠佳，并将其归因于微博文本中诸如拼写错误、大小写错用、滥用缩写等不规范的用语，以及上下文不足，缺乏充足语言资源等问题。针对中文微博实体链接，朱敏等人^[37]提出使用改进的拼音编辑距离对不规范的用语进行模糊匹配。

此外，有些研究考虑利用微博的结构特点获得额外的信息来扩充上下文。Jiang 等人^[38]将同用户的其他帖子，连同其评论和转发纳入指称项的上下文，Shen 等人^[16]进一步分析用户的微博内容，并对其兴趣进行建模，从而利用兴趣模型为候选实体分配权重。以上两种方法结合了更多上下文以外的微博独有特性，但对于发文、回复、转发量较少的用户，难以建立合适的模型，此外上述方法基于一个假设，即同

一个用户发表的博文主题相关，该假设的效力有待进一步的研究。

1.3 本文的研究问题

如上所述，受到微博文本特点的限制，传统的针对长文本的特征已经不能满足面向微博文本的实体链接任务的需要；另一方面，传统协同消歧方法迭代过程中易受到非相关实体的干扰，且迭代效率较低，不适于微博文本实时高效处理的应用需求。本文针对以上问题展开研究，研究思路如图 1-2 所示。

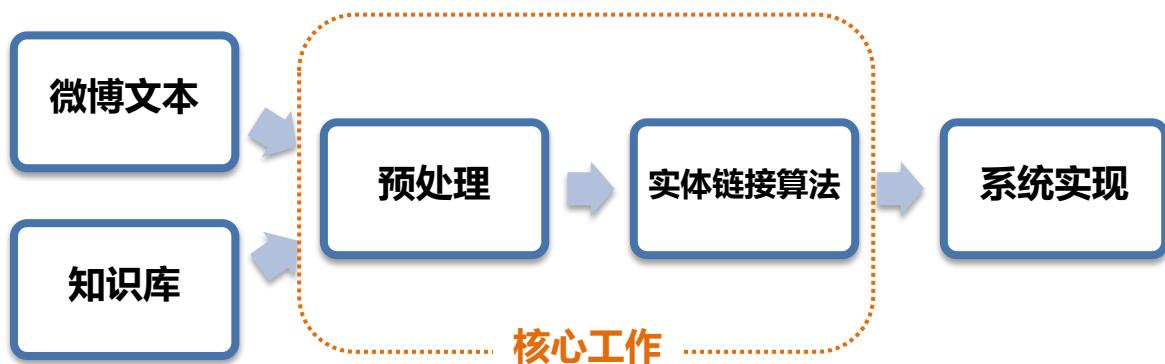


图 1-2 本文的研究思路

实体链接核心工作包括预处理和链接算法两部分，对于实体链接预处理环节，我们主要对以下问题进行研究：1) 传统算法提取特征实体流行度过程中，由于维基百科重定向和歧义页面机制的存在，许多链接文本并没有直接指向实体页面，而是重定向到歧义页面，传统的实体流行度提取只关注那些能够链接到实体页面的链接文本，这样无疑中就丢失了很多维基百科中存在提及-实体信息；2) 传统方法中文本上下文可以利用基于词典的向量空间模型来表征并利用余弦相似度来衡量上下文与候选实体文本内容的相似程度，但向量空间在表示文本的时候维度过高且十分稀疏，对于需要结合多种异构的特征进行综合相似度的计算的实体消歧模型并不适用，同时对于本研究对象微博文本，有些长尾实体在整个知识库中出现的频率较低（或上下文信息不足），对于这类数据在计算余弦相似度的时候很容易出现相似度为零的情况，使算法丧失在长尾中进一步的分辨能力；3) 传统利用上下文文本相似度的算法对与微博文本表现并不理想，首要原因在于微博文本长度较短，可供上下文相似度使用的语义信息十分有限，有的微博甚至会出现无上下文的极端情况，不仅如

此，我们发现一些同名实体即便主题相关，具有相似的上下文，仍会因为其自身类别属性的不同而具有较大的差异，如电影“泰坦尼克号”和游轮“泰坦尼克号”。以上问题会严重影响实体链接预处理的效果，进而影响到消歧算法的表现。

根据我们的分析，目前主流的链接算法存在如下问题：首先基于独立特征的消歧算法的特征选择存在一定的局限性，没有充分利用同一文档中实体主题相关性的假设，忽略了实体间相关性的作用，而对于富含用户特征、主题标签的微博文本中，实体之间的关联对消歧算法有这很重要的指导作用；其次，对于考虑实体相关性的协同消歧算法，目前的方法将全部候选实体同时建模，并通过对传播函数的多轮迭得到全局最优解，然而这种策略会引入不相关实体的负面影响，且迭代算法效率较低，增加了时间成本。

针对上述问题，本文首先从改善特征提取方法、引入新特征、改善上下文相似度计算方法这三个角度对实体链接预处理步骤进行改进，进而基于优化的预处理结果，创新性提出层次化的实体链接算法来解决传统消歧算法存在的问题。

1.4 论文主要工作与创新点

本文主要工作围绕面向微博文本的实体链接任务展开，结合当前实体链接的研究成果，针对其在短文本链接中存在的问题，从算法预处理和算法模型两个方面进行创新性实践，并结合本文算法设计实现了一套实体链接原型系统。主要工作概括如下：

1. 对国内外实体链接研究现状进行调研，对主流知识库特性及实体链接常用特征特性进行分析比较，并结合微博文本特点提出传统基于长文本实体链接方法面临的问题；
2. 利用维基百科作为知识库，提出基于微博文本的层次化实体链接方法，将微博词条中的实体别名链接到知识库相关实体页面，从而消除微博文本中实体名词歧义。实体链接任务包括两个关键步骤，本文首先对知识库进行预处理、特征提取并产生候选实体词典，在此基础上提出层次消歧算法对指称项候选实体进行歧义消解。针对传统实体链接预处理方法对微博文本针对性不足，传统协同消歧方法建模成本高，且受到非相关实体影响等问题，分别提出面向微博文本的预处理方法与基于信息函数的层次化实体消歧方法，解决了上述问题，提高了实体链接方法在微博文本

应用场景下的效果和效率；

3. 结合文本算法设计并实现实体链接原型系统。本文利用 WPF 技术，基于模块化设计思路将包括预处理、链接算法、算法评价在内的实体链接关键步骤分别独立实现，并在此基础上预留功能扩展接口。软件试用表明，本文所设计的实体链接系统在功能上满足科研需求，并能够对实体链接结果进行有效地组织表达，极大的增强了文本挖掘研究的可读性，同时为下一步基于链接结果的二次开发提供了平台。

本文主要在以下方面开展创新性研究：

1. 传统实体链接预处理方法需要充足上下文特征，而微博文本由于长度受限，无法满足这一要求。本文针对该问题对传统预处理方法进行优化改进。首先引入临近上下文相似度特征，捕捉实体类别信息，从而在语义相似或缺少语义信息的情况下，为消歧算法提供分辨依据；进一步利用了基于朴素贝叶斯的文本相似度特征，避免了传统方法中易于出现相似度为零的情况；最后提出改进的基于均分词频的实体流行度解决了传统词频统计方法对重定向页面统计缺失或重复统计的问题。实验表明，本文改进的预处理方法在提高了实体流行度特征的可靠性的基础上，降低了传统方法对上下文的依赖，为下一步面向微博文本的实体消歧算法提供了更具分辨力的特征，确保了消歧算法在微博文本集上的消歧能力。

2. 针对协同消歧算法模型训练复杂度高、对于微博文本链接准确率低的问题，提出层次化实体消歧算法（Hierarchical Entity Linking Algorithm, HEL）。该算法将同一用户下的不同消歧任务根据其指称项的模糊程度逐层消歧，并利用信息函数对指称项模糊性进行定量计算。同时利用确认实体池记录历史消歧结果，表征用户偏好，进一步指导下一层实体消歧任务。实验结果表明，本文算法在降低了模型训练复杂度的同时，提高了消歧算法在微博文本集上的链接准确率，也为其他基于微博文本的实体链接研究及应用提供了可借鉴的思路。

1.5 本文的内容安排

本文主要分为六章，各章主要内容安排如下：

第一章主要论述了本文的研究背景、研究意义，简要概括当前实体链接任务的研究现状与研究进展，并引出本文研究问题、主要工作及创新点。

第二章概述了微博文本实体链接相关理论与方法，包括任务定义、知识库概述、

候选实体词典构建方法及与本文相关的实体链接中文本处理方法，最后，给出实体链接效果评价方法及相关评测竞赛。

第三章主要研究实体链接的预处理方法。首先简要介绍本文候选实体词典构建方法，接着介绍传统预处理方法，在此基础上具体介绍本文对微博文本的预处理方法，分别提出基于均分词频的实体流行度、基于朴素贝叶斯的文本相似度及临近上下文特征，并针对实验结果进行分析讨论。

第四章主要研究基于信息函数的层次化实体消歧方法。首先介绍传统实体消歧相关方法，在此基础上分别阐述了层次化实体消歧算法框架、信息函数定义、算法流程及空实体预测，并针对实验结果进行分析讨论。

第五章研究基于微博文本的实体链接原型系统设计。首先给出系统总体设计框架，在此基础上详细介绍各逻辑模块功能，最后具体演示系统运行步骤及功能界面。

第六章总结全文内容。对本文所取得的成果进行归纳总结，并针对本文方法存在的不足对后续研究工作进行展望。

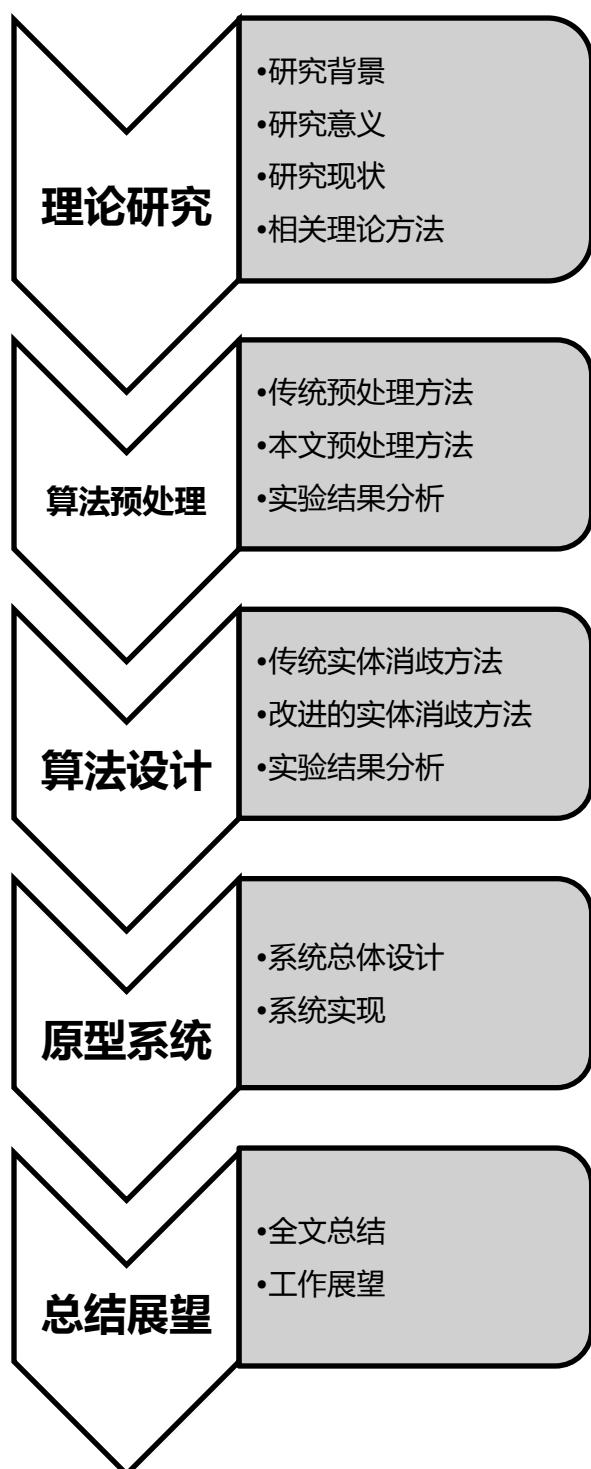


图 1-3 本文组织结构

第二章 微博实体链接相关理论与方法

2.1 引言

上一章我们介绍了实体链接研究的背景意义及针对微博文本实体链接的主要研究问题。本章将首先给出实体链接的完整定义，并对实体链接相关专有名词进行解析。作为本文的研究对象，微博文本具有许多独有的特性，充分了解这些特性可以便于我们有针对性的设计算法。知识库既是实体链接模型的训练集，同时也是任务输出，本章将对目前主流的知识库进行全面概述，并着重针对本文使用的维基百科的各类页面特征进行详细介绍。此外，本章还对候选实体构建、文本处理、实体链接效果评价等相关理论与方法进行全面梳理和介绍，为后文的研究工作提供理论指导。

在文章结构上，第一、二节分别介绍实体链接任务及微博，第四节概述当前主流知识库及其特点；第五节介绍基于知识库的候选实体词典的生成方法；在此基础上，于第六节详细介绍与本文研究相关的实体链接文本处理方法，最后简要介绍链接效果评价方法及相关评测竞赛。

2.2 实体链接任务定义

实体链接任务本质上是根据给定查询实例来返回目标实体的过程。其中查询实例由查询词和查询文档组成，查询词可以理解成目标实体的别名，查询文档是查询词所在的文章。我们注意到在给定查询词的条件下，可以限定一部分目标实体，即那些别名中不包含查询词的目标实体自然被排除在外，因此查询词与目标实体之间的关系是固定的，或者说是“静态的”。

与查询词相比，查询文档对目标实体的限定能力更宽泛，因为没有一个准确的规则来限定什么样的上下文（查询文档）更“适合”某一个目标实体。因此这里查询文档与目标实体之间的关系非固定的，或者说是“动态的”。

通过对以上两类输入条件进行学习，便可以筛选出对于查询实例最匹配的候选实体作为目标实体。其中查询词约束对应实体链接的第一步，即通过生成候选实体

词典，找到所有查询词潜在对应的实体作为候选实体；而利用查询文档筛选目标实体的过程对应实体链接第二步，即在已知候选实体的基础上，利用查询文档提供的特征进行实体消歧。下面分别给出本文涉及到的实体链接相关名词的定义：

1. 命名实体 (entity)：现实世界中的一个无歧义的事物，存在知识库中唯一的页面(Article)对其进行描述。

2. 实体提及 (mention) :亦被称作指称项，是命名实体的别名，并与知识库中的实体潜在相连。

3. 候选实体词典(candidate dictionary):对于任意实体提及 $m_j^i \in M$ ，他的映射实体 e_j^i 应该能够被 m_j^i 所指代，因此我们可以首先得到所有可以被 m_j^i 所指代的实体集合 R_j^i ，即候选实体集。为了能够得到每一个 m_j^i 所对应的 R_j^i ，需要生成一个词典 D ，该词典尽可能多的收纳包括词态变化、缩写、拼写错误、不规范用此等各种形式的提及以及他们可能对应的候选实体。这个词典可以理解成一组键值对，其中键代表实体提及，值代表一组候选实体。

4. 微博实体链接 (tweet entity linking): 基于任务设置，我们将(1)来自不同用户的 tweet 集合 T 以及(2) T 中识别得到的一组命名实体提及 M ，作为输入。令 $|T|$ 为 T 中微博数， T 中每条微博用 $1 \leq i \leq |T|$ 索引，第 i 条微博用 t_i 表示。 $t_i \in T$ 中被识别出的提及用 $M^i \in M$ 表示，其中 $|M^i|$ 为 M^i 中的提及数。 M^i 中的每条提及用 $1 \leq j \leq |M^i|$ 索引，其中第 j 条提及用 m_j^i 表示。下面正式给出微博实体链接任务的定义：

定义 2.1: 微博实体链接。给定由一些微博用户发布的微博文本集 T ，及知识库中命名实体提及集合 M ，目标是确定提及 $m_j^i \in M$ 在知识库中的映射目标实体 e_j^i 。如果 m_j^i 在知识库中不存在对应的目标实体，返回 NIL 。

2.3 微博文本定义

微博 (Mircoblog) 泛指通过关注机制分享简短实时信息的广播式社交网络平台。本文主要研究对象是国外著名微博服务平台 Twitter 上产生的文本消息，被称作 Tweet (推文，后文称为微博)，其单篇最大支持 140 个字符。以下简要介绍微博文本中特性成分：

- 1.提及用户：Tweet 中通过“@+用户名”的格式来提及相关用户，被提及用户可以收到消息，从而产生互动效应，此处的提及不同于后文提到的实体的提及，为了区分，此处我们称之为用户提及。
- 2.话题标签：Tweet 中通过“#+话题”的格式来参与话题，话题作为分类讨论的标识符，可以将所有提到该话题的推文组织在一起。
- 3.短链接²：Tweet 中使用短链接算法压缩字符长度。

Tweet 中经常会提及一些命名实体，如图 2-1 所示三篇微博中就包含三个命名实体，实际上经分析^[16]我们所使用的人工标注数据集中超过 45.08% 的 tweet 包含一个以上的命名实体。

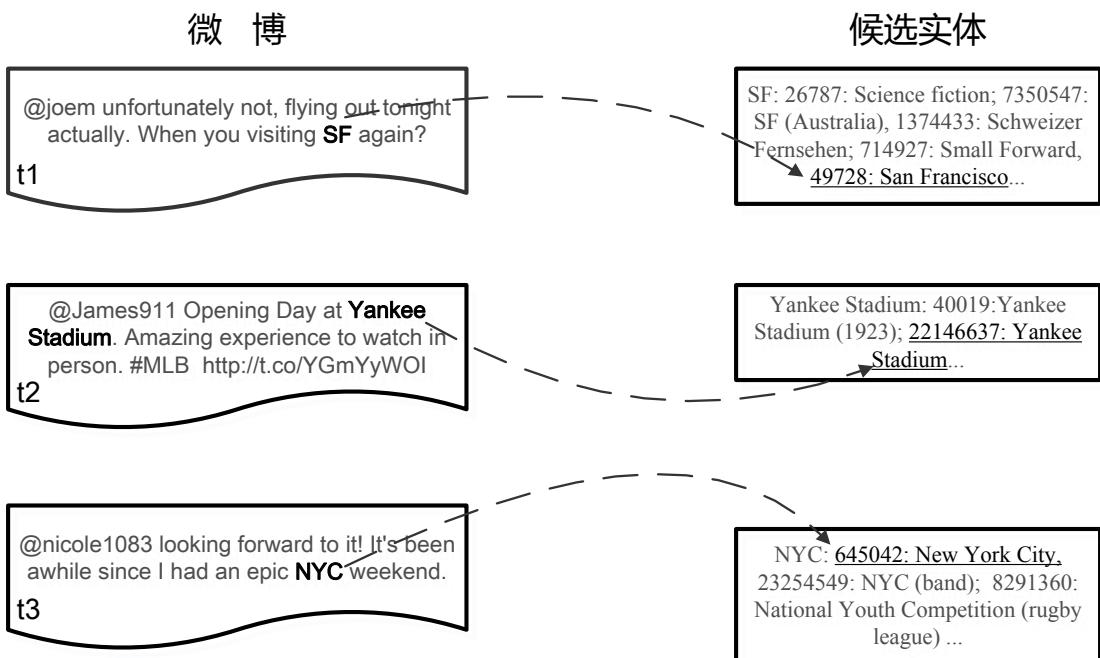


图 2-1 实体链接任务示例。识别出的指称项加粗表示，映射到的真实候选实体下划线表示。

² 用户提及、话题标签和短连接示例请参照表3-1第二条微博中“@James911”，“#MLB”，以及“<http://t.co/YGmYyWOI>”

2.4 知识库概述

如上一章提到的，知识库是知识图谱的数据中心，是实体链接的输出来源，更是实体链接方法重要的知识来源。知识库的完备程度决定了实体链接方法的召回率，同时由于目前实体链接方法的候选实体词典是基于知识库提取的，因此知识库对于实体的描述能力和覆盖率也对实体链接算法有很大的影响，本文采用维基百科作为知识库。2.4.1 中将例举当前主流的知识库，2.4.2 中将对维基百科页面分类及可用信息进行介绍。

2.4.1. 主流知识库综述

1. 维基百科

维基百科³是一个自由、免费、内容开放的百科全书协作计划，参与者来自世界各地。这个站点使用 Wiki，这意味着任何人都可以编辑维基百科中的任何文章及条目。维基百科是一个基于 wiki 技术的多语言百科全书协作计划，也是一部用不同语言写成的网络百科全书，其目标及宗旨是为全人类提供自由的百科全书，是一个动态的、可自由访问和编辑的全球知识体，也被称作“人民的百科全书”。维基百科由来自全世界的自愿者协同写作。自 2001 年英文版成立以来，维基百科不断的快速增长，已经成为最大的资料来源网站之一，在 2008 年吸引了超过 684,000,000 的访客。目前在超过 250 种的语言版本中，共有 6 万名以上的使用者贡献了超过 1000 万篇条目。每天有数十万的访客作出数十万次的编辑，并建立数千篇新条目以让维基百科的内容变得更完整。

2. DBpedia

DBpedia^[39]是由柏林自由大学和莱比锡大学研究人员发起的一个众包（Crowd-Sourced）项目，致力于从维基百科中抽取结构化的信息，更充分的利用维基百科中海量的数据和知识。这些结构化的信息包括信息框、类别标签、图像、地理坐标以及外部链接等，并以资源描述框架(Resource Description Framework, RDF) 形式将这些信息发布于互联网上，从而可以与其他符合 RDF 规范的开放数据（如

³<http://www.wikipedia.org/>

Freebase, OpenCyc 等)互联。DBpedia 允许用户针对维基百科提出复杂的检索请求，并返回维基百科知识作为答案。DB 第一个版本发布于 2007 年，据最新数据统计⁴，英文版 DBpedia 可以描述超过 458 万种实体

3. Freebase

Freebase^[40]最初由美国 Metaweb 公司开发，并于 2007 年公开发布。2010 年 Metaweb 被 Google 收购。Freebase 是一个协作性知识库，主要尤其社区成员创制的元数据 (metadata) 构成。Freebase 继承了从多个来源采集并整理的结构化数据。最近的 2014 版本包括大约 4300 万个主题。2014 年 12 月，Google 决定关闭 Freebase，并将其全部转移到 Wikidata 上⁵。

4. YAGO

YAGO^[41, 42]是由马克思·普朗克计算机科学研究所 (Max Planck Institute for Computer Science) 开发的基于 Wikipedia、WordNet 和 GeoNames 的大规模语义知识库。第一个版本发布于 2008 年，最新版本的 YAGO3 包含逾 1000 万实体，以及对于这些实体的超过 1.2 亿的事实 (facts)。YAGO 的内容包含从 Wikipedia 中抽取类别、重定向和信息框信息，从 WordNet 中抽取同义词和上位关系信息以及从 GeoNames⁶中获得的地理信息。YAGO 的人工评价准确率达到 95%。

5. Wikidata

Wikidata 是由维基媒体基金会运营的一个协作项目。Wikidata 项目始于 2012 年。2014 年的版本包含 1400 万主题。Wikidata 与 DBpedia 的区别在于，DBpedia 作为来源以抽取结构化信息，而 Wikidata 并不依赖于 Wikipedia，而是通过众包的方式直接编辑数据库条目。

6. Knowledge Graph& Knowledge Vault

Google 于 2012 年发布 Knowledge Graph，旨在通过语义搜索改善其检索结果，2012 年版本包括 5.7 亿个实体，超过 180 亿个事实^[43]。2014 年，Google 又发布了另一款知识库 Knowledge Vault^[44]，其中包含 16 亿事实。Knowledge Graph 与

⁴<http://wiki.dbpedia.org/About>

⁵“Transferring Data to Wikidata and Shutting Down.” Google+. Dec 16, 2014.

⁶<http://www.geonames.org/>

Knowledge Vault 的主要区别在于 Knowledge Graph 通过从 Freebase、Wikipedia 等多种可信众包知识源抽取信息，而 Knowledge Vault 的信息来源则涵盖整个网络，既包含可信的结果，也包括低可信度的噪声，通过机器学习方法将他们排序呈现。

7. Satori

Satori 是微软集成于其 Bing 搜索引擎中的知识库，类似于 Google 的知识图谱，随着时间的推移，Satori 将能够利用数十亿实体内容及关系不断学习、增长知识，并逐渐了解整个 Web。微软同时将 Satori 延伸到其在线问答系统、搜索引擎优化和广告推荐中。

2.4.2. 维基百科页面分类

作为一个庞大的百科型知识库，维基百科具有完整而又复杂的页面分类体系。了解它的结构和特性有助于我们对其进行深入的认识和挖掘。下面详细介绍维基百科页面分类：

条目页面：条目（entry），即维基百科文章（Wikipedia article），指维基百科上所有的“百科全书式”文章以及目录索引，是维基百科全书的最基本组成单元（如图 2-2 a）。合格的条目应基于可靠来源就某一百科全书主题进行系统概括，并表现该主题与其他相关主题的联系。文章中除了对实体非结构化的文字描述之外，还包括以表格方式呈现的类别标签（Category）及信息框（Infobox）。信息框通过预设模板，可以反映实体的结构化自然属性，比如生日、位置、配偶等信息。条目属于维基百科各类页面中的主要名字空间（main namespace），也被称为文章空间或简称主空间。主要名字空间作为维基百科的默认名字空间，不包含前缀，区别于非主要名字空间的页面。在条目名字空间中，并非所有页面皆作为条目出现，也不计入统计。

非条目页面：条目页面之外的其他页面被称为非条目页面（Non-article pages），其中包括特殊的主要名字空间下的页面：重定向页（Redirect pages）、消除歧义页（Disambiguation pages）、部分长度过短的文章以及其他非主要名字空间的页面，诸如条目的对话页（Talk:）⁷、分类页面（Category:）、主题页面（Portal:）、模板页面（Template:）、模块页面（Module:）、文件描述页（File:）、用户页（User:）、用

⁷括号内为前缀。

户对话页（User talk:）、维基百科页（Wikipedia:）、帮助页面（Help:）、特殊页面（Special:）、MediaWiki 页面（MediaWiki:）等以及他们的对话页。

我们希望候选实体词典能够尽可能多的捕捉实体的不同别名，因此在提取维基百科信息的时候，主要处理的对象包括条目页面、重定向页、消歧义页面以及其他超链接。

重定向页：是一种特殊的页面，它提供一种运作机制，使得人们在输入实体别名进入条目或者点击指向该条目的内部链接时，系统能够自动导航到重定向页面内部指定的有关条目页面中，从而实现相关条目页面可以以多个名称进行访问。例如，如果设定了名称为“澳洲”，而内容指向“澳大利亚”的重定向页之后，任何人都可以通过检索或点击“澳洲”这一名称进入到澳大利亚条目中⁸。参见图 2-2 b, c。

消除歧义页：是维基百科中一种解决一词多义冲突的机制。当用户给出的关键词具有明显歧义，且指向多个项目的时候，维基百科会引导用户首先进入消除歧义页面进行选择，比如“Michael Jordan”既可以表示美国篮球运动员，也可以表示英国足球运动员、英国赛车手、机器学习教授，甚至是一首歌的名字。参见图 2-2 d。

超链接：维基百科条目中富含大量的将实体别名指向某一条目的超链接（锚文本），这些链接经人工编辑，可信度较高，且富含对于词语的语义理解。换而言之，链接的锚文本就是链接目标实体的一种表达。

⁸关于在危急中使用重定向的方针政策，请参看<http://en.wikipedia.org/wiki/Wikipedia:redirect>

Michael Jordan

From Wikipedia, the free encyclopedia

For other people named Michael Jordan, see Michael Jordan (disambiguation)."Air Jordan" redirects here. For the shoe and athletic wear company, see Air Jordan.

Michael Jeffrey Jordan (born February 17, 1963), also known by his initials, **MJ**,²⁷ is an American former professional basketball player, entrepreneur, and principal owner and chairman of the Charlotte Hornets. He played 15 seasons in the National Basketball Association (NBA) for the Chicago Bulls and Washington Wizards. His biography on the NBA website states: "By acclamation, Michael Jordan is the greatest basketball player of all time."²⁸ Jordan was one of the most effectively marketed athletes of his generation and was considered instrumental in popularizing the NBA around the world in the 1980s and 1990s.²⁹

A four-time NBA regular season Most Valuable Player (MVP) and five-time NBA Finals MVP, he won six NBA championships during his 15-year career. In the 1980s, when he was a member of the "Tutti Frutti" Central Division-winning Chicago Bulls, Jordan proved the NBA's Cleaning Rules in 1984. He quickly earned a reputation as a league star, entertaining crowds with his prolific scoring. His leaping ability, illustrated by performing slam dunks from the free throw line in air dunk contests, earned him the nicknames "Air Jordan" and "His Airness". He also gained a reputation for being one of the best defensive players in basketball.³⁰ In 1991, he won his first NBA championship with the Bulls, and followed that achievement with titles in 1992 and 1993, becoming a "three-peat". Although Jordan abruptly discontinued his basketball career in 1993, he returned to the NBA in 1995, winning a second title with the Chicago Bulls in 1996 and then to three additional championships in 1996, 1997, and 1998, as well as an NBA-record 72 regular-season wins in the 1995–96 NBA season. Jordan retired for a second time in 1999, but returned for two more NBA seasons from 2001 to 2003 as a member of the Wizards. Jordan's individual accolades and accomplishments include five Most Valuable Player (MVP) Awards, ten All-NBA First Team designations, nine All-Defensive First Team honours, twelve NBA All-Star Game appearances, three All-Star Game MVP Awards, ten scoring titles, three NBA抢断王 titles, and three NBA抢篮板王 titles. In 2009, he was inducted into the Naismith Memorial Basketball Hall of Fame. Jordan holds the NBA records for highest career regular season scoring average (30.12 points per game) and highest career playoff scoring average (33.45 points per game). In 1989, he was named the greatest North American athlete of the 20th century by ESPN, and was second to Babe Ruth on the Associated Press's list of athletes of the century. He is a two-time inductee into the Basketball Hall of Fame – in 2009 for his individual career, and in 2010 as a member of the 1995 United States men's Olympic basketball team ("The Dream Team").

Jordan has been inducted into the Naismith Memorial Basketball Hall of Fame twice.

He founded the business of Nike Air Jordan sneakers, which were introduced in 1985 and became a popular fixture.³¹ Jordan also starred in the 1996 feature film Space Jam as himself. In 2006, he became part owner and head of basketball operations for the then-Chicago Bulls, buying controlling interest in 2010. In 2015, Jordan became the first athlete in history to have a following foundation for children.³²



Michael Jordan in April 2006

Wikimedia Commons

Mj

From Wikipedia, the free encyclopedia

Redirect page**MJ**

- This is a redirect from a title with another method of capitalisation. It leads to the title in accordance with the Wikipedia naming associated in some way with the conventional capitalisation of this redirect title. This may help writing, searching and international link.

- Use this root to tag only mainspace redirects; when other capitalisations are in different namespaces, use {{R| from modifica

This page was last modified on 5 September 2012, at 08:37.

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy non-profit organization.Privacy policy About Wikipedia Disclaimers Contact Wikipedia Developers Mobile view

(a)

MJ

From Wikipedia, the free encyclopedia

(Redirected from Mj)**MJ** may refer to:

- Manufacturers' Junction Railway's reporting mark
- Líneas Aéreas Privadas Argentinas' IATA code
- Jeep Comanche or MJ, a pickup truck
- Megapoule (MJ), or millijoule (mJ), units of energy
- MJ (New York City Subway service), a defunct New York City subway service
- M_J, Jupiter mass, a unit of mass
- Master of Jurisprudence, a graduate law degree
- MJ, citation abbreviation for West's Military Justice Reporter
- Marc Jacobs (born 1963), luxury brand from LVMH Group
- Maotaijing, a Chinese strategy game

People [edit]

- Michael Jackson (1958–2009), American recording artist, entertainer, and businessman
- Michael Jordan (born 1963), former American professional basketball player, and current owner and chairman of Charlotte Hornets
- MJ Hibbett (born 1970), English guitarist singer-songwriter

Fictional [edit]

(c)

(b)

Michael Jordan (disambiguation)

From Wikipedia, the free encyclopedia

Michael Jordan (born 1963) is an American basketball player.**Michael Jordan** may also refer to:

- Michael Jordan (mycologist), English mycologist
- Michael Jordan (footballer) (born 1986), English goalkeeper (Arsenal, Chesterfield, Lewes)
- Michael Jordan (insolvency baron) (born 1931), English businessman
- Mike Jordan (1958), English racing driver
- Mike Jordan (baseball) (1863–1940), baseball player
- Michael Jordan (Irish politician), Irish Farmers' Party TD from Wexford, 1927–1932
- Michael B. Jordan (born 1987), American actor
- Michael I. Jordan (born 1977), American researcher in machine learning and artificial intelligence
- Michael H. Jordan (1936–2010), American executive for CBS, PepsiCo, Westinghouse
- Michael-Hakim Jordan (born 1995), American professional basketball player
- Michael Jordan (born 1990), Czech ice hockey player
- "Michael Jordan", a song by Kendrick Lamar featuring Schoolboy Q on the album *Overly Dedicated*

See also [edit]

- Michael Jordan statue, Chicago statue of the basketball player Michael Jordan
- Jordan (name)

(d)

图 2-2 维基百科页面示例

(a) Michael Jordan 的条目页面; (b) Mj 重定向到 MJ 的重定向页面; (c) MJ 的重定向页面;

(d) Michael Jordan 的歧义页面

2.5 候选实体词典构建方法

根据 2.2 的介绍，在候选集生成模型中，对于每一个实体的指称项 $m_j^i \in M$ ，实体链接系统尽可能多的找到该指称项指代的候选实体 E_j^i 。候选实体的生成方法

主要依赖于实体指称项字面和知识库中实体名称的字符串匹配。根据 Hachey 等人^[45]的研究，候选实体词典的质量与消歧算法同等重要，它的准确性、覆盖率和体量直接决定了后续消歧算法的准确率。

目前主流实体链接方法的候选集通过学习知识库获得。根据上一章对于维基百科的介绍，可以了解到维基百科积累了规模庞大的知识，直到今天仍在迅速增长和更新^[46]，其丰富的语义资源^[47]，包括词语的重定向、消歧义、链接关系，可以为候选实体词典构建提供丰富的实体别名。Mihalcea^[48]，Milne^[31]等通过对维基百科进行

锚文本挖掘，找到以查询词作为锚文本的字面词（surface name），然后将其对应的目标实体列为候选实体。Bunescu^[1], Cucerzan^[11]通过对包含锚文本在内的维基页面特征进行分析，得到字面词和目标实体页面作为提及-实体对（mention-entity pair）^[1, 11]，并通过建立提及到实体的倒排索引得到候选实体词典。

除了完全匹配，针对不同的应用场景，一些研究^[2, 6-8, 49, 50]使用部分匹配以提高召回率，Lehmann^[39]等使用模糊匹配提高精却匹配概率。Chakrabarti , Cheng 等人^[49, 51, 52]利用了用户点击记录和网络文档来发现实体别名。相似的方法还有利用查询扩展^[53]降低提及的模糊性，包括 Cucerzan^[11]和 Varma^[50]在文档内查找长名称（long name）代替短名称（short name）以及 Zhang 等使用分类器筛选那些潜在的缩写全称。Guo^[54]等人结合查询扩展和基于规则的候选实体过滤，平衡了候选实体集合的召回率和候选实体数目。一些其他的相关工作还包括在与词典进行匹配之前，针对实体提及拼写错误的问题，Varma^[50]等人利用变音位算法（metaphone algorithm^[55]）来识别提及的拼写变化，Chen 等人^[4]利用朗讯公司（Lucene）的 spellchecker 进行拼写检查，以及利用 Google 提供的拼写纠正服务来解决实体提及中的拼写错误问题^[5, 15, 50]。上述方法结合了现有技术，在一定程度上对实体词典的准确性和全面性有所改善，但更多的研究表明仅通过知识库特征提取就能够保证候选实体词典的完备性，结合现有研究能力，本文使用传统方法进行知识库构建，3.5.1 中实验表明本文候选实体词典质量与当前研究水平一致，为后续实体消歧工作提供了知识保障。

2.6 实体链接文本处理方法

2.6.1. 字符串比较

独立特征中直接对提及和实体的名称进行比对是一种最简单直接的方法，字符串的比较特征包括：

1. 提及与实体名称是否严格匹配
2. 提及与实体名称是否正（反）向最大匹配
3. 实体名称是否是提及的前（后）缀
4. 提及是否是实体名称的子串
5. 提及中所有字符是否依顺序被实体名称包含

6. 字符数是否一致

常用的比较字符串相似度的方法包括编辑距离(edit distance)^[5]、Dice 相似性系数(Dice coefficient score)^[6]、汉明距离(Hamming distance)^[7]等。

编辑距离是一种常用的计算字符串相似度的方法，通过对从一个字符串向另一个字符串转变的最少操作数进行计数来得到。根据操作方式的不同，编辑距离又有很多定义，其中最常见的一种叫 Levenshtein 距离。Levenshtein 允许三种操作，即插入、替换和删除。下面给出正式定义。

基于字母表 Σ （如 ASCII 字符）给定两个字符串 a 和 b ，他们的编辑距离 $d(a,b)$ 定义为从 a 转换到 b 的最小操作序列^[56]。其中插入操作表示成 $\varepsilon \rightarrow x$ ，删除表示成 $x \rightarrow \varepsilon$ ，替换表示成 $x \rightarrow y, x \neq y$ 。其中 ε 表示空字符。例如“kitten”到“sitting”的编辑距离为 3，通过如下方式计算得到：

1. kitten → sitten: $k \rightarrow s$
2. sitten → sittin: $e \rightarrow i$
3. sittin → sitting: $\varepsilon \rightarrow g$

Dice 相似性系数用来比较两个样本的相似度。给定两个字符串 a 和 b ，他们的 Dice 相似性定义为：

$$Sim_{Dice} = \frac{2n_t}{n_a + n_b} \quad (2-1)$$

其中 n_t 表示 a 和 b 共享的双连词(Bigram)数， n_a, n_b 分别代表 a 和 b 包含的双连词数。

比如“night”和“nacht”各自的二连词数组表示为： $\{ni, ig, gh, ht\}$ 和 $\{na, ac, ch, ht\}$ ，二者共享一个而连词“ht”，Dice 相似性为 $2*1/4+4=0.25$ 。

汉明距离通过统计对应位置字符的一致性来比较等长字符串的相似性。如“karolin”和“Kathrin”第三四五位不同，因此其汉明距离为 3。

2.6.2. 向量空间模型

向量空间模型对于任意提及，将其全部上下文^[2, 4, 9, 20, 22]表示成一个无序的词的集合（词袋，bag of words），或者设定一个窗口阈值^[1, 3, 12, 13]选择其部分上下文。例

如下面两篇文章：

Passage A: “John likes to watch movies. Mary likes movies too.”

Passage B: “John also likes to watch football games.”

基于其文本可以得到词典：

Dictionary={"John": 1, "likes": 2, "to": 3, "watch": 4, "movies": 5, "also": 6, "football": 7, "games": 8, "Mary": 9, "too": 10}

从而利用词典中的序数对文章重新编码，得到两个十维向量：

Passage A: {1,2,1,1,2,0,0,0,1,1}

Passage B: {1,1,1,1,0,1,1,1,0,0}

其中每一维上的数字代表该词在文章中刚出现的频数。除此之外，也可以用该词在文档中的 tf-idf 来代替。Bagga 等人早在 1998 年就是用向量空间模型来表示文本，并利用余弦相似度来衡量文本之间的相似性，从而区分不同文章中出现的指称项是否对应同一个实体。考虑到向量空间在文本表示中维度高且稀疏的问题，本文在第三章利用基于朴素贝叶斯的加权相似度作为特征，从而将上下文特征降维至一维。

2.6.3.余弦相似度

余弦相似度用来衡量两个向量的相似性，给定两个向量 A 和 B ，它们的余弦相似度 $\cos(\theta)$ 表示为：

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}} \quad (2-2)$$

余弦相似度的取值范围从 -1 (完全相反) 到 1 (完全相同)，0 代表二者正交(相互独立)。在计算文本相似度的时候，余弦相似度将文本向量长度正则化成长度为 1 的向量，因此文件的长度对相似度的影响不大，而词汇的权重影响较大。常用的权重计算方式为词频与逆文档频率的乘积，亦即 $TF \times IDF$ 。

2.6.4.TF-IDF 权重

TF-IDF 是一种常用的信息检索加权技术,用以评估一个字词在语料库中的重要程度^[57]。字词的重要性随着他在文件中出现的次数成正比增加,但同时会随着它在语料库中出现的频率成反比下降。*TF-IDF* 实际上是: $TF \times IDF$, 即 *TF* 词频 (Term Frequency) 与 *IDF* 逆文档频率 (Inverse Document Frequency) 的乘积。其中词频定义为:

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (2-3)$$

其中 $n_{i,j}$ 表示在文档 d_j 里词语 t_i 出现的次数, 分母表示文件 d_j 中所有次出现的次数总和。

逆文档频率用来衡量词语的普遍重要性, 可以由文件总数与包含该词语的文件数之比的对数来表示:

$$IDF_i = \log \frac{|D|}{1 + |\{j : t_i \in d_j\}|} \quad (2-4)$$

其中 $|D|$ 表示语料库中总文件数, $|\{j : t_i \in d_j\}|$ 表示包含词语 t_i 的文件数。

综上, 文档 d_j 中词语 t_i 的 *TF-IDF* 权重可以表示成:

$$TFIDF_{i,j} = TF_{i,j} \times IDF_i \quad (2-5)$$

对于 *TF-IDF* 可以如此理解: 如果某个词或短语在一篇文章中出现的频率高 (*TF* 高), 并且在其他文章中很少出现 (*IDF* 高), 则认为此词或者短于具有很好的类别区分能力。

2.6.5.朴素贝叶斯分类器

朴素贝叶斯分类器是一种应用基于独立假设贝叶斯定理的概率分类器。理论上, 概率模型分类器是一个条件概率模型:

$$p(H | F_1, \dots, F_n) \quad (2-6)$$

独立的类别变量 H 存在若干类别, 条件依赖于若干特征变量 F_1, F_2, \dots, F_n 。但如果特

特征数 n 较大或者每个特征能取大量值的时候, 基于概率模型列出概率表变得不现实, 根据贝叶斯定理:

$$p(H|F_1, \dots, F_n) = \frac{p(H)p(F_1, \dots, F_n | H)}{p(F_1, \dots, F_n)} \quad (2-7)$$

其中分母部分不依赖于 H 且 F_i 给定, 因此分母可以认为是一个常数。而对于分子部分, 则可以表示成联合概率分布:

$$p(H|F_1, \dots, F_n) \propto p(H)p(F_1|H)p(F_2|H, F_1)\dots p(F_n|H, F_1, F_2, \dots, F_{n-1}) \quad (2-8)$$

由于假设特征之间相互独立, 上式可以表示成:

$$\begin{aligned} p(H|F_1, \dots, F_n) &\propto p(H)p(F_1|H)p(F_2|H)\dots p(F_n|H) \\ &\propto p(H)\prod_{i=1}^n p(F_i|H) \end{aligned} \quad (2-9)$$

从而进一步得到朴素贝叶斯分类器的决策公式:

$$h_{NB} = \arg \max_{h \in H} p(F|h)p(h) = \arg \max_{h \in H} p(h)\prod_{i=1}^n p(F_i|h) \quad (2-10)$$

朴素贝叶斯分类器可以应用于文本上下文相似度度量, 即寻找与指称项上下文文本相似性最大的候选实体, 可以类比成一个朴素贝叶斯分类问题^[58]。其中候选实体集合对应朴素贝叶斯分类器的类别, 指称项所在的上下文向量空间对应朴素贝叶斯分类器的待分类文本, 如此上述任务就转化为将指称项上下文中每一个词分到某一个候选实体的问题。

2.7 实体链接效果评价

本节首先介绍实体链接系统的评价方法, 接着介绍 TAC-KBP 评测竞赛的相关内容。

2.7.1. 实体链接的评测方法

当前主要的实体链接效果评价方式包括准确率、召回率、F1 值以及 B^3+ 评价, 其中准确率评价最为常用。

准确率: 是实体链接任务中系统标注正确的结果占查询实例集合的比例, 评价

要求测试集提供提及对应的准确实体（或 NIL），准确率具体分三类：

1.全部准确率 (All-Accuracy): 即全部查询实例的准确率。该指标用来定量衡量实体链接的综合表现，定义如下：

$$Accuracy^{All} = \frac{\sum_{q \in Q} G(q)}{|Q|} \quad (2-11)$$

式中 Q 为查询实例集合， $G(q)$ 是评价积分，定义为：

$$G(q) = \begin{cases} 1, & E_g(q) = E_s(q) \\ 0, & \text{其他} \end{cases} \quad (2-12)$$

式中 $E_g(q)$ 和 $E_s(q)$ 分别表示查询 q 的目标实体和系统返回的结果。

2.NIL 准确率 (NIL-Accuracy): 即目标实体在知识库外的查询准确率。用来衡量消歧算法识别空实体的能力。定义如下：

$$Accuracy^{NIL} = \frac{\sum_{q \in Q_{NIL}} G(q)}{|Q_{NIL}|} \quad (2-13)$$

式中 Q_{NIL} 为真值为 NIL 的查询集合，即 $Q_{NIL} = \{q \in Q \cap E_g(q) \notin \kappa\}$ ，其中 κ 是知识库中实体集合。

3.InKB 准确率 (InKB-Accuracy): 即包含在知识库中的目标实体的查询准确率。用来衡量消歧算法的在有条件找到目标实体的条件下的消歧能力。定义如下：

$$Accuracy^{InKB} = \frac{\sum_{q \in Q_{InKB}} G(q)}{|Q_{InKB}|} \quad (2-14)$$

式中 Q_{InKB} 为真值非空的查询集合，即 $Q_{InKB} = \{q \in Q \cap E_g(q) \in \kappa\}$ ，其中 κ 是知识库中实体集合。

召回率： 召回率反映了实体链接算法的查全率，定义如下：

$$Recall = \frac{\sum_{q \in Q} G(q)}{|L|} \quad (2-15)$$

其中 L 为所有应该被链接到实体的提及的集合， $\sum_{q \in Q} G(q)$ 表示所有链接正确的查询数。

F1 值： 综合考虑准确率和召回率，F1 值可以用来综合评价链接系统的性能，

其定义如下：

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \quad (2-16)$$

B³+评价：通过对查询实例进行聚类来评价实体链接效果。如果查询实例对应相同的实体，则这些查询实例应该被聚为一类，即被标记为相同的目标实体；同理，如果查询实例对应 NIL，则所有 NIL 型查询实例应该被聚为一类，被标记一个唯一的标识。B³+具体按照如下规则进行。首先两两比较查询实例，如果实体链接结果和真值一致，则对这一对结果奖励 1 分，否则不奖励。积分公式如下：

$$G(q_i, q_j) = \begin{cases} 1, & L(q_i) = L(q_j) \wedge I(q_i) = SI(q_i) = GI(q_j) = SI(q_j) \\ 0, & \text{其他} \end{cases} \quad (2-17)$$

其中 $L(q)$ 表示查询实例 q 在真值中的类别， $C(q)$ 表示实体链接算法给出 q 的类别， GI 和 SI 分别表示真值和链接结果。B³+的准确率和召回率分别表示为：

$$precision = \frac{\sum_{q_i \in Q} \frac{\sum_{q_j \in Cq_i} G(q_i, q_j)}{|C(q_i)|}}{|Q|} \quad (2-18)$$

$$recall = \frac{\sum_{q_i \in Q} \frac{\sum_{q_j \in Lq_i} G(q_i, q_j)}{|L(q_i)|}}{|Q|} \quad (2-19)$$

2.7.2. 实体链接相关评测竞赛

TAC KBP (Knowledge Base Population, 知识库扩充) 是目前国际上最具影响力的实体链接评测比赛之一，由 NIST 于 2009 年在其主办的 TAC 会议上提出。实体链接任务是 KBP 的子任务之一，从 2009 到 2013，NIST 共组织举办了五届实体链接评测竞赛。KBP 使用 2008 年英文版维基百科作为知识库，从其中包含 818741 条实体（具体类别及比例见表 2-1，分别抽取了实体名称及部分描述性文字。图 2-3 展示了该知识库中一个实体的例子，可以看出 KBP 知识库中的实体以 XML 格式存储，分为结构化信息 (facts) 及非结构化信息 (wiki_text)，其中结构化信息主要来自 infobox，非结构化信息来自页面正文。

用于评测的标注数据来自一个大语料库中抽取的部分文档，语料库主要来自博客文本和新闻文章，表 2-2 给出近几年 KBP 评测中查询实例的具体组成情况，我们注意到评测数据中 NIL 查询的数量约占总查询实例数的 50%左右，可以看出 KBP 评测关注非知识库实体的发现，而在实体分类这个层面上，各类别实体的比例趋于平衡，2009 年之后，数据来源中增加了博客文本，这一项的比例也稳定在三成左右。在查询任务中，每个查询实例包含查询名称及所在的查询文档标识。评测规则要求参评系统需给出查询名称在相应查询文档中对应的实体在评测知识库中的实体 ID，若知识库中不存在对应的实体，则返回 NIL。图 2-4 给出一个查询实例及其对应的查询文档的例子。

表 2-1 TAC KBP 知识库中实体分类情况

类别	缩写	数量	百分比
人物实体	PER	11 万	14%
机构实体	ORG	5.5 万	6.8%
地理实体	GPE	11 万	14.2%
其他类别实体	UKN	53 万	65%

表 2-2 TAC KBP 评测数据纵览

年份	查询实例#	查询类型		实体分类			数据来源	
		InKB#	NIL#	PER#	ORG#	GPE#	新闻#	博客#
2009	3904	1675	2229	627	2710	567	3904	0
2010	2250	1020	1230	751	750	749	1500	750
2011	2250	1124	1126	750	750	750	1491	759
2012	2226	1177	1049	918	706	602	1471	755

实体ID, 名称	<pre><entity wiki_title="Abbott_Laboratories" type="UKN" id="E0272065" name="Abbott Laboratories"> <facts class="Infobox_Company"> <fact name="company_name">Abbott Laboratories</fact> <fact name="company_type"><link>Public</link> (<link entity_id="E0231100">NYSE</link> :? ABT)</fact> ... <fact name="homepage">www.abbott.com</fact> </facts> <wiki_text><![CDATA[Abbott Laboratories Abbott Laboratories (NYSE:?) is a diversified pharmaceuticals health care company. It has 68,000 employees and operates in 130 countries. The corporate headquarters are in Abbott Park, Illinois, located near North Chicago, Illinois. ...]]></wiki_text> </entity></pre>
结构化信息	
非结构化信息	

图 2-3 TACKBP 知识库实体实例

查询名称	<pre><query id="EL29"> <name>Abbott Laboratories</name> <docid>AFP_ENG_20070326.0058.LDC2009T13</docid> </query></pre>
查询文档ID	
查询文档	<pre><DOC> <DOCID> AFP_ENG_20070326.0058.LDC2009T13 </DOCID> <DOCTYPE SOURCE="newswire"> NEWS STORY </DOCTYPE> <DATETIME> 2007-03-26 </DATETIME> <BODY> <HEADLINE> Thai AIDS activists rally against US firm over generic drug row </HEADLINE> <TEXT> <P> Some 100 AIDS activists on Monday rallied outside the Bangkok office of US pharmaceutical giant Abbott Laboratories over its decision to withdraw new medicines from Thailand amid a generic drugs row. </P> ... </TEXT> </BODY> </DOC></pre>

图 2-4 TAC KBP 查询实例及对应查询文档

2.8 本章小结

本章详细介绍了实体链接任务的相关理论和方法。实体链接任务本质上是根据给定查询实例通过对候选实体排序消歧，并返回目标实体的过程。微博文本作为本文主要研究对象，主要特点包括文本长度有限、具有用户性、文本中包含丰富的关联信息。知识库对于实体的描述能力和覆盖率对实体链接算法有很大的影响，文本使用维基百科作为知识库。实体链接任务主要分两个步骤，构建候选集词典和实体消歧。本章还介绍了包括特征相似度计算、朴素贝叶斯分类器在内的常用实体链接相关文本处理方法。最后，对实体链接任务通常使用准确率、召回率、F1 值以及 B^3+ 等实体链接任务链接效果的评价方式进行了介绍，并概述 TAC-KBP 评测竞赛中实体链接子任务的内容和发展。本章对于实体链接相关理论与方法的介绍，将为全文研究工作提供理论和方法上的指导。

第三章 面向微博文本的预处理方法

3.1 引言

预处理是实体链接的重要环节，包括构建候选实体词典与特征提取两部分内容。其中候选实体词典由知识库中提取的“实体-提及”对经排序整理得到，候选实体词典收词完备性直接决定了消歧算法的准确率的上限。而特征的选择^[59]和提取则影响到消歧算法对于查询实例与候选实体之间相关性的理解。

本章首先根据传统的实体词典构建方法，具体介绍词典的构建过程。进而介绍传统实体链接预处理中常用的若干方法，包括字符串相似度、实体流行度、文本上下文相似度以及实体相关性特征。目前的实体链接方法使用的特征可以分为上下文无关特征与上下文相关特征，其中上下文无关特征（诸如实体流行度）不受文本长度的影响，因此该特征在微博短文本中依然有效⁹。然而由于微博文本篇幅较短¹⁰，不能为消歧算法提供充足的上下文相关信息，因此上下文相关特征在基于微博短文本的实体链接任务中的重要性减弱。本章针对传统实体链接算法预处理过程中存在的问题，对现有流行度特征提取方法进行改进，并结合微博文本特点，引入更具有分辨力的临近上下文特征以及基于朴素贝叶斯的文本上下文相似度计算方法，以全面提高面向微博的实体链接算法预处理能力，为下一步消歧算法的准确率提供保障。

在文章结构上，本章在第二节首先介绍本文候选实体词典构建方法；接着在第三节阐述传统实体连接预处理常用方法及特征；针对传统方法存在的不足，第四节详细介绍本文面向微博文本的预处理过程及对预处理方法的优化改进，最后结合实验结果分析论证本文改进方法的有效性。

⁹实际上由于受到流行度统计方式的影响，传统的基于知识库的实体流行度并不能客观的反映实体在微博文本中的流行度情况，具体讨论见第六章

¹⁰ Twitter中规定一篇微博最大长度为140字符

3.2 候选实体词典构建

目前候选实体词典的构建方法较为成熟，本文在构建候选实体词典的过程中参考前人工作，使用 2014 年 3 月 4 日的维基百科作为数据源，综合利用实体页面、重定向页面、消除歧义页面、姓氏页面以及页面中 Infobox 和锚文本信息，得到收词量逾千万的实体候选词典。下面介绍具体构建过程：

1. 提取全部条目页面：根据 2.4.2 我们首先从 wiki 原始 xml 中提取全部条目页面（namespace=0）共计 10835922 条；

2. 提取重定向页面：从 enwiki-20140304-redirect.sql 中提取重定向页面数据 7202728 条，结合条目页面进行过滤筛选，得到 6138063 条重定向记录；

3. 提取歧义页面：从 enwiki-20140304-categorylinks.sql 中提取歧义页面数据 153458 条，并筛选得到歧义实体 150003 条；

4. 提取姓氏页面：人名是实体消歧任务中主要的研究对象之一，维基百科中使用 Surname 页面（见图 3-1）记录人名的姓氏及对应的实体条目，我们从 enwiki-20140304-categorylinks.sql 中提取 Surname 页面数据，与已知的重定向页面进行交叉比对过滤，得到 34553 条 id 记录。结合已获得的条目页面信息，去掉未知条目，得到 33537 条姓名歧义数据；

5. 提取主要页面标题：显然，实体在条目页面中的标题也是实体的一个别名，因此我们从第一步全部条目页面中去掉重定向页和歧义页，得到所有普通页面标题共 4541623 条；

6. 锚文本提取：利用 annotated wikiextractor¹¹工具提取所有页面的链接和纯文本内容，得到锚文本信息 71219387 条，去掉部分链接不到的页面或该页面不被包含的情况，并经过转码，共得到有用数据 69299307 条；

7. Infobox 信息提取：Infobox 中含有包括实体别名在内的多种实体自然属性。我们通过 wikixmlj 提取到共 2204978 个页面的 Infobox 信息，通过解析得到 146224 条别名信息。

¹¹ Joachim Daiber (jo.daiber@fu-berlin.de) Version: 0.1 (Jan 26, 2010)

The screenshot shows the Wikipedia article for "Jordan (name)". The page title is "Jordan (name)". The main content discusses the name's etymology from Hebrew, Latin, Arabic, Spanish, Portuguese, Dutch, French, Irish, Romanian, and Catalan. It notes that "Jordan" is also used as a geographical name, such as the Jordan River. A sidebar on the right provides detailed information about the name, including its gender (Unisex), origin (Hebrew), word/name (Jordan), meaning (Descend or flow down), and other names (Jorden, Jourdan, Jordanus, Jordán, Jordy, Jordán, Jordáin, Jordy, Jordáin). Below the sidebar is a link to Wiktionary.

图 3-1 Jordan 的姓氏页面

8. 提取提及-实体对：对于每个实体，分别从主要页面标题、重定向页面、歧义页面、Infobox 别名中查找其对应的别名，形成实体到提及的映射表，并将锚文本挖掘结果追加其中总计得到 13174673 条词条。进而建立倒排索引并统计实体出现频率，得到候选实体词典；

得到候选实体词典，就完成了实体链接的第一个重要环节，对于输入的每个提及，首先会检索候选实体词典，如果词典中不包含该提及，或者该词条下无候选实体，则实体链接算法会返回 *NIL*，否则将会把提及以及所对应的候选实体传递给下一步实体消歧算法进行进一步的处理。

3.3 传统实体链接预处理方法

由于实体链接任务输入的约束，消歧方法的特征选择只能通过查询词和相关文档中获得，常用的特征可以分为上下文独立的特征与上下文相关特征，其中上下文独立特征包括实体流行度，上下文相关特征包括文本上下文和实体相关性。

3.3.1. 实体流行度提取

实体流行度可以反映一个提及被链接到该候选实体的先验概率，反映了实体的热门程度，是一种被广泛使用的特征。通过实验观察，对于同一个提及 m_j^i 的不同候选实体，其流行度可能差别很大，有些十分常见，有些则很生僻，例如“AK47”表示一种武器的概率要远高于表示一款鸡尾酒，Michael Jordan（篮球运动员）出现的概率要高于 Michael Jordan（足球运动员），二者又远高于 Michael Jordan（Berkeley 教授）。目前主流的提取实体流行度的方法是基于对维基百科的统计信息，对于某待消歧的命名指称项 m ，其备选实体集合为 E_m ，对于其中的备选实体 $e_i \in E_m$ ，其实体流行程度定义如下：

$$Pop(e_i) = \frac{count_m(e_i)}{\sum_{e_j \in E_m} count_m(e_j)} \quad (3-1)$$

其中 $count_m(e_j)$ 表示提及 m 指向实体 e_j 的次数。

除了实体频度，有些研究使用文本长度作为流行度衡量方法，规定描述页面的文本越长，其被关注的程度可能就越大。Guo^[9], Gattani^[60]等人利用维基页面访问统计信息来估计实体流行度。本文在 3.4.1 中提出基于均分词频的实体流行度，有效解决了词频统计中对重定向页面统计出现的问题。

由于大多数的实体指称项在日常用语中通常指代那些显著地实体，因此仅使用实体流行度作为排序特征既能够获得较好的链接结果。Ji 和 Grishman^[61]的实验表明仅利用实体流行度的实体链接系统的准确率可以达到 71%，我们在实验中（见 4.3.4）取得了类似的结果。由此可以看出，实体流行度是实体消歧重要且有效的特征之一。

然而目前主流的实体流行度提取方法简单的将锚文本的指向页面记为一次锚文本指称项的观察记录，考虑到维基百科丰富的页面类型，这种方法存在对于重定向页面和消歧义页面的重复统计问题，在一定程度上影响了实体流行度的质量。

3.3.2. 文本上下文相似度提取

上述独立特征虽然有一定的区分能力，但仅凭实体和提及名称提供的信息不足以实现更好的消歧效果，因此十分有必要进一步挖掘上下文特征，从语义层面上进

行消歧。最简单直观的一种利用上下文的方式是直接比较提及和实体所在文档的上下文的相似性。其中文档可以通过向量空间模型（Vector Space Model, VSM）、概念向量（Concept Vector）等来表征。对于每个实体，可以使用整个维基页面^[1, 3, 12, 22]、描述段^[12]、实体名称所在上下文窗口^[16]或者取文中所有词的 TF-IDF 最高的前 k 个词作为实体上下文^[9, 14]。

概念向量的组成包括关键的短语^[10]、锚文本^[12]、实体名^[23]、种类^[7, 11]、描述性标签^[60]、维基百科概念^[6, 11, 15, 30]等，此外实体的上下文也可以通过其相关文章、属性以及信息框中的信息^[4, 6, 7]来表示。Han 和 Sun^[27]利用一元语言模型对候选实体上下文建模，以表示该实体出现在特定上下文中的概率。主题模型^[62]也被用来对上下文潜在的主题分布进行建模^[2, 19, 20, 32, 63-65]。

文本上下文特征通常使用余弦相似度进行计算，本文在 2.6.3 节详细介绍了余弦相似度的原理及计算方法。试验中我们发现，目前的文本上下文相似度计算方法存在两个主要问题：首先，文本向量维度高且稀疏，不适用于实体消歧模型结合多种异构的特征综合计算的应用场景；另一方面，对篇幅较短或包含词典外词语的文本，无法得到其准确的余弦相似度，因此传统基于余弦相似度的文本上下文特征已经无法满足微博文本表示的需求。

3.3.3. 实体相关性提取

实体的上下文文本无疑提供了大量的语义相关性信息。根据 1.2 节实体消歧相关研究方法的综述，最新的研究^[9-19]开始利用实体相关性特征进行协同消歧，这类实体消歧任务基于“同一篇文档所提及的实体主题相关”这一假设。Cucerzan^[11]首先用实体类型一致性来定义实体相关性，Shen 等人^[15]将全局主题一致性纳入到其排序函数中，并在第 2013 年 SIGKDD 会议上进一步提出对微博间用户兴趣与微博内局部知识构建图模型以进行协同消歧（如图 3-2），此外 Han 等人^[13]、Hoffart 等人^[10]在研究中也将实体间相关性作为全局建模的重要特征。许多方法^[10, 12-16, 20]使用基于维基百科链接的实体相关性度量方法（Wikipedia Link-based Measure, WLM）^[31]来评价实体关联，该方法假设语义相近的两篇文档应该链接到更多的共同文档，利用归一化谷歌距离^[66]建模。

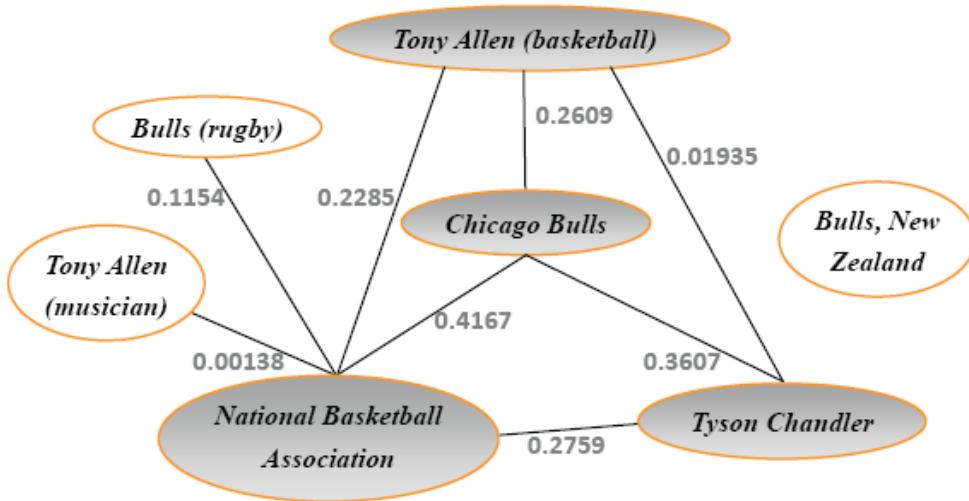


图 3-2 KAURI 算法图模型示例

给定两个维基百科实体 e_1, e_2 , 则它们之间的主题一致性定义成:

$$Coh_G(e_1, e_2) = 1 - \frac{\log(\max(|E_1|, |E_2|)) - \log(|E_1| \cap |E_2|)}{\log(|WP|) - \log(\min(|E_1|, |E_2|))} \quad (3-2)$$

其中, E_1 和 E_2 表示分别链接到 e_1 和 e_2 的文档集合, WP 表示维基百科全部实体集。 $Coh_G(e_1, e_2)$ 的取值范围为 [0.0, 1.0]。此外, Ratinov^[14]提出指向性互信息 (point-wise mutual information, PMI-like) 来衡量维基实体实体间的主题相关性:

$$Coh_p(e_1, e_2) = \frac{|E_1 \cap E_2| / |WP|}{|E_1| / |WP| \cdot |E_2| / |WP|} \quad (3-3)$$

Guo^[9]等人利用 Jaccard 距离来衡量实体间主题相关性:

$$Coh_J(e_1, e_2) = \frac{|E_1 \cap E_2|}{|E_1 \cup E_2|} \quad (3-4)$$

除了基于维基百科链接结构的方法, 最近 Ceccarelli^[67]提出利用学习排序模型来学习实体间的主题相关性, 该方法假设一个好的测量方法会增强正确链接实体之间的相关性。实验表明其方法通过组合 27 种测量方法, 得到了更好的结果, 但效率不及其他方法。此外, Han^[19]等人利用主题模型对实体相关性建模。南开大学沈玮^[18]综合利用了页面类型的相似度和分布式上下文相似度。

尽管实体间的相关性特征较文本上下文能够捕捉更多的主题层面的语义关联,

对实体链接效力有一定提升，但其存在的问题也显而易见：首先对于查询实例中的每一个提及的链接，都需要预先得到文档中其他提及所对应的实体，而这些实体的确定又依赖于其他相关实体，研究表明这是一个 NP 难的优化问题，增加了算法的时间复杂度，不利于实时任务；其次，实体相关性特征的使用完全依赖于通篇文档中实体主题相关这一假设，而在临近实体较少的微博文本中，这一假设的有效性有待进一步讨论。

3.4 面向微博文本的预处理方法

3.4.1 基于均分词频的实体流行度提取

实体流行度可以反映一个提及被链接到该候选实体的先验概率，反映了实体的热门程度，是一种被广泛使用的特征，实验表明^[61]实体流行度为实体消歧提供了较为有效的先验知识，多数的指称项指向少数高频实体。我们在上一节详细介绍了目前常用的实体流行度计算方法，本文的实体流行度计算也选择基于实体频度的统计方法，同时针对传统词频统计方法对重定向页面统计缺失或重复统计的问题，提出基于均分词频的实体流行度对传统特征提取进行改进。

在对维基百科的预处理的时候，我们参照传统方法通过遍历维基百科页面中的锚文本，得到锚文本及其对应的实体页面的映射，即命名指称项到命名实体的映射。但实际操作中我们发现，由于维基百科重定向和歧义页面机制的存在，许多链接文本并没有直接指向实体页面，而是重定向到歧义页面，如图 3-3 中“Mj”一词首先被重定向到重定向页“MJ”，“MJ”又有一条路径指向歧义页面“Michael Jordan(disambiguation)”，最终由歧义页面中的超链接文本“Michael Jordan”链接到篮球运动员 Michael Jordan 的实体页面。传统的实体流行度提取只关注那些能够链接到实体页面的链接文本，这样无疑中就丢失了很多维基百科中存在提及-实体信息。

我们为此改进了提取实体流行度的方法，提出均分词频的概念。在遍历知识库的过程中，程序会记录下从某一链接文本开始到指向某一实体页面结束的路径上经过的全部链接文本以及所有指向的实体页面，并平均分配这一个命名实体的指称项

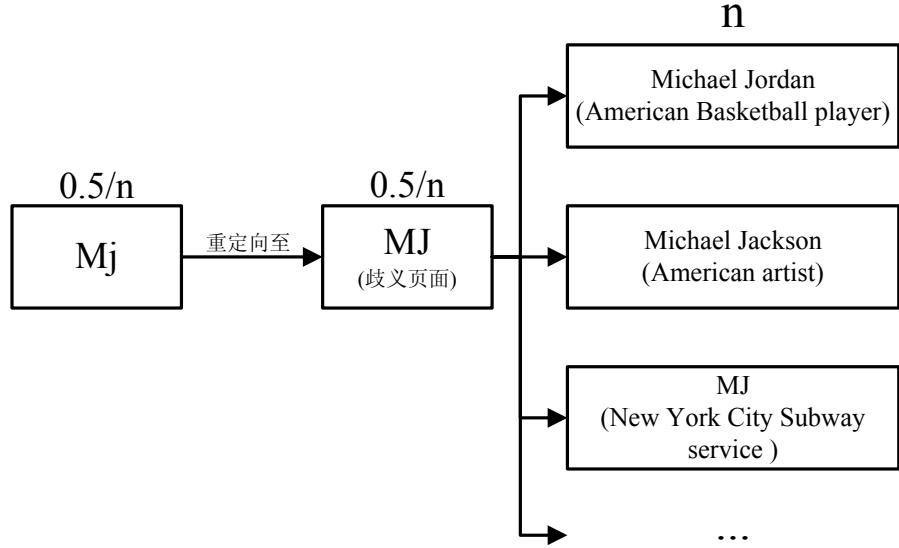


图 3-3 基于均分词频的流行度举例

以及其指向某一命名实体沿途经过的所有指称项链接到该实体的这一次观察记录。具体来讲，假设在某条链接路径 L 上，某指称项 m 经过 $k-1$ 次链接到达实体歧义页面 E ，其中包含 n 个实体即 $E = \{e_1, e_2, \dots, e_n\}$ ，则对于这条路径上的每一个提及-实体对，其在本次观察中的均分词频表示为

$$Freq_{div} = \frac{1}{n \cdot k} \quad (3-5)$$

均分词频可以理解为：所有重定向指称项，均分其链接到某一实体的这条观察记录。通过对每一个指称项的候选实体统计其均分词频，可以得到命名指称项到命名实体的映射词典及对应词频，进而可以得到基于均分词频的实体流行度：

$$Pop_{div}(e_{j,q}^i) = \frac{Freq_{div}(e_{j,q}^i)}{\sum_{e_{j,q}^i \in E_j^i} Freq_{div}(e_{j,q}^i)} \quad (3-6)$$

其中 m_j^i 为第 i 篇微博的第 j 条命名指称项， E_j^i 为 m_j^i 对应的候选实体集， $e_{j,q}^i$ 为 E_j^i 中的第 q 个候选实体， $Freq_{div}(e_{j,q}^i)$ 为 $e_{j,q}^i$ 的均分词频和。

利用均分词频可以有效地避免传统流行度提取方法中漏掉的大量有效词义信息，使我们算法中对于流行度的统计更为准确、客观、全面。

3.4.2. 基于朴素贝叶斯的文本相似度提取

文本上下文是实体消歧算法中最重要的语义特征之一，如果说上一节中提到的

流行度是基于先验知识的主观猜测，那么上下文特征才是真正从文本内容出发，通过“理解”语义来选择合适的候选实体。

根据 3.3.2 中介绍，文本上下文可以利用基于词典的向量空间模型去表征，并利用余弦相似度来衡量微博上下文与候选实体文本内容的相似程度。但由于我们的模型要结合多种异构的特征进行综合相似度的计算，向量空间在表示文本的时候维度过高且十分稀疏，不利于进一步的计算。因此我们考虑直接利用上下文相似度作为特征，从而将上下文特征降维至一维。

同时，对于本研究对象微博文本，有些长尾实体（如表 3-3 中命名指称项“NYC”的一个备选实体“John F. Kennedy International Airport”）在整个知识库中出现的频率较低，亦或是上下文信息不足，在计算余弦相似度的时候很容易出现相似度为零的情况，使算法丧失在长尾中进一步的分辨能力。因此，本文参考唐^[68]的方法，利用基于朴素贝叶斯的加权方法计算文本相似度。

寻找与指称项上下文文本相似性最大的候选实体上下文这一任务，可以类比成一个朴素贝叶斯分类问题^[58]，其中候选实体集合 $E = \{e_1, e_2, \dots, e_n\}$ 对应朴素贝叶斯分类器的类别 $H = \{h_1, h_2, \dots, h_n\}$ ，指称项 m 所在的上下文向量空间 $D(m) = \{d_1, d_2, \dots, d_k\}$ 对应朴素贝叶斯分类器（参见 2.6.5）的待分类文本，如此上述任务就转化为将 $D(m)$ 分到 E 中某一个类别的问题。

然而当我们把朴素贝叶斯分类器应用到相似文本度量的时候，会遇到类条件概率 $p(d_i | e)$ 为零的问题，实际上但凡 D 中出现新的词，由于所有类别中都不包含该属性值，就会导致条件概率为零，并进一步导致经过联乘后的未知数据样本 D 在所有分类下的后验概率 $p(D | e)$ 均为零。为了避免类似情况的发生，我们使用 m-estimate 来计算每个词在文档中出现的频率：

$$P(d_i | e_{j,q}^i) = \frac{n_q^{(k)} + 1}{n_j + v} \quad (3-7)$$

其中， $n_q^{(k)}$ 表示在第 i 条微博的第 j 个指称项对应的第 q 个候选实体 $e_{j,q}^i$ 的上下文词汇表中词 d_k 出现的次数， n_j 表示备选实体 $e_{j,q}^i$ 的上下文词汇表中词的总数（包括全部重复的词）， v 表示整个文档集中无重复的词的个数，即整个维基百科中词的

类别数。

至此，我们将每个类中所有词的类条件概率乘积的相对大小作为候选实体 e 对应指称项 m 的相似度：

$$\prod_{i=1}^k P(d_i | e) \quad (3-8)$$

同一篇文档中的词的重要程度并不相同，为了更好的反应文档特征，我们进一步对相似度进行 TF-IDF 加权¹²，根据每个词语 d_i 在候选实体集 E 中的分布情况为每一个分量 $P(d_i | h)$ 赋予一个 IDF 权值。然而由于公式 3-9 中指称项 m 与其候选实体 e 的相似度是类条件概率的连乘，如果直接为每一个分量乘以一个 IDF 权值，相当于为每个备选实体 e 的相似度乘以一个常数，无法起到加权的作用，因此对公式 3-9 取对数转化为类条件概率和的形式，得到上下文加权相似度

$$sim_{ct}(e_{j,q}^i) = \sum_{k=1}^m \left(\log P(d_k | e_{j,q}^i) \times \log \frac{|E_j^i|}{1 + |\{t : d_k \in e_t\}|} \right) \quad |\{t : d_k \in e_t\}| > 0 \quad (3-9)$$

其中 $|E_j^i|$ 为命名指称项 m_j^i 的候选实体个数， $|\{t : d_k \in e_t\}|$ 为实体集合 E_j^i 中，上下文中包含词语 d_k 的实体个数。 $P(d_k | e_{j,q}^i)$ 利用公式 3-8 求得。

实验中我们发现 $|\{t : d_k \in e_t\}|$ 为零的时候，表示上下文中不存在词 d_k ，对应的相似性应该取最小值，但由于 TFIDF 的计算公式没有考虑这个问题，因此在实际操作中我们对 $|\{t : d_k \in e_t\}|$ 进行判断，如果为零，则该词对应的 TFIDF 为零。同样，当某个提及的候选实体为空，即 $|E_j^i|=0$ 的时候，TFIDF 权重应该为 0。结合 3-10，这里给出完整的基于朴素贝叶斯的上下文加权相似度计算公式：

$$sim_{ct}(e_{j,q}^i) = \sum_{k=1}^m \left(\log P(d_k | e_{j,q}^i) \times TFIDF_{i,j}(d_k) \right) \quad (3-10)$$

其中， $P(d_k | e_{j,q}^i)$ 代表词 d_k 在文档 $e_{j,q}^i$ 中出现的频率，利用公式 3-8 求得， $TFIDF_{i,j}(d_k)$ 表示词 d_k 在文档 $e_{j,q}^i$ 中的 TF-IDF 权重，定义如下：

¹² 详见2.6.4中公式2-4~2-6对TF-IDF定义

$$TFIDF_{i,j}(d_k) = \begin{cases} \log \frac{|E_j^i|}{1 + |\{t : d_k \in e_t\}|} & \left(|\{t : d_k \in e_t\}| > 0 \right) \\ 0 & \left(|\{t : d_k \in e_t\}| = 0 \cup |E_j^i| = 0 \right) \end{cases} \quad (3-11)$$

其中 $|E_j^i|$ 为命名指称项 m_j^i 的候选实体个数, $|\{t : d_k \in e_t\}|$ 为实体集合 E_j^i 中, 上下文中包含词语 d_k 的实体个数。

利用基于朴素贝叶斯的加权相似度作为特征向量, 可以有效地对上下文特征进行降维表示, 同时避免了由于微博文本上下文内容较少导致余弦相似度计算中相似度为零的情况, 保证了算法在长尾文本中的分辨能力。

3.4.3. 临近上下文文本特征提取

上一小节中, 我们通过提取指称项和命名实体上下文之间的加权相似度, 可以得到其语义层面上的相关度信息。然而观察发现, 一些同名实体即便主题相关, 具有相似的上下文, 仍会因为其自身类别属性的不同而具有较大的差异。比如电影“泰坦尼克号”和游轮“泰坦尼克号”, 其相关上下文都会涉及对其自然属性的描述以及对沉船事故的介绍。单独利用上下文计算文本相似度, 无异于通过生硬的比较单词分布来判断两篇文章是否主题相关, 这种方法仅对主题内容差异较大的上下文场景有效, 而对于主题差异较小但名词类别不同的实体上下文场景, 我们需要一种更为精细化的相似度比较方式。另一方面, 由于微博文本长度较短, 传统的上下文相似度捕捉到的语义信息十分有限, 甚至有的微博会出现无上下文的极端情况(如表 3-1 中第 111 条微博)。为了解决上述问题, 本文使用 *临近上下文相似度*(*Adjacent Context Similarity*) 对指称项和命名实体的类别特征进行提取和比较。

我们将命名实体或指称项的前一个词和后一个词分别称为临近上、下文。通过观察发现这些与实体名词位置紧密相连的词包含着丰富的能够反映名词类别的信息, 比如微博中提到 “How did we get a New Benz? I'll show you.....”¹³, 文中的指称项既可以表示人名 Karl Benz, 也可能汽车品牌 Mercedes-Benz。显然文中提供的上下文并不足以支撑模型做出正确的判断, 但实际上 “new” 这个词更多用来形容汽车

¹³ UID35619, Index85

而非人物，经过对候选实体的上文词典检索发现，“new”在“Mercedes-Benz”前出现过 23 次，而从未在“Karl Benz”前使用过，这验证了我们之前的假设，说明实体类别属性在语义相近的上下文场景中，对实体的区分能力要高于实体上下文特征。

在知识库预处理阶段，我们对每一个实体 e 前后两个词进行统计，分别得到该命名实体的临近上文词典 $AD_{left}(e)$ 和临近下文词典 $AD_{right}(e)$ 。与实体上下文的预处理类似，我们会参照停用词词典滤除停用词，但临近上下文停用词词典会保留一些具有实体类别指示性的词，比如表方位的介词“in”、“on”、“above”等，除此之外，一些无特殊指示性的介词（如“and”）、数字、单个字母、符号等均被列为停用词。

表 3-1 微博语料库举例

用户	编号	微博正文
50888543	77	RT @MarcCarig: RT @BryanHoch: RT @DKnobler: And <u>Yankee Stadium</u> remains only current <u>AL</u> park where <u>Justin Verlander</u> has never won.
50888543	111	Here we go. E...A...G...L...E...S..... <u>EAGLES!!!</u>
101935227	27	Opening Day at <u>Yankee Stadium</u> . Amazing experience to watch in person. http://t.co/jiEyI6kE
4081481	4	Evan Turner's personal war with the <u>Bulls</u> front line ended with a foul.

指称项与命名实体临近上下文的相似度计算与全文上下文相似度类似，此处不予以赘述¹⁴。对于第 i 条微博的第 j 个待消歧的命名指称项 m_j^i ，其临近上下文 $D(m_j^i) = \{d_l, d_r\}$ 分别与候选实体 $e_{j,q}^i \in E_j^i$ 的临近上文词典 $AD_{left}(e)$ 和临近下文词典 $AD_{right}(e)$ 计算加权上下文相似度，并对结果取均值，得到指称项与候选实体的临近上下文相似度 $sim_{ac}(e_{j,q}^i)$ ，表示如下

¹⁴关于基于朴素贝叶斯的加权相似度推导参见3.4.2

$$sim_{ac}(e_{j,q}^i) = \frac{\log P(d_l | AD_{left}(e_{j,q}^i)) \times TFIDF_{i,j}(d_l) + \log P(d_r | AD_{right}(e_{j,q}^i)) \times TFIDF_{i,j}(d_r)}{2} \quad (3-12)$$

其中, $P(d_l | AD_{left}(e_{j,q}^i))$ 和 $P(d_r | AD_{right}(e_{j,q}^i))$ 分别表示 m_j^i 的临近上下文在实体 $e_{j,q}^i$ 的临近上下文词典中出现的频率, 利用公式(3-7)求得, $TFIDF_{i,j}(d_k), k \in \{l, r\}$ 表示词 d_k 在 $e_{j,q}^i$ 的临近上下文词典 $left_{j,q}^i$ 和 $right_{j,q}^i$ 中的 TF-IDF 权重, 定义见公式(3-11)。

实验表明, 临近上下文在微博短文本中具有较高的分辨力, 是对传统基于上下文特征的补充和完善。

3.4.4.面向层次消歧的实体相关性提取

我们利用前面三节分别介绍了实体流行度、基于朴素贝叶斯的加权相似度以及临近上下文相似度, 根据 2.5 中的介绍, 这些特征属于独立特征, 即不同指称项之间的消歧任务是独立进行的。由于同一篇文档中往往存在多个指称项需要进行消歧, 直观上同一篇文章中涉及的实体应该具有一定的主题一致性, 如果能够充分挖掘待链接实体之间的关系, 并综合利用未知实体与已知确认实体之间的关联关系, 就可以进一步提高消歧结果的准确性。

本文从维基百科的超链接信息中提取实体之间的关系信息。通过对维基百科进行锚文本挖掘, 得到关系矩阵, 以对这种实体间的链接关系进行建模。具体来讲, 我们利用 Milne 和 Witten 提出的维基百科概念之间的语义关联度方法 WLM^[69]计算实体之间关联度。对于两个实体 e_1 和 e_2 , 本文中用 $TR(e_1, e_2)$ 表示其实体相关性。

$$TR(e_1, e_2) = 1 - \frac{\log(\max(|E_1|, |E_2|)) - \log(|E_1| \cap |E_2|)}{\log(|WP|) - \log(\min(|E_1|, |E_2|))} \quad (3-13)$$

通过对知识库的预处理, 可以利用上式得到任意两个实体间的相似度, 从而构造实体相似度矩阵。实体相似度是根据知识库归纳出的实体间静态特征, 由于实体链接任务的输入中并不包含任何确认的实体, 因此我们无法直接利用实体相关性, 传统协同消歧方法^[16]将实体消歧转化成基于图的兴趣传播问题, 而将实体相似度作为边的权重, 引导提及对于某一候选实体兴趣在图上的传播。本文在 4.3 节提出层次化实体消歧思想, 对于利用实体相关性特征提供了一种新的思路。

3.5 实验结果与分析

3.5.1. 候选实体词典分析

本小节对本文构建的候选实体词典的质量进行分析。实体词典的质量泛指词典的收词量、候选实体覆盖率等指标，综合反映了候选实体的完备程度。由于实验条件有限，我们无法直接得到其他研究中所使用的候选实体词典的收词量等数据，因此本节利用消歧算法仅使用流行度特征条件下的独立消歧准确率作为候选实体词典的质量评估标准。这样做的依据在于，独立消歧算法只利用流行度进行消歧，即返回词典每个条目下流行度最高的实体，由于流行度是上下文无关特征，而不同方法对于实体流行度的统计又相对稳定，因此质量相似的候选实体词典理论上应该具有相似的准确率。

本文使用与当前微博实体链接最新研究 Shen^[16]相同的人工标注数据集，对照组采用其文中算法 KAURI 的独立特征模式(令 $\beta = 0, \gamma = 0$)，数据集详细信息见 4.4.1。

实验结果如表 3-2 所示。

表 3-2 候选实体词典质量分析

方法	InKB 准确率	NIL 准确率	平均准确率
$KAURI_{\beta=0, \gamma=0}^{local}$	79.60%	81.20%	79.9%
$HEL_{\beta=0, \gamma=0}^{local}$	79.28%	71.00%	77.92%

表中准确率均为 KAURI 与本文 HEL 算法在使用独立消歧函数，且只是用流行度特征下的算法准确率。我们注意到二者准确率大体相当，说明本文提取的候选实体词典质量与目前先进方法一致。同时我们注意到本文方法 NIL 准确率较对照组低 13%，说明与对照组相比，本文候选实体词典多找出 13% 的空实体的候选实体，说明本文的候选实体集内容更丰富，多出的部分可能是潜在的被用户忽略的存在链接实体的指称项。

3.5.2. 实体流行度分析

如表 3-3 所示，试验中我们发现虽然每一个命名指称项拥有多个候选实体，但其频数分布十分不均。

例如，“NYC”作为命名指称项出现时，有 274.5625 次都是代表实体“New York City”，而仅有 1.0625 次代表实体“John F. Kennedy International Airport”，因此我们可以认为，对于命名指称项“NYC”，“New York City”是相对比较流行（Popular）的实体，命名指称项“NYC”代表该城市的概率要比代表乐队“NYC (band)”或 JFK 机场的概率要大得多。表 3-2 表明，超过 70% 的指称项链接到流行度最高的候选实体。

表 3-3 维基百科中提及-实体对的均分词频统计

命名指称项	命名实体	均分词频
NYC	New York City	274.5625
	NYC (band)	12.0625
	John F. Kennedy International	1.0625
	Airport	

New Orleans	New Orleans	8550.649
	New Orleans Saints	53
	New Orleans Union Station	1

.....

3.5.3. 临近上下文分析

实验中我们发现，对于微博文本，其上下文相似度对算法准确度的提升不及临近上下文相似度（提升幅度差值达到了 2.13%），这也符合 3.4.3 中我们的假设。具体来看，表 3-1 第四个例子中指称项“Yankee Stadium”利用本文算法在不同特征下的打分如表 3-4 所示。在 Wikipedia 中指称项“Yankee Stadium”被指向电影“Yankee Stadium(1923)”的次数要高于运动场“Yankee Stadium”，另一方面由于微博文本较

短，所提供的上下文信息也十分有限，而两个候选实体描述的对象也很相似，故上下文相似性特征的引入仅将二者差距缩小 45%，并未起到很强的区分作用。最终这条指称项得以链接成功很大程度上取决于临近上下文的引入，通过对临近上下文词典中二者词条的分析，我们发现微博中介词“at”出现在地点名词 Yankee Stadium 前的概率远高于出现在电影之前。正是通过这种词类层面的特征，使消歧算法可以认识到此处需要链接到一个地点而不是电影，从而返回正确结果。

表 3-4 不同参数对静态链接方程打分的影响

特征	Yankee Stadium	Yankee Stadium (1923)
Pop	0.45	0.54
Pop+Ct	-1.58	-1.38
Pop+Ct+Ac	-6.26	-7.34

3.6 本章小结

本章针对上文对微博文本特性的分析，对实体链接预处理方法进行优化改进。首先参考主流的候选实体词典生成方法，选择维基百科作为知识库，充分挖掘其各类页面中丰富的链接信息以及 Infobox 中结构化信息，实验表明本文生成的候选实体词典质量与当前最新方法保持一致。

在特征选择和提取方面，针对微博文本上下文不足这一问题，引入临近上下文特征以增强算法对实体类别的分辨能力。对于文本上下文特征，将其相似度计算视作一个朴素贝叶斯分类问题，从而实现对传统基于余弦相似度方法降维表示，同时避免了由于微博文本上下文内容较少导致余弦相似度计算中相似度为零的情况，保证了算法在长尾文本中的分辨能力。最后针对传统流行度词频统计方法对重定向页面统计缺失或重复统计的问题，提出基于均分词频的实体流行度，避免了有效词义信息遗漏，提升了流行度特征的准确性。利用 WLM 表征实体相关性，并在下一章提出基于信息函数的层次化实体消歧思想，利用用户特征为衡量实体相关性提供参考，以解决协同消歧中实体相关性特征对确认实体的全局依赖问题。

值得一提的是，本文改进的预处理方法在提高了实体流行度特征的可靠性的基础上，降低了传统方法对上下文的依赖，为下一步面向微博文本的实体消歧算法提供了更具分辨力的特征，确保了消歧算法在微博文本集上的消歧能力。

第四章 基于信息函数的层次化实体消歧方法

4.1 引言

通过上一章对实体链接预处理方法的改进，得到了较为完备的候选实体词典，提取了适用于表示微博文本特点的特征，基于以上工作，本章将针对传统协同消歧算法存在的问题，提出基于信息函数的层次化实体消歧方法，在保证算法链接准确率的基础上，降低了模型复杂度。

我们受到 Shen 等人^[16]的启发，尝试将用户兴趣这种微博周边信息引入实体链接算法，从而增强链接效果。基于用户兴趣的实体链接思想基于同用户微博实体主题相关假设，通过将用户微博中歧义较小的那部分指称项的链接实体汇总成确认实体池（Certain Entity Pool, CEP），从侧面对用户兴趣进行描述。

根据上一章的工作，本文算法综合利用实体流行度，上下文相似度，临近上下文相似度等独立特征；此外，传统的基于独立特征的实体消歧方法没有有效地利用实体间相关性的信息，而协同消歧方法由于实体链接候选实体正负样本严重不均，无关实体在模型计算的时候会对消歧结果产生影响，针对上述问题，提出层次化的实体消歧思想，在保留了有效相关实体的对于消歧算法的积极影响的同时，减少了非相关实体的干扰。层次化实体消歧算法依赖于对同一用户指称项模糊程度的排序，本文提出信息函数的概念来定量计算实体别名的模糊程度。

在文章结构上，第二节主要介绍实体消歧相关工作，第三节结合前人工作，提出层次化实体消歧算法，详细介绍信息函数、算法框架及空实体预测等技术要点，并于第四节对实验结果进行分析讨论。

4.2 传统实体消歧相关方法

本节对相关实体消歧方法进行概述，并就独立消歧与协同消歧方法在针对微博文本消歧任务存在的问题进行讨论。

4.2.1. 独立消歧方法

独立消歧方法认为对于一篇文档，其指称项之间的消歧任务相互独立。由于不考虑实体之间、指称项之间的关联关系，独立消歧算法模型中只能利用上一章提到非相关特征（如流行度、上下文文本相似度……）。这些特征的特点是静态的、不随消歧任务执行顺序而变化，因此独立消歧方法可以转化为有监督的二分类问题或无监督的打分排序问题来进行求解。

对于二分类问题，给定一个提及和它对应的候选实体对 $\langle m, e_i \rangle$ ，算法通过训练一个二分类分类器来判断哪个实体是提及真正的链接对象。 $\langle m, e_i \rangle$ 被表示成一组特征向量，训练阶段利用部分标注的 $\langle m, e_i \rangle$ 对分类器进行训练，测试阶段如果分类器认为 e_i 是链接结果，返回正值，否则为负。分类器可以选择包括 SVM、决策树、关联规则、神经网络在内的不同方法。然而该方法的主要问题是，对于一个指称项，可以有多个候选实体被判断为真，在这种情况下算法只能利用其他方法进行二次选择，如进一步使用 SVM 通过对训练集训练一个与最近的正负样本间隔最远的超平面进行分类。实际上二分类问题最终仍将转化为一个排序问题，因为分类器永远不会刚好产生一个正样本的分类结果。另一方面，分类器的训练样本极度不均，Ji 等人^[61]的研究表明，TAC-KBP2011 的数据集中，每个指称项平均对应 13.1 个候选实体，也就是说训练集中正负样本的比例在 10 倍以上，在我们的试验中，对于某些模糊的指称项，这一数字可以达到上百倍，样本不均同样会影响分类器准确性。

另一个主流的独立消歧方法是排序消歧，即通过构造融合多个特征的打分函数，对每一个候选实体根据其相对于指称项的相似度进行打分排序，最终返回得分最高的候选实体作为链接结果。Shen 等人^[15]利用了实体流行度、上下文相似度、语义相似度和全局主题一致性四个特征的线性组合作为排序函数，并训练 SVM 分类器学习权重 ω 。基于排序 SVM 模型^[70]的方法利用基于训练集的最大间隔技术（max-margin technique）对模型进行训练，对于提及 m 给定真值 $e^m \in E_m$ ，则真值实体打分 $Score(e^m)$ 应该比其他实体打分 $Score(e_i), e_i \in E_m \quad e_i \neq e^m$ 高出一定的差额，由此对于每个指称项的候选实体一个 SVM 线性约束：

$$\forall m, \forall e_i \neq e^m \in E_m : Socre(e^m) - Socre(e_i) \geq 1 - \xi_{m,i} \quad (4-1)$$

使 $\xi_{m,i} \geq 0$ 以及 $\|\omega\|_2^2 + C \sum_{m,i} \xi_{m,i}$ 最小化, 其中 C 是平衡间隔距离与训练误差的参数。

排序消歧相比于二分消歧, 其优势在于算法支持返回唯一解作为实体连接结果, 从而避免了二次分类的问题, 但问题在于特征的训练依旧没有摆脱分类器训练, 因此通过分类器获得的权重不够准确。另一个问题在于, 排序消歧思想并没有考虑空实体的判别问题, 理论上只要指称项存在候选实体, 算法就会返回一个实体作为链接结果, 空实体的判别是独立于消歧算法之外的工作。

此外, 基于独立特征的消歧算法的特征选择存在一定的局限性, 没有充分利用同一文档中实体主题相关性的假设, 忽略了实体间相关性的作用。

4.2.2. 协同消歧方法

一种协同消歧方法将实体链接建模成基于图的兴趣传播问题。该方法首先对于每一个微博用户建立一个图来对其微博指称项以及候选实体间的依存关系建模 (如图 3-2), 进而利用上一节提到的独立特征来对个候选实体的打分进行初始化, 最后利用兴趣传播算法将用户兴趣打分在全局候选实体间进行传播, 并根据算法收敛结果得到实体链接结果。在算法初始化阶段, 基于独立特征的打分函数可以表示成:

$$p_{j,q}^i = \vec{\omega} \cdot F_m(e) \quad (4-2)$$

其中 $F_m(e) = \langle sim_1(e), sim_2(e), \dots, sim_n(e) \rangle$ 为一系列候选实体 e 与指称项 m 的特征相似度, $\vec{\omega}$ 为这些特征的权重向量。

在兴趣传播阶段, 对于每个用户构建一个图 $G = (V, A, W)$, 其中 $|V|$ 表示顶点数, $v_k, 1 < k < |V|$ 表示第 k 个顶点, 其初始值根据公式(4-2)表示为 p_k 。任意两定点间的边定义为 $\langle v_k, v_{k'} \rangle$, 边的权重定义为 $W(v_k, v_{k'})$, 表示定点间兴趣传播强度, 经过归一化得到归一化的兴趣传播强度 $NW(v_k, v_{k'})$

$$NW(v_k, v_{k'}) = \frac{W(v_k, v_{k'})}{\sum_{v_c \in V_{v_k}} W(v_k, v_c)} \quad (4-3)$$

其中 V_{v_k} 是与顶点 v_k 相连的边的集合。

令全部实体最终的兴趣得分为 $\vec{s} = (s_1, \dots, s_k, \dots, s_{|V|})$, 则任意实体最终得分 s_k 由其初始分数和其他相关节点传递给它的分数决定, s_k 可以通过矩阵相乘得到。令初始分数向量为 $\vec{p} = (np_1, \dots, np_k, \dots, np_{|V|})$, B 为 $|V| \times |V|$ 的兴趣传递强度矩阵则有

$$\vec{s} = \lambda \vec{p} + (1 - \lambda) B \vec{s} \quad (4-4)$$

其中 $\lambda \in [0,1]$ 为调节初始分数权重的参数。

协同消歧的过程就是从初始分数 $\vec{s} = \vec{p}$ 开始, 随着迭代, \vec{s} 不断收敛的过程。我们注意到由于每一次迭代是对全局实体的同步更新, 因此图 G 中不相关的实体同样会对局部消歧产生负面影响, 而不相关实体在图中又占多数。

综上所述, 协同消歧所得到的全局最优解并不是产生局部最优解的充分条件, 同时协同消歧算法要进行多轮迭代, 迭代次数受到初始分布等多方因素的影响, 算法效率较低。

4.3 层次化实体消歧算法

本节针对上文协同消歧算法存在的问题, 首先介绍层次化实体消歧算法框架, 给出信息函数定义, 并详细介绍算法思路。对于非监督的实体消歧方法, NIL 的确定是一个难点, 本章最后将结合传统得分阈值 τ 与候选实体相似度方差阈值 ς 对空实体进行预测。

4.3.1.方法框架

根据 4.2 的介绍, 当前主流的实体链接方法按照链接任务的独立性不同主要分为独立消歧和协同消歧。然而通过我们关于对我们对歧义实体的理解过程分析, 微博文本的实体链接任务中, 针对同一个用户微博中不同指称项的消歧任务之间既不是独立的, 也不是同时发生, 而是按照实体别名的模糊程度, 逐层消解。独立消歧、协同消歧及层次消歧算法的模式比较参见图 4-1。

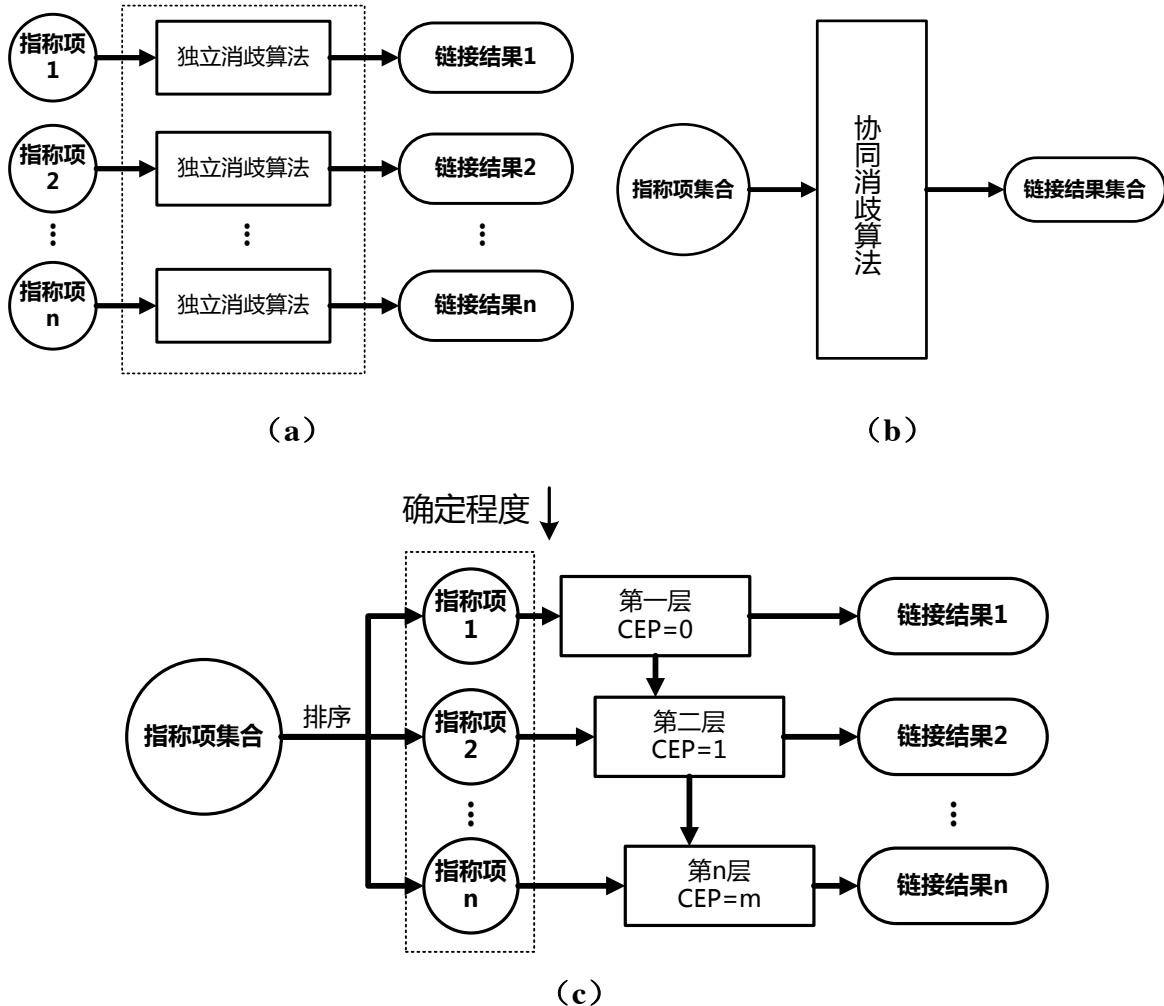


图 4-1 三种消歧算法流程。

(a)独立消歧; (b)协同消歧; (c)层次消歧

算法流程如图 4-2 所示。算法的输入是来自不同用户的微博集合 T 以及 T 中识别得到的一组命名实体提及 M 。算法首先对全部微博和提及按照用户进行分类, 对于每一个用户的全部提及 M_{user} , 利用信息函数计算其信息量并降序排列, 得到排序后的提及 M'_{user} , 从 M'_{user} 中依次按照提及明确程度由高到低取出第 i 篇微博的第 j 个提及 m_j^i , 通过查询候选实体词典判断其是否为空实体, 若不是则进一步判断确认 CEP 是否为空。如果 CEP 为空, 则使用冷启动消歧算法对 m_j^i 进行消歧, 返回相似性最高的候选实体; 如果 CEP 不为空, 则使用层次化实体消歧算法对 m_j^i 进行消歧, 返回相似性最高的候选实体。对于返回的候选实体, 进行 NIL 判断, 如果满足 NIL

条件，则输出 *NIL*，若不符合 *NIL* 条件，则将该候选实体作为链接结果输出并对其进行 CEP 准入判断，将符合条件的链接结果加入 CEP。并读入下一条指称项或结束算法。

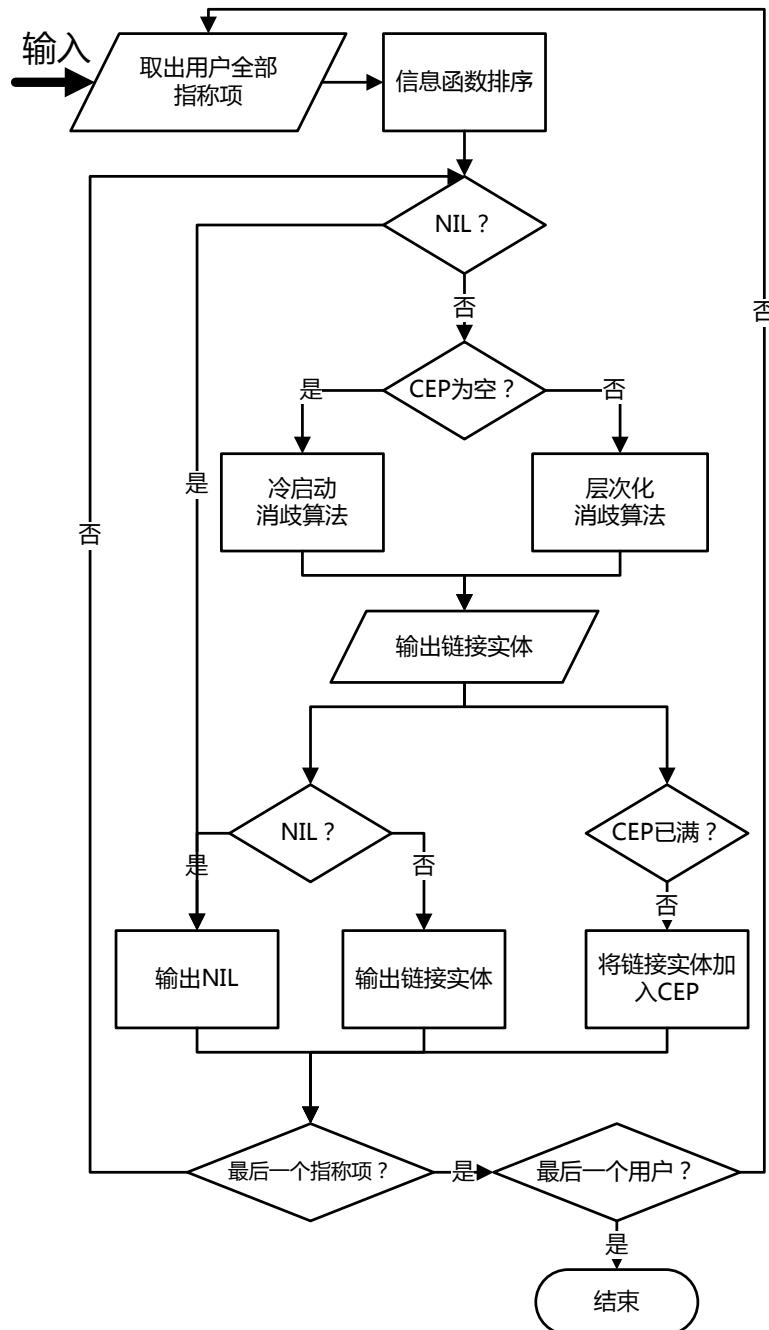


图 4-2 层次化实体链接算法框架

4.3.2.信息函数

层次化的实体消歧顺序依赖于对微博中指称项模糊程度的排序，本节将首先介绍这种排序思想，进而提出信息函数的概念来对指称项的模糊程度进行定量计算。

通过观察人们对于微博文本中出现的实体指称项的理解过程，我们发现在人们试图去理解一篇微博中的不同指称项的含义的时候，如果遇到不能确定的模糊名称，会很自然的倾向于先去理解那些容易理解的名称，然后带着从确认实体中得到的先验知识去理解那些模糊的提及。例如，对于下面所示的一条微博文本（实体别名用下划线表示）：

“今天华盛顿天气很好，我们陪同家人分别游览了乔治华盛顿大学和国立博物馆！”

当我们在见到“华盛顿”、“乔治华盛顿大学”和“国立博物馆”这三个独立的名词的时候，会直观的认识到这三个词的模糊程度是有差异的，比如“国立博物馆”有可能代表阿姆斯特丹国立博物馆，国立故宫博物馆东京国立博物馆亦或是美国国立博物馆；华盛顿可能代表美国国父乔治华盛顿，美国位于西海岸的华盛顿州或者美国首府；而乔治华盛顿大学则基本上没有歧义，代表位于美国首府华盛顿的一所大学。在泛读这样一条微博的时候，拥有如上常识的人（此处可以理解成知识库覆盖到这些词条），可以准确的将这些实体别名与背后的实体相对应，表面上看似乎是一同时或者顺序完成的，但实际上这个认知过程是根据实体别名的模糊程度逐层完成的。对于“乔治华盛顿大学”，由于提供了完整长度的实体名称，且该名称下没有其他歧义实体，因此含义最为清晰。我们对于“华盛顿”和“国立博物馆”的理解其实都是带着“乔治华盛顿大学”这个先验知识进行的，由于“乔治华盛顿”大学坐落于美国首府“华盛顿 DC”，因此第一个指称项代表华盛顿的几率更大，同样，即便“国立博物馆”可能指代若干个国家的博物馆，但由于它经常与“乔治华盛顿大学”同时出现，且坐落于“华盛顿 DC”，因此在以上两个先验知识的帮助下，我们可以确定这里的“国立博物馆”指的就是“美国国立博物馆”。

为此，我们提出信息函数的概念来衡量实体名词模糊的模糊程度。对于实体名称的模糊程度，一个很明显的特征是其候选实体的数量，如果一个名称对应着较多的实体，自然增加了做出选择的成本，比如上文例子中华盛顿可以代表若干地名和

人名，确定程度自然不及只有一个候选实体的“乔治华盛顿大学”；此外，名称的长度也是一个影响其模糊性的重要指标，名称越长的实体，对自身的描述能力越强，其模糊性自然也就越弱。根据以上分析，我们进行如下假设：(1)候选实体较少的提及信息量较高；(2)字符串较长的提及信息量较高。信息函数定义如下：

$$Info(m) = \log\left(\frac{Len(m_j^i)}{|e_j^i|}\right) \quad (4-5)$$

其中 $Len(m_j^i)$ 为提及 m_j^i 的字符串长度， $|e_j^i|$ 为提及 m_j^i 的候选实体数量。

实验证明信息函数可以反映我们对指称项模糊程度的认识（见表 4-1），且经过信息函数排序后，前 10% 的提及链接准确度可以达到 99% 以上（见 4.4.3 中确认实体池代表性分析实验），这一点保障了后文中确认实体池的效力。在接下来的消歧任务中，我们将按照信息函数的打分对不同模糊程度的指称项进行层次化的消歧。

表 4-1 命名指称项的信息量

Mention	len(o)	C(o)	Info(o)
Washington	10	30	-0.48
George Washington University	26	1	1.41
National Museum	14	23	-0.22

4.3.3. 层次消歧算法

层次化实体消歧方法是一种非监督的基于打分排序思想的实体消歧方法，对指称项的每一个候选实体根据各特征之间的相似性进行打分，并选择分数最高的那个候选实体作为链接结果。打分函数包含一个随着链接任务进行不断自增长的确认实体池，确认实体池定义如下：

定义 4.1：确认实体池(Certain Entity Pool, CEP)。经过信息函数排序并消歧后得到的前 n 个实体称为确认实体，确认实体池是确认实体的集合，用来表征用户偏好。

在未得到确定的实体之前，无法使用实体间的相似性（冷启动过程），而当消歧算法完成了对那部分明确的实体别名的消歧任务之后，基于已知的确认实体先验知识，利用实体间相关性对后续的消歧任务进行增强（层次消歧过程）。本节将分

别介绍冷启动和层次消歧的具体过程，随后给出空实体预测方法。

4.3.3.1. 消歧算法冷启动

对于一个新的用户，在算法起始阶段，系统并未掌握任何带有用户偏好的确认实体，即确认实体池为空，故此时的消歧任务主要依赖数据的独立特征进行。我们将微博 t_i 中的第 j 个指称项 m_j^i 的第 q 个候选实体表示为 $e_{j,q}^i$ 。首先对于该用户的全部指称项 m_u^i ，根据信息函数利用指称项形式上的特征进行排序得到 m'_u ，排在前面的指称项具有更清晰表达和较低的歧义。针对每一个指称项的全部候选实体 E_j^i ，从发生概率层面上，其流行度能够有效的反应其被人们所熟知的概率；语义上层面上，候选实体与指称项之间的上下文相似度可以捕捉其语义关联；词性层面上，二者的临近上下文相似度可以为描述候选实体的类别属性提供更多的参考。这三项特征在候选实体间相互独立，因此可以作为静态独立特征来描述候选实体与指称项之间的亲疏程度。基于以上特征，我们对每个候选实体 $e_{j,q}^i \in E_j^i$ 利用静态相似方程加权打分得到候选实体 $e_{j,q}^i$ 相对于指称项 $m_{j,q}^i$ 的分数 $P_{j,q}^i$

$$P_{j,q}^i = \alpha Pop(e_{j,q}^i) + \beta Sim_{ct}(e_{j,q}^i) + \gamma Sim_{ac}(e_{j,q}^i) \quad (4-6)$$

其中， $Pop(e_{j,q}^i)$ 代表实体 $e_{j,q}^i$ 的流行度， $Sim_{ct}(e_{j,q}^i)$ 代表实体 $e_{j,q}^i$ 与指称项 $m_{j,q}^i$ 上下文的相似度， $Sim_{ac}(e_{j,q}^i)$ 代表实体与指称项临近上下文的，对上述三个特征的推导和计算，详见第三章公式(3-6)，公式(3-10)和公式(3-12)。 α, β, γ 分别为各特征的权重系数，可以表示为一个权重向量 $\vec{\omega} = \langle \alpha, \beta, \gamma \rangle$ 且 $\alpha + \beta + \gamma = 1$ ，通过学习得到。

冷启动算法与传统的独立特征消歧方法一致，但由于算法输入为经过排序的，歧义较小的指称项，因此冷启动阶段的消歧结果比较准确。

4.3.3.2. 迭代消歧

在经历过一轮消歧算法冷启动之后，会得到一些链接结果实体，由于经过信息函数的排序，前几轮的消歧结果具有较高的准确性，可以将他们视为确认实体，存入确认实体池 C_u 用来作为用户 u 的兴趣的表征。

为了保证确认实体池的效力，我们首先根据对数据的分析，制定流行度阈值 θ

来限制哪些实体可以进入确认实体池，并设置实体池容量阈值 V ，以在保证信息量充足的前提下降低算法复杂度。具体参见算法 4-1。

当确认实体池不为空的时候，如图 4-2 所示，模型将选择使用层次消歧算法，得到平均实体相关度 $\overline{TR}_{j,q}^i$ ：

$$\overline{TR}_{j,q}^i = \frac{\sum_{c_s^u \in C_u} TR(e_{j,q}^i, c_s^u)}{|C_u|} \quad (4-7)$$

其中 $e_{j,q}^i$ 为用户 u 的指称项 $m_{j,q}^i$ 对应的候选实体， $c_s^u \in C_u, 0 < s < V$ 表示用户 u 当前确认实体池 C_u 中的第 s 个确认实体， $TR(e_{j,q}^i, c_s^u)$ 为 WLM 实体相关性计算方法，参见公式 3-14。通过计算 $e_{j,q}^i$ 与 C_u 的平均 WLM 相关性，可以衡量候选实体 $e_{j,q}^i$ 是否符合当前用户的偏好。从而在应用独立特征消歧的基础上，进一步加强了消歧性能。

将 $\overline{TR}_{j,q}^i$ 作为特征补充进公式 3-16，便可得到层次化实体消歧方法打分方程：

$$d_{j,q}^i = p_{j,q}^i + \mu \overline{TR}_{j,q}^i \quad (4-8)$$

其中， $p_{j,q}^i$ 代表指称项 $m_{j,q}^i$ 与其候选实体 $e_{j,q}^i$ 的独立相似性得分， $\overline{TR}_{j,q}^i$ 为候选实体 $e_{j,q}^i$ 与确认实体池 C_u 的平均相关性， μ 为权重，通过实验训练获得。

结合上文中所述的特征，对于指称项 $m \in M$ 的候选实体 $e \in E_j^i$ ，可以得到它的特征向量 $F_m(e) = \langle Pop(e|m), Sim_{ct}(e), Sim_{ac}(e), \overline{TR}_m(e) \rangle$ 。此处对权重向量进行扩充得到 $\vec{\omega} = \langle \alpha, \beta, \gamma, \mu \rangle$ 。根据(4-8)计算得到其分数 $Score_m(e) = \vec{\omega} \cdot F_m(e)$ ，通过对候选实体根据其得分排序，最后选择 $e_{top} = \arg \max_{e \in E_m} Score_m(e)$ 作为指称项 m 的链接结果。

对于权重向量的学习，我们基于训练数据集，利用 Clementine 数据挖掘工具分别尝试利用 C5.0、Apriori、神经网络、Logistic 回归以及 SVM 对 \vec{w} 进行学习。然而正如 1.2.1 中提到，基于二分类的方法可能存在不止一个候选实体被标为正值，因此需要对这些正值进行二次选择，而由于模型的负样本远多于正样本，模型分类效果并不稳定。因此我们参考上述几种模型结果，经实验反复调优得到权重向量 $\vec{\omega}$ 。

层次消歧算法完整伪代码如算法 4-1 所示：

算法 4-1 层次消歧算法

输入: 用户 u 的全部待链接指称项 $m_j \in M$

初始化: 清空确认实体池

Step 1: 根据指称项模糊性排序

For $j = 1, \dots, |M|$

根据公式(4-5)计算 $Info(m_j)$

End

根据 $Info(m_j)$ 对 M 进行排序得到 M'

Step 2: 迭代消歧

For $j = 1, \dots, |M'|$

根据公式(4-9), 利用 $IsNIL_{pre}(m)$ 判断空实体

If $(|CEP| \neq 0)$

For $q = 1, \dots, |E_j|$

根据公式(4-8)计算候选实体打分

$$d_{j,q}^i = \alpha Pop(e_{j,q}^i) + \beta Sim_{ct}(e_{j,q}^i) + \gamma Sim_{ac}(e_{j,q}^i) + \mu \overline{TR}_{j,q}^i$$

End

Else

For $q = 1, \dots, |E_j|$

根据公式(4-6)计算候选实体打分

$$d_{j,q}^i = \alpha Pop(e_{j,q}^i) + \beta Sim_{ct}(e_{j,q}^i) + \gamma Sim_{ac}(e_{j,q}^i)$$

End

返回排序最高的实体 e_j 作为消歧结果

Step 3: 利用公式(4-10), 利用 $IsNIL_{post}(m)$ 判断空实体

Step 4: 判断是否加入确认实体池

If $(res.pop > \theta \cap |CEP| < V)$

将 e_j 加入 CEP

END

输出: 每个指称项 m_j^i 对应的消歧结果 $e_j^i \in E$

4.3.4. 空实体预测

上文提到的算法假设知识库覆盖全部查询实体，但实际上由于用户描述不准确、知识库不完整、候选实体词典统计不全面等因素，有时我们无法得到一个确定的实体作为链接结果。实体链接任务支持返回空实体（NIL）来描述上述情况。本文算法会进行两次空实体判断：

首先算法初始阶段如果指称项 m 对应的候选实体词典中候选实体集 E_m 的大小为零，我们即认为该指称项为不可链接指称项，并返回 NIL；如果 E_m 大小为 1，则无疑算法会返回唯一的候选实体作为链接结果；如果 E_m 包含多于一个候选实体，则依照算法 3-1 开始实体消歧，并返回打分最高的消歧结果，上述方法可以表示成：

$$IsNIL_{pre}(m) = \begin{cases} NIL & |E_m| = 0 \\ E_m & |E_m| = 1 \\ e_{top} = \arg \max_{e \in E_m} Score_m(e) & |E_m| > 1 \end{cases} \quad (4-9)$$

在得到消歧结果之后，算法需要再一次预测该消歧结果是否有可能为 NIL。传统方法包括通过训练集学习一个得分的阈值 τ ，如果 $Score_m(e_{top}) < \tau$ 则返回空实体。通过实验观察，我们认为对于空实体的判断不仅取决于候选实体与指称项的相似度大小，同时也与候选实体间的差异大小相关，如表 4-2 所示如果候选实体之间得分相似，即便得分较高，其分辨率也受到影响。因此本文在 τ 的基础上学习得到候选实体相似度方差阈值 ς 以度量该因素带来的影响，从而得到第二步空实体判断准则：

$$IsNIL_{post}(m) = \begin{cases} NIL & Score_m(e_{top}) < \tau \cap D(Score_m(E_m)) \\ e_{top}^m & other \end{cases} \quad (4-10)$$

其中 e_{top}^m 为消歧算法得到的结果， $D(Score_m(E_m))$ 为 m 的候选实体相似度方差。方差约束将候选实体集模糊的实体指称项也归为 NIL，由于提高了准入门槛，我们的 NIL 预测方法牺牲了部分 NIL 准确率以提高 InKB 准确率，这将在下一节具体讨论。

表 4-2 区分度不足的候选实体集

Vishnu	Vishnu	Vishnu	Vishnu	Vishnu	Vishnu	Vishnuvardhan	Vishnuvardhan
Smriti	(actor)	(band)	(1995 film)	(1994 film)	Purana	(director)	(actor)
-2.1247	-2.1588	-2.1598	-2.1598	-2.1599	-2.1600	-2.1600	-2.1601

4.4 实验结果与分析

本节主要介绍针对我们算法进行实验分析，以检验算法的效果，论证可行性及存在的问题。

4.4.1.实验数据

根据调研，目前尚没有针对微博文本的统一公开数据集，我们使用南开大学沈玮授权使用的人工标注数据集^[16]进行实验。该数据集利用 Twitter API 从随机采样得到的 71937 名 Twitter 用户的微博中提取 3200 条最新微博，并进一步从中随机选择 20 名用户，每名用户 200 条微博（不足 200 则全部纳入）进行人工标注，形成标准实验集。关于该数据集的详细情况见表 4-3，从中可以看出 3818 条微博中有 1721（45.08%）至少含有一个实体指称项。另有 241 条指称项难以对其作出准确判断，标记为 uncertain；437 条指称项被判定无任何知识库中的实体与之对应而链接到 NIL。最终，我们过滤掉不确定指称项，保留不可链接指称项，得到总计 2677 条测试用指称项。

我们下载了 2014 年 3 月 4 日的维基百科作为训练集，分别对其进行实体流行度提取、上下文文本提取及预处理、实体临近上下文词典构建、实体相关性矩阵构建并建立指称项的候选实体词典。

表 4-3 数据集概况

用户数	20
微博数	3818
至少含有一个实体指称项的微博数	1721
命名实体指称项总数	2918
不确定指称项数	241
测试用指称项总数	2677
可链接指称项总数	2239
不可链接指称项总数	438

4.4.2. 评测指标

根据 2.7 中对当前实体链接任务评价方法的概述, 我们选择 NIL 准确率和 InKB 准确率分别衡量实体链接对于空实体的判别能力以及多候选实体的消歧能力。其中

$$\text{NIL准确率} = \frac{\text{被链接到NIL的不可链接指称项个数}}{\text{不可链接指称项个数}} \quad (4-11)$$

$$\text{InKB准确率} = \frac{\text{被链接到NIL的不可链接指称项个数}}{\text{不可链接指称项个数}} \quad (4-12)$$

此外, 算法效力的综合评价指标参考 TAC KBP 实体链接评测任务的主要评测指标, 使用 Micro-averaged accuracy, 即所有链接结果的平均准确率:

$$P_{Micro} = \frac{\sum_{q \in Q} G(q)}{|Q|} \quad (4-13)$$

其中 Q 是所有查询实例的集合, $G(q)$ 为 2.7.1 中公式(2-12)提到的评价积分, 重写如下:

$$G(q) = \begin{cases} 1, & E_g(q) = E_s(q) \\ 0, & \text{其他} \end{cases} \quad (4-14)$$

式中 $E_g(q)$ 和 $E_s(q)$ 分别表示查询 q 的目标实体和系统返回的结果。

实验中我们发现, 与其他自然语言处理任务的评价一样, 实体链接算法的效果并不能完全通过定量的分析来进行评判, 本文使用的人工标注数据集的准确性受到标注者的知识范围的限制及主观判断的影响, 比如 11414222 用户的第 156 篇微博中 iPad 被指向空实体 (实际上我们的算法将其正确链接到苹果公司的 iPad)。此外不同的应用领域也对链接效果的评判有着不同的要求, 比如文献检索系统可能需要较高的准确率, 而诸如“百度知心”搜索中实体推荐系统就要求对用户感兴趣的实体进行启发性的链接, 保证召回率而弱化准确率; 知识库更新中对未登记新实体的检测, 要求空实体的准确率要足够高, 而在微博文本实体自动链接以进行阅读增强中, 则牺牲了部分 NIL 准确率以提高 InKB 准确率。

此外, 实验集中有些微博文本比较模糊, 存在模棱两可的答案, 这也是为自然语言处理任务的定量评价带来困难, 因此在下一节的实验结果分析中, 我们会结合定性分析和定量计算, 对本文算法效率进行分析。

4.4.3.实验结果

实验中我们首先对本文提出的确认实体池、信息函数等概念的效力进行定性定量分析，进而研究算法内部在不同变量条件下的表现，最后与其他微博实体链接算法进行横向比较。

确认实体池代表性分析：通过对实验中产生的确认实体池进行观察发现，CEP 在一定程度上能够反映用户的特征，表 4-4 是通过本文方法得到的部分确认实体池，可以看出其捕捉到了用户能够反映用户偏好且的歧义较小的实体，显然用户 14473492 对科技主题更感兴趣，而用户 4081481 热衷于篮球赛事，结合 3.4.4 中提到的实体相关性，那些同主题的模糊实体消歧效果可以得到增强：例如表 3-1 中对于类似 4081481 的第 4 条微博中的指称项“Bulls”，HEL 会根据 CEP 将其链接到 Chicago Bulls（篮球队）而非 Bulls (rugby union)（橄榄球队）。

表 4-4 确认实体池举例

用户	14473492	4081481
CEP	OS X	LeBron James
	Android(operating system)	Kobe Bryant
	San Francisco Bay Area	Utah Jazz
	Google	Michael Jordan
	Nokia	YouTube

信息函数用来定量计算微博提及的确定程度，通过对同一用户下所有指称项按照信息函数计算结果进行排序，理论上排名靠前的指称项应该具有更高的链接准确率。图 4-3 为经过信息函数排序后指称项排名与链接准确率关系，其中每条曲线代表一个用户 u ，其中横坐标 x 表示指称项排名，纵坐标 y 表示平均链接准确率，图上每个坐标点即代表用户 u 当前指称项排名 x 及以上的平均链接准确率 y 。通过实验发现，前 10% 的指称项用户平均链接准确率达到 98.63%（其中 80% 的用户链接准确率达到 100%），这一指标对前 60% 的指称项仍能保持在 90% 以上（见表 4-5）。考虑到

确认实体池的容量在 $8 \leq |C_u| \leq 18$ 范围内效果最佳（参见确认实体池敏感性分析实验），而每名用户平均拥有超过 130 个提及，因此该实验证明确认实体池在有效范围内表现出较高的准确率水平。

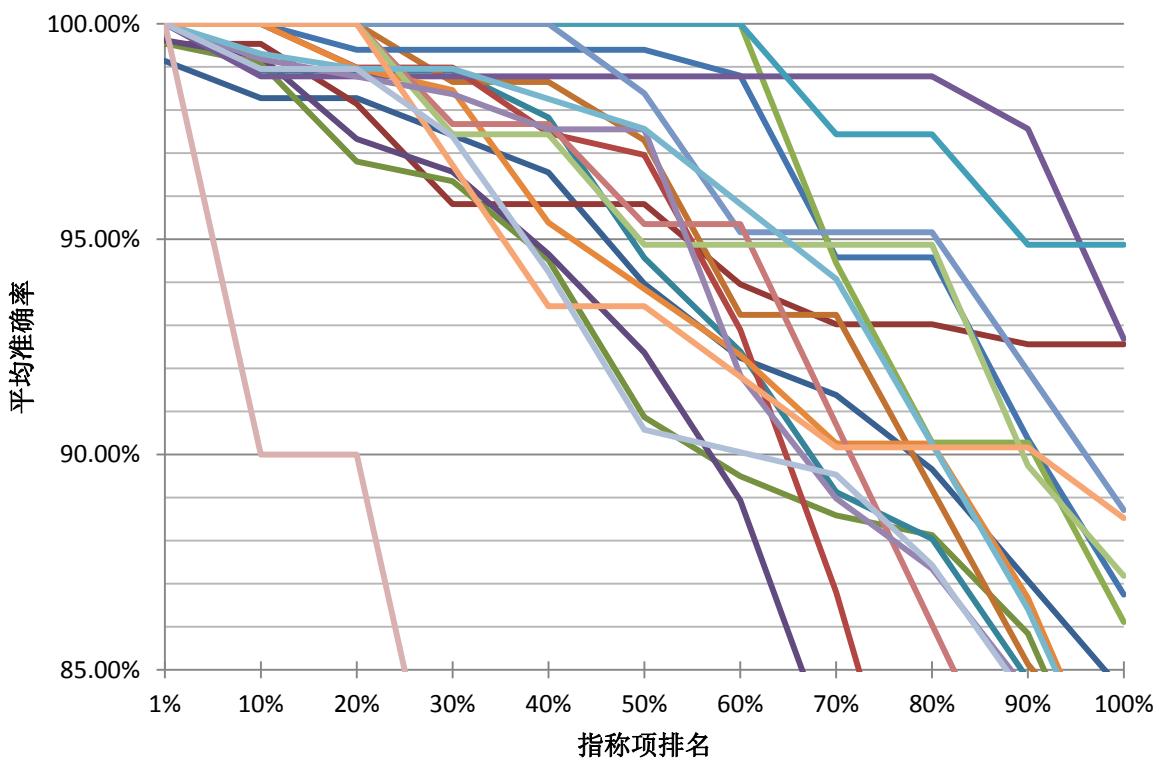


图 4-3 信息函数排序后的指称项排名-准确率

表 4-5 依指称项排名的用户平均准确率

指称项排名 (前百分比)	10%	20%	30%	40%	50%	60%
用户平均准确度	98.63%	97.73%	95.99%	94.65%	92.93%	90.62%

确认实体池敏感性分析：为了更好的理解 HEL 中确认实体池的特性，我们进一步针对确认实体池大小 $|C_u|$ 对算法效率的影响进行敏感性分析。图 4-4 中展示了一次实验中 $|C_u|$ 取不同值的条件下的算法准确率。可以看出，当 $8 \leq |C_u| \leq 18$ 的条件下，算法的准确率都保持在 86.7% 以上。因此我们可以说候选实体池容量在 8 到 18 这个范围内波动时，算法对 $|C_u|$ 不敏感。

用户偏差: 实体链接依赖于用户规范的语言表达, 而实际微博用户的语言习惯、兴趣偏好差异较大。本文算法在一定程度上受到这些用语问题的影响, 对于本文所使用的包含 20 名用户的数据集, 其用户准确率如图 4-5 所示, 准确率方差达到 0.05。通过对异常用户的分析, 影响算法准确率的因素主要来自于形式不一的缩写、俚语及长尾名词, 如用户 20 的全部微博中仅包含的 24 条指称项中, 就出现超出候选实体词典的长尾实体。本文在 6.2 中提出下一步研究内容, 以期从扩大实体词典的角度进一步的降低算法的用户敏感性。

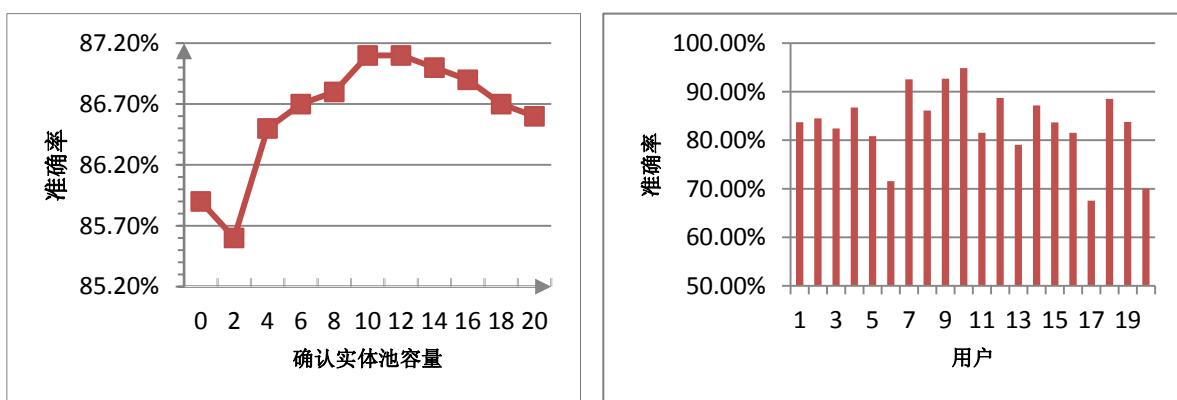


图 4-4 确认实体池容量对准确率的影响

图 4-5 用户准确率

特征分析: 基于控制变量的原则, 我们分别对静态消歧方程和层次消歧方程(分别用 LOCAL、HEL 表示)在不同特征下进行实体链接实验, 实验结果如表 4-6 所示。我们发现, 静态消歧方程在仅使用实体流行度作为消歧特征的条件下, 其准确率已经达到 77.92% (其中 InKB 准确率为 79.28%), 这也满足我们之前对于实体流行度在实体消歧中占据重要作用的讨论。网络数据分析中常使用用户点击率来表示实体流行度, 同样取得了较好的结果, 实际上流行度这种用户点击(引用)行为, 不仅仅描述了实体出现的概率, 更代表用户对该词条语义相关性的肯定, 蕴含着一种隐晦的语义信息。上下文和临近上下文特征的引入分别使静态消歧算法的准确率提高了 5.38% 和 2.59%, 最终使用全部特征的算法准确率达到 85.88%。

与静态消歧方法相比, 层次化消歧方法在使用相同特征的条件下, 其消歧准确率都得到了一定程度的提升。当 HEL 在只使用实体流行度时比相同条件下的静态消歧方法准确率提升了 0.44%, 在使用上下文和临近上下文的条件下分别提升了 0.23% 和 3.10%, 而使用全部特征的 HEL 算法准确率提升了 1.43%, 达到最大值 87.11%。

表 4-6 不同参数下实验结果分析

特征	可链接		不可链接		综合	
	正确链接数	准确率	正确链接数	准确率	正确链接数	准确率
LOCAL_{$\beta=0, \gamma=0$}	1775	79.28%	311	71.00%	2086	77.92%
LOCAL_{$\gamma=0$}	1887	84.28%	311	71.00%	2198	82.11%
LOCAL_{$\beta=0$}	1944	86.82%	311	71.00%	2255	84.24%
LOCAL_{full}	1988	88.79%	311	71.00%	2299	85.88%
HEL_{$\beta=0, \gamma=0$}	1783	79.63%	312	71.23%	2095	78.26%
HEL_{$\beta=0$}	1891	84.46%	312	71.23%	2203	82.30%
HEL_{$\gamma=0$}	2012	89.86%	313	71.46%	2325	86.85%
HEL_{full}	2019	90.17%	313	71.46%	2332	87.11%

算法横向比较：根据 1.2.2 中我们的调研，目前对微博实体链接方法的研究相对较少，但如果将每个用户下的全部微博视为一个独立文档集，我们可以将其与传统基于独立文档集的实体链接方法^[1, 10-13]进行比较。出于实际考虑，我们选择目前较先进的 LINDEN^[15]和 KAURI^[16]作为对比方法，在相同数据集上进行实验。LINDEN 与 KAURI 同样利用了上下文相似性和实体相关性等特征，前者是一种基于静态特征的网络文本实体链接方法，后者则利用协同消歧对微博文本进行实体链接。实验结果如图 4-6 所示。可以看出，本文方法在 InKB 准确率和全局准确率两项指标中均较对比方法有所提升，其中使用静态特征的 HEL_{local} 较 LINDEN 提高 4.25%，较使用静态特征的 KAURI_{local} 提高 3.62%，与使用基于图模型的 KAURI 持平。而使用全新迭代消歧框架的 HEL 表现最好，较 KAURI 提升 1.52%。我们注意到 HEL 在对于 NIL 与 InKB 实体的判断能力上有较大的差异，这是因为本文算法对 NIL 设立了较高的门限阈值，试验中我们发现基于排序的实体链接方法对于空实体的判断仍没有更好的解决思路，本文对于微博实体链接任务的要求是尽可能提高 InKB 实体的链接精度，实验证明我们方法在 InKB 实体判别的准确率高达 90.2%，比对照组最好结果高出五个百分点，对于 NIL 的问题，本文会在下一章继续讨论。

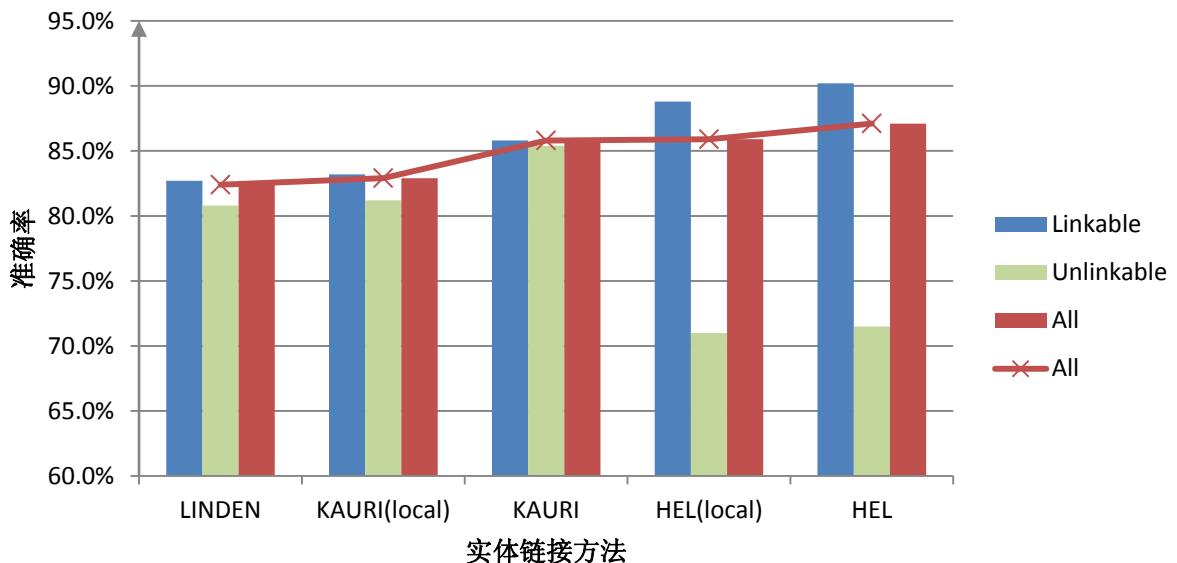


图 4-6 算法性能横向比较

4.5 本章小结

本章提出一种层次化的实体消歧方法，针对传统协同消歧方法在微博实体链接任务中特征描述性差、受到非相关实体影响及建模成本高等问题，结合对歧义实体的理解过程，创新性将微博文本的实体链接问题视为同一用户下依实体别名模糊程度逐层消歧的任务。用户特征通过确认实体池表示，命名实体提及的模糊性依照信息函数定量计算，并基于信息函数排序对指称项进行由上到下的消歧任务。通过在用户标注数据集上的实验分析，本文方法切实可行，并具有较高的链接准确率，尤其对传统方法的 InKB 准确率有较大提升，对空实体判别问题仍有较大进步空间。

结合第三章改进的实体链接预处理方法，本章算法可以较为准确地对微博文本中的实体名词进行歧义消解。本文将通过下一章实体链接原型系统实现层次化实体链接方法，并基于该平台开发扩展应用，实现算法增值。

第五章 实体链接原型系统设计

5.1 引言

实体链接问题研究的最终目的旨在使计算机具备从模糊的自然语言中分辨具体实体的能力，并结合这种知识进行信息组织、检索、问答、推荐等实际应用。作为一项植根于应用需求的科学的研究，各个领域都对实体链接系统有着迫切的需求，其中军事上主要利用实体链接系统对情报进行跨平台的组织关联，自动标注分类以及结合武器知识库的阅读增强；商业上能够通过对用户关注的关键实体的统计分析，实现定向推荐。科研实验中，实体链接系统便于对算法进行直观的呈现、统计分析以及应用扩展。因此，开展对于实体链接原型系统的设计研究有助于对算法应用价值的快速理解和转化。

本章结合上文提出的面向微博文本的层次化实体消歧方法，设计实现实体链接原型系统。在文章结构上，首先给出系统总体设计框架，在此基础上详细介绍各逻辑模块功能，最后具体演示系统运行步骤及功能界面。

5.2 系统总体设计

本文实体链接系统的总体设计如图 5-1 所示。本系统在.NET Framework 平台下利用 C# 语言和 WPF 技术开发完成，系统输入端首先与预先提取的维基百科知识库和微博数据集进行数据引接，原始数据通过预处理得到清洗过的文本、候选实体词典、实体流行度分布表、实体上下文词典、临近上下文词典、实体关系矩阵等特征数据，消歧算法内核进而利用这些特征，对用户微博文本中的查询指称项进行层次化的实体消歧。在得到消歧结果后，系统能够进一步对当前参数下链接算法的效果进行评估并生成评估报告。此外，作为实体链接算法实验试用平台，本系统支持基于链接结果的功能扩展，后续研究工作可以在其基础上增加诸如个性推荐、实体检索、自动问答等增值应用。

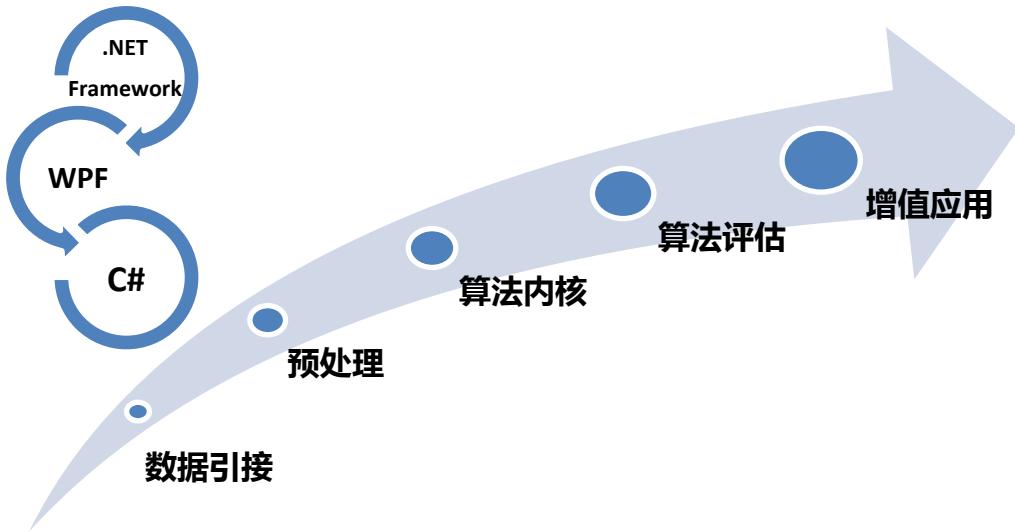


图 5-1 系统总体设计

5.2.1. 软件开发平台

.NET Framework 是 Microsoft 为开发应用程序而创建的一个具有革命意义的平台, 包含一个庞大的代码库, 可以在客户语言(如 C#)中通过面向对象编程技术(OOP)来使用这些代码。其代码托管和垃圾回收机制保障了代码的安全性、内存效率及跨语言调试的可能性。本文主要使用其 WPF 应用程序模块进行程序开发。

Windows Presentation Foundation(WPF)技术可以编写出独立于平台的应用程序, 借用并扩展了许多技术概念和类, 包括 Windows 窗体、ASP.NET、XML、数据绑定技术和 GDI+等。其显著特点在于清晰地定义了设计和功能之间的界限, 这样, 设计任务和 C#开发任务可以分别独立完成, 而在之前, 要达到这样的灵活程度需要使用高级设计概念或第三方工具。

在 WPF 中, 用户界面设计使用的语言是 Extensible Application Markup Language (XAML)。它类似于 ASP.NET 中使用的语言, 但不限于 HEML 的功能, 允许显卡通过 DirectX 提供的包括浮点数坐标和矢量图变换、2D/3D 高级渲染、字体高级渲染、UI 纹理填充及透明度、动画、可重用资源样式在内的全部高级功能。

WPF 的后台处理任务由 C#语言编写, C#由 C++演化而来, 是 Microsoft 专门为使用.NET 平台而创建的面向对象的程序设计语言。与 C++不同, C#是一种类型安全的语言, 类型转换之间必须遵守严格的规则, 因此 C#代码更加健壮, 调试简单, 尤其适合本章中短期高效的程序开发任务。

5.2.2.功能模块设计

本文系统出于可扩展性的考虑，本文实体链接系统采用模块化设计，将数据预处理，算法与应用分离开来，模块间采用统一的接口和类进行数据交换和存储，方便日后对改进的实体链接系统进行试验试用及功能扩展。本文系统架构如图 5-2 所示，下面分别对各模块功能进行介绍。

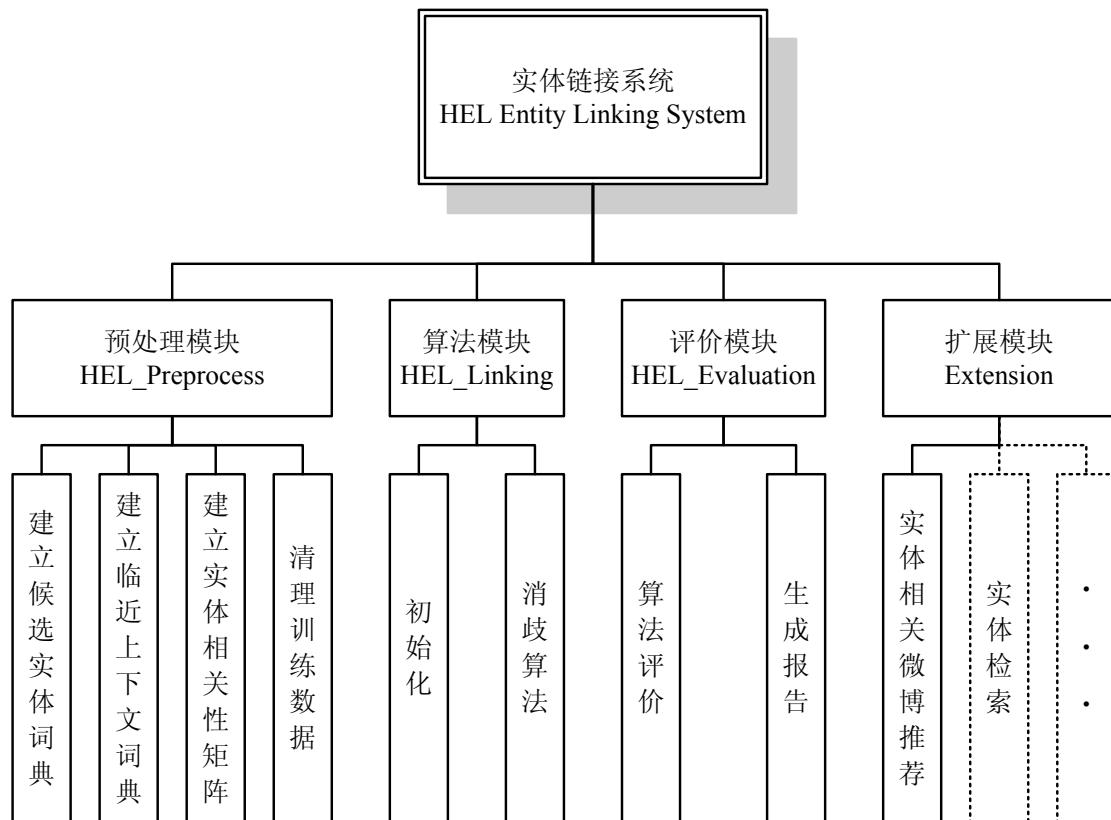


图 5-2 HEL 实体链接系统框架

5.2.2.1.预处理模块

预处理模块主要任务是结合算法需要对知识库和微博文本进行特征采集和数据清洗。

知识库预处理任务包括建立候选实体词典及词频统计，建立临近上下文词典，建立实体相关性矩阵等工作。这些知识都能够通过遍历数据库得到，同时它们都是与算法无关的数据本身的属性，因此预处理模块只需将提取结果存储成独立的特征文件，供算法模块直接读取，无需随程序反复运行。

训练数据预处理主要针对微博文本的清洗。我们所使用的实验数据¹⁵来自真实的 Twitter 用户发布的微博，结合 2.3 中关于微博文本特征的分析，我们在预处理中去除了超链接，@，# 等无意义微博文档部件，对于语言表达中“简称|全称”的形式，只保留指称项的全称。同时我们还注意到，本文所使用的人工标注数据集会将标注者不确定的指称项链接到未知实体（“unknown”），对于这部分训练数据，我们也在预处理阶段予以排除。

5.2.2.2. 算法模块

算法模块是实体链接系统的核心，包括初始化和消歧算法两部分。其中初始化主要在上一步预处理生成特征文件的基础上，完成对包括特征文件、知识库、停用词表、候选实体词典的读取和存储任务。系统中针对“用户→微博→指称项”的三级数据结构建立哈希表进行存储，保证了对全部数据的可访问性。

消歧算法则根据 4.3 中所述层次化实体链接算法流程，首先完成对指称项根据信息函数的排序，并查询候选实体词典得到指称项的候选实体集，进而利用公式(3-6)、(3-10)、(3-12)、(3-13)计算特征相似度，最后使用迭代消歧策略完成实体消歧。系统会记录指称项与每一个候选实体的链接结果，表示成 DisRis 类的一个实例，DisRis 中包括该指称项和实体，消歧过程中产生的个特征的原始相似度，打分公式给出的总分以及其对应的确认实体池。消歧算法的输出表示成一个按照打分排序后的 DisRis 列表。

5.2.2.3. 评价模块

评价模块负责对当前实体链接任务进行分析总结。根据算法模块输出的链接结果列表，从用户层面和全局层面计算 InKB 准确率、NIL 准确率以及全局准确率¹⁶，并生成算法分析报告。下一步，该模块还将集成更多的评价指标和统计功能。

5.2.2.4. 扩展模块

扩展模块在当前实体链接结果的基础上进行应用扩展。软件提供诸如面板、状态栏、选项卡、Expander 等丰富显示资源，开发者可以针对具体需求在微博列表、实体浏览器、信息窗等多处进行二次开发。作为可扩展项目，本文中所介绍的 1.0

¹⁵详见4.4.1中对实验数据的介绍

¹⁶详见4.4.2中对评测指标的介绍

版本只实验性加入实体相关文档推荐功能，用户可以在通过指称项链接查询到某一实体的基础上，查看所有与这条实体有关的微博或用户，从而进一步获取更多相关信息。

5.3 系统实现

在上一节对于系统功能模块的介绍的基础上，本节将结合软件截图进一步说明系统操作流程及界面设计。

5.3.1.软件主界面

软件主界面如图 5-3 所示。顶部菜单栏和工具栏为用户提供包括数据加载、算法运行、设置、生成报告、帮助页面、浏览模式设置等功能入口。界面中央主体区域左边为用户列表，点选用户可以在中央微博列表浏览该用户发布的全部微博。右侧为链接结果表示区，上方为实体浏览器，用于显示链接实体页面的内容，下方为信息窗，用于集中显示与当前链接相关的全部信息。主界面下方为状态栏，用于告知用户当前系统运行状态，引导用户操作。

程序启动后，系统会首先通过状态栏要求用户加载数据（此时算法运行按钮和生成报告按钮均不可使用，如图 5-4a 所示）。数据加载后，用户列表对微博数据按用户分类显示，用户点选某一个用户，微博列表会显示该用户的全部微博条目原文、微博索引数以及其中所包含的指称项（右下）。运行链接算法后，微博列表会将文中指称项显示为超链接文本（如图 5-5）。点击超链接文本，右侧实体浏览器会采用“标题+正文”的格式显示实体链接算法得到的链接结果对应的维基百科内容，如果链接到空实体，则显示“NIL”。

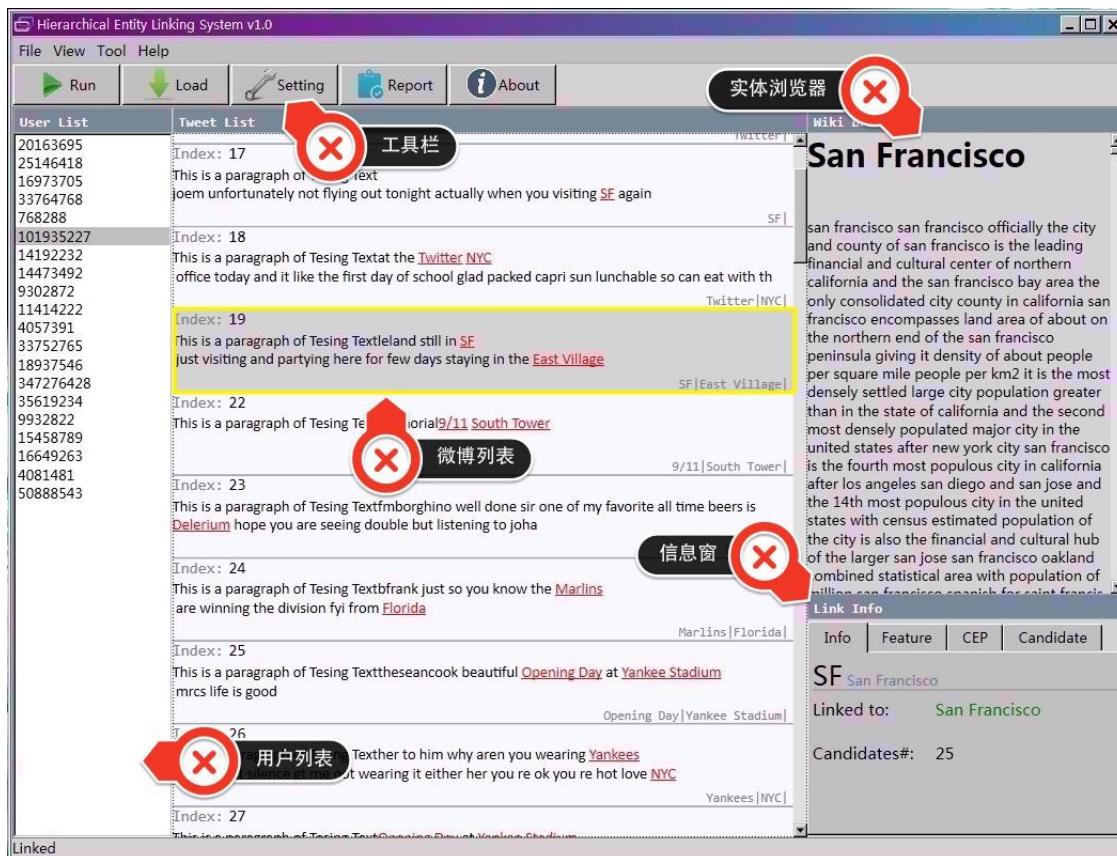


图 5-3 系统主界面

5.3.2.链接信息窗

随着指称项被选中，信息窗同步刷新，实时显示链接信息。信息窗采用标签项（TabControl）的形式，分别显示链接指称项基本信息、链接实体特征值、当前候选实体以及该指称项的候选实体集（如图 5-6）。其中，指称项基本信息选项卡第一行分别显示指称项名称和链接真值实体，第二行显示算法链接结果，这里我们利用 WPF 数据绑定，撰写转换器将链接正确性与字体颜色绑定（如图 5-7），在有限的显示空间内兼顾了信息的完整和布局的简洁。



图 5-4 工具栏显示状态。

上：未加载；中：加载未链接；下：已链接

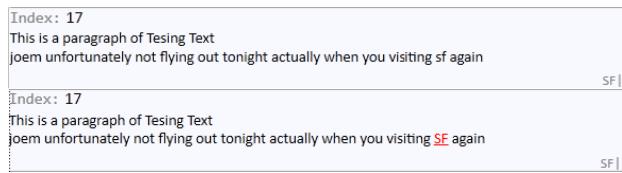


图 5-5 微博列表显示状态。

上：未链接；下：已链接

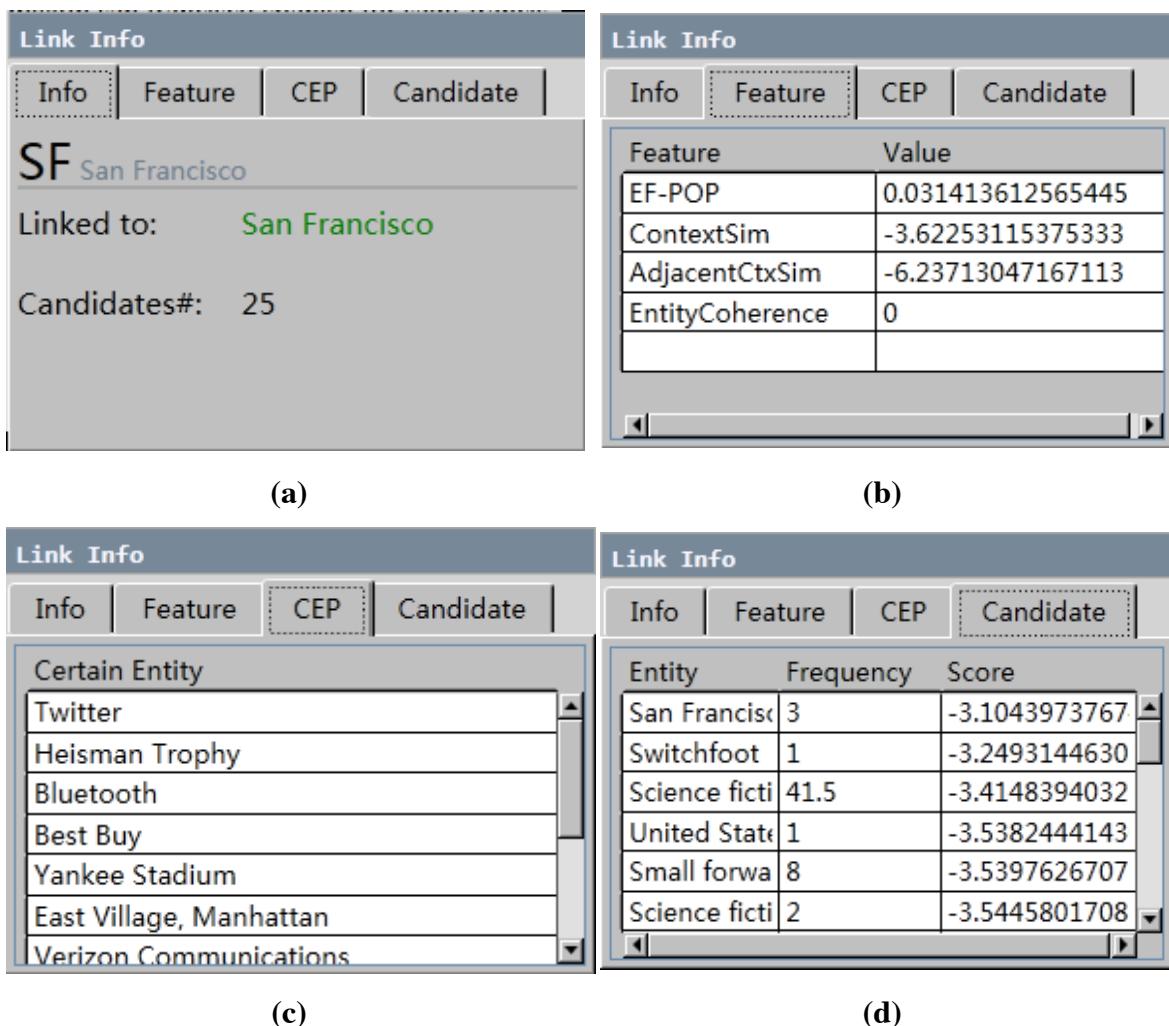


图 5-6 信息窗标签页。

(a) 基本信息(b)特征值(c)确认实体池(d)候选实体集



图 5-7 实体名称颜色与链接正确性数据绑定

5.3.3. 算法设置面板

系统支持对 HEL 算法进行配置，单击工具栏中设置按钮进入设置面板（图 5-8），目前可以对特征权值和候选实体池容量进行设置，如果权重为零，则算法自动不予计算该特征对应相似度，提高效率。

5.3.4. 算法评估报告

链接算法运行后，系统还支持自动生成报告（图 5-9），报告内容包括数据集信息、算法配置信息及三种准确率。系统还提供报告的导出存档功能。

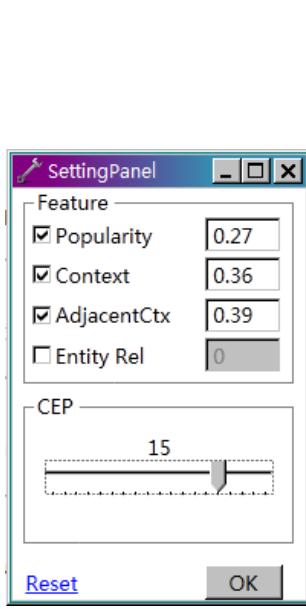


图 5-8 设置面板

ENTITY LINKING REPORT		
04/12/2015		
User#:	20	
Test Sample#:	2677	
NIL#:	438	
In-KB#:	2239	
Popularity:	0.26	
ContextSim:	0.35	
Adjacent CTX Sim:	0.28	
Entity correlation:	0.11	
CEP#:	10	
Metrics	Count	Precision
In-KB Precision	1988	88.79%
NIL Precision	311	71.00%
Total Precision	2299	85.88%

图 5-9 实体链接报告

5.3.5. 增值应用扩展

在得到链接结果之后，我们可以进一步在系统中二次开发基于链接结果的新功能，并将其以插件的形式集成到系统界面中，目前作为扩展模块的功能演示，我们对链接结果建立倒排索引，从而统计出还有哪些微博链接到当前实体，实现实体相关文章的推荐功能（如图 5-10），并将推荐结果以超链接的形式显示于信息窗中。

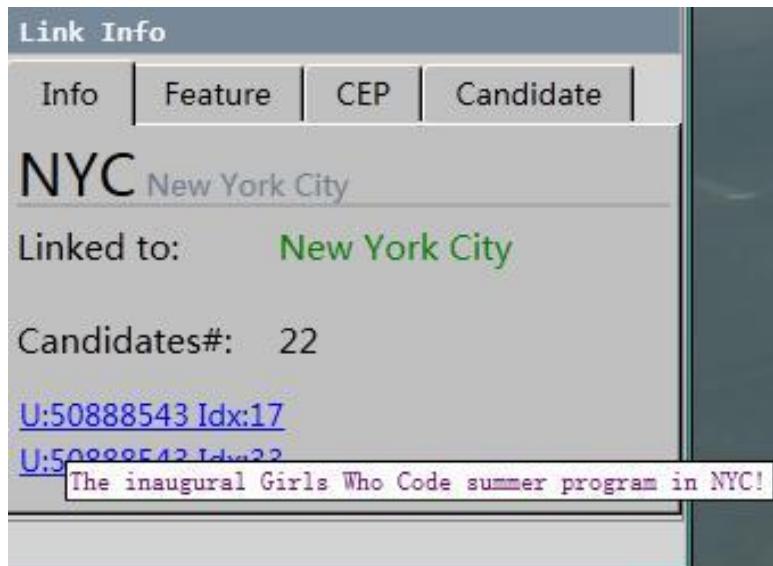


图 5-10 实体相关微博推荐

5.4 本章小结

本章对实体链接原型系统的设计和功能进行了全面的介绍，既是对上一章提出的实体链接算法的总结，也是对实体链接后续工作的引接。

实体链接任务具有很强的应用价值，通过软件设计实现实体链接系统，一方面可以直观有效地观察链接结果，评估算法性能；另一方面可以借助软件平台进行基于链接结果的应用扩展。本文系统的核心算法采用本文提出的 HEL 实体链接算法，在 Visual Studio2010 开发环境下，利用 C#语言在.NET 平台完成后台核心算法及软件功能的设计，并利用 WPF 技术强大灵活的 UI 设计和用户交互完成系统前端设计，前后端通过数据绑定进行数据交换。系统架构上，本文系统基于模块化的设计思路，将预处理、链接算法、算法评价和应用扩展分别独立实现，模块之间具备统一的接口和文件格式，使系统具有算法可移植性和功能扩展性。

试验试用表明，本文所设计的实体链接系统在功能上满足科研需求，实现了实体链接基本功能，并能够对实体链接结果进行有效地组织表达，极大的增强了文本挖掘研究的可读性，并实现了实体相关微博推荐等扩展应用，为日后基于链接结果的二次开发提供了平台。

第六章 总结与展望

本文针对实体链接方法在微博文本集上的应用，结合微博文本特征对实体链接预处理方法进行扩展和改进，并针对传统协同消歧算法存在的问题，提出基于信息函数的层次化实体消歧算法，以提高算法准确率和性能。本章对全文的主要研究内容进行回顾和总结，并对后续的研究工作和方向作出展望。

6.1 全文内容总结

本文从实体链接预处理方法与实体消歧方法两个角度，对当前面向微博文本的实体链接方法进行研究，主要工作内容总结如下：

1. 对传统实体链接预处理方法在微博文本实体链接中存在的不足进行多项改进。首先针对传统词频统计方法对重定向页面统计缺失或重复统计的问题，提出基于均匀分词频的实体流行度，通过对链接到重定向页面中的锚文本进行跟踪，将一次观察记录均分给链接路径上的全部实体，增强了传统流行度特征的准确性和全面性；针对微博文本上下文信息不足的问题，引入临近上下文特征以捕捉实体类别信息，首先为实体建立临近上下文词汇表，进而利用基于朴素贝叶斯的方法对指称项与实体的临近上下文计算相似度，有效地对上下文特征进行降维表示，同时避免了由于微博文本上下文内容较少导致余弦相似度计算中相似度为零的情况，提高了消歧算法在微博短文本中的分辨能力，是对传统基于上下文特征的补充和完善；
2. 针对协同消歧算法模型训练复杂度高、对于微博文本链接准确率低的问题，提出层次化实体消歧算法。该算法将同一用户下的不同消歧任务根据其指称项的模糊程度逐层消歧，并提出利用信息函数对指称项模糊性进行定量计算。本文还创新性利用确认实体池记录历史消歧结果，表征用户偏好，从而利用候选实体与确认实体池的平均实体相关性指导下一层实体消歧任务。实验结果表明，本文算法在降低了模型训练复杂度的同时，保证了在微博文本集上较高的链接准确率，改善了现有协同消歧方法的不足，为其他基于微博文本的实体链接研究提供了可借鉴的思路。
3. 结合本文实体链接算法，设计并实现一套实体链接算法实验应用平台。直观有效地呈现链接结果，评估算法性能，扩展应用场景，有助于对算法应用价值的快速理解和转化。

6.2 下一步工作展望

由于实验条件有限，本文的研究中还存在以下问题有待解决，在后续的工作中需要进一步完善：

1.本文提出的 HEL 算法从本质上讲依旧是一种非监督的消歧实体消歧算法，这种方法的一个弊端是难以进行空实体的判断，目前只能通过训练分类器来对空实体进行分类或者通过实验观察寻找其他特征加以约束。但目前的方法都不具有很强的鲁棒性。另一方面，空实体又是一个模糊的概念，它与知识库的知识量和微博上下文的描述能力有关，理想的情况是微博中提到的每一个名称都应该对应一个知识库中的确认实体。根据以上分析，提高 NIL 的准确率分为两个工作，首先要提高候选实体词典的容量，另一方面要进一步提高消歧算法对于 NIL 的辨别能力；

2.实验中我们注意到基于维基百科的实体流行度统计并不能完全反应实体在微博文本中的流行度，作为百科类网站，维基百科涵盖的内容广、时间跨度长，而且用语较书面化。相反，微博反映了社会最新的关注热点实体，主题倾向于时政、娱乐、体育等大众关注的话题。在实验中我们遇到一些类似的问题，如微博中“SF”一词的候选实体中，“Science Fiction”的流行度大大高于“San Francisco”(尽管“SF”在微博中更常用)。通过学习微博中用词规律，不仅可以得到更为准确的实体流行度，更能够扩充候选实体词典的收词范围，从而提高 NIL 准确率，降低算法的用户敏感性。目前还没有一个统一的用于模型训练的微博数据集，有待进一步的研究。

3.本文算法的效率主要受制于基于 WLM 的实体相似度矩阵的计算，以及候选实体池大小（迭代深度）的选择，下一步的工作将集中在提高预处理的效率和候选实体池准入条件的进一步优化，使得算法能够满足实际工程应用要求。

4.本文中实体链接系统虽然基本实现了预定功能，但尚处于原型开发阶段，各部分功能设计有待完善，下一步主要工作包括增强系统稳定性，并尝试实现其他相关实体链接算法及应用模块，以为系统提供更多算法选择和扩展功能。

参考文献

- [1]Bunescu RC, Pasca M. Using Encyclopedic Knowledge for Named entity Disambiguation[C]. EACL, 2006: 9-16.
- [2]Zhang W, Su J, Chen B, et al. I2r-nus-msra at tac 2011: Entity linking[C]. Proceedings of Text Analysis Conference (TAC 2011), 2011:
- [3]Lin T, Etzioni O. Entity linking at web scale[C]. Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction: Association for Computational Linguistics, 2012: 84-88.
- [4]Chen Z, Tamang S, Lee A, et al. CUNY-BLENDER TAC-KBP2010[J], 2010,
- [5]Zheng Z, Li F, Huang M, et al. Learning to link entities with knowledge base. Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Los Angeles, California: Association for Computational Linguistics, 2010:483-491.
- [6]Lehmann J, Monahan S, Nezda L, et al. LCC approaches to knowledge base population at TAC 2010[C]. Proc. TAC 2010 Workshop, 2010:
- [7]Dredze M, McNamee P, Rao D, et al. Entity disambiguation for knowledge base population. Proceedings of the 23rd International Conference on Computational Linguistics. Beijing, China: Association for Computational Linguistics, 2010:277-285.
- [8]McNamee P. HLT COE efforts in entity linking at TAC KBP 2010[C]. Proc. TAC 2010 Workshop, 2010:
- [9]Guo S, Chang M-W, Kiciman E. To Link or Not to Link? A Study on End-to-End Tweet Entity Linking[C]. HLT-NAACL, 2013: 1020-1030.
- [10]Hoffart J, Yosef MA, Bordino I, et al. Robust disambiguation of named entities in text[C]. Proceedings of the Conference on Empirical Methods in Natural Language Processing: Association for Computational Linguistics, 2011: 782-792.
- [11]Cucerzan S. Large-Scale Named Entity Disambiguation Based on Wikipedia Data[C]. EMNLP-CoNLL, 2007: 708-716.
- [12]Kulkarni S, Singh A, Ramakrishnan G, et al. Collective annotation of Wikipedia entities in web text[C]. Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining: ACM, 2009: 457-466.
- [13]Han X, Sun L, Zhao J. Collective entity linking in web text: a graph-based method. Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval. Beijing, China: ACM, 2011:765-774.
- [14]Ratinov L, Roth D, Downey D, et al. Local and global algorithms for disambiguation to wikipedia[C]. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1: Association for Computational Linguistics, 2011: 1375-1384.
- [15]Shen W, Wang J, Luo P, et al. Linden: linking named entities with knowledge base via semantic knowledge[C]. Proceedings of the 21st international conference on World Wide Web: ACM, 2012: 449-458.
- [16]Shen W, Wang J, Luo P, et al. Linking named entities in tweets with knowledge base via user interest modeling[C]. Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining: ACM, 2013: 68-76.

- [17]Štajner T, Mladenić D. Entity resolution in texts using statistical learning and ontologies[M]. The Semantic Web: Springer, 2009:91-104.
- [18]Shen W, Wang J, Luo P, et al. LIEGE:: link entities in web lists with knowledge base[C]. Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining: ACM, 2012: 1424-1432.
- [19]Han X, Sun L. An entity-topic model for entity linking[C]. Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning: Association for Computational Linguistics, 2012: 105-115.
- [20]Liu X, Li Y, Wu H, et al. Entity Linking For Tweets. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, 2013:1304-1311.
- [21]Guo Y, Qin B, Liu T, et al. Microblog entity linking by leveraging extra posts[C]. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013: 863-868.
- [22]Zhang W, Sim YC, Su J, et al. Nus-i2r: Learning a combined system for entity linking[C]. Proc. TAC 2010 Workshop, 2010:
- [23]Zhang W, Su J, Tan CL, et al. Entity linking leveraging: automatically generated annotation. Proceedings of the 23rd International Conference on Computational Linguistics. Beijing, China: Association for Computational Linguistics, 2010:1290-1298.
- [24]Chen Z, Ji H. Collaborative ranking: a case study on entity linking. Proceedings of the Conference on Empirical Methods in Natural Language Processing. Edinburgh, United Kingdom: Association for Computational Linguistics, 2011:771-781.
- [25]Monahan S, Lehmann J, Nyberg T, et al. Cross-lingual cross-document coreference with entity linking[C]. Proceedings of the Text Analysis Conference, 2011:
- [26]Varma V, Bysani P, Kranthi Reddy VB, et al. iiit hyderabad at tac 2009[C]. Proceedings of Test Analysis Conference 2009 (TAC 09), 2009:
- [27]Han X, Sun L. A generative entity-mention model for linking entities with knowledge base[C]. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1: Association for Computational Linguistics, 2011: 945-954.
- [28]Demartini G, Difallah DE, Cudré-Mauroux P. ZenCrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking[C]. Proceedings of the 21st international conference on World Wide Web: ACM, 2012: 469-478.
- [29]Ji H, Grishman R. Knowledge base population: Successful approaches and challenges[C]. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1: Association for Computational Linguistics, 2011: 1148-1158.
- [30]Han X, Zhao J. Nlpr_kbp in tac 2009 kbp track: a two-stage method to entity linking[C]. Proceedings of Test Analysis Conference 2009 (TAC 09), 2009:
- [31]Milne D, Witten IH. Learning to link with wikipedia[C]. Proceedings of the 17th ACM conference on Information and knowledge management: ACM, 2008: 509-518.
- [32]Zhang W, Sim YC, Su J, et al. Entity linking with effective acronym expansion, instance selection and topic modeling. Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume Three. Barcelona, Catalonia, Spain: AAAI Press, 2011:1909-1914.
- [33]Nemeskey DM, Recski GA, Zséder A, et al. Budapestacad at tac 2010[J], 2010,
- [34]Meij E, Weerkamp W, de Rijke M. Adding semantics to microblog posts[C]. Proceedings of the fifth ACM international conference on Web search and data mining: ACM, 2012: 563-572.
- [35]Guo W, Li H, Ji H, et al. Linking Tweets to News: A Framework to Enrich Short Text Data in Social

- Media[C]. ACL (1): Citeseer, 2013: 239-249.
- [36]Derczynski L, Maynard D, Aswani N, et al. Microblog-genre noise and impact on semantic annotation accuracy[C]. Proceedings of the 24th ACM Conference on Hypertext and Social Media: ACM, 2013: 21-30.
- [37]朱敏, 贾真, 左玲, et al. 中文微博实体链接研究[J]. 北京大学学报: 自然科学版, 2014, 50(1):73-78.
- [38]Jiang L, Yu M, Zhou M, et al. Target-dependent twitter sentiment classification[C]. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1: Association for Computational Linguistics, 2011: 151-160.
- [39]Auer S, Bizer C, Kobilarov G, et al. Dbpedia: A nucleus for a web of open data[M]. Springer, 2007.
- [40]Bollacker K, Evans C, Paritosh P, et al. Freebase: a collaboratively created graph database for structuring human knowledge[C]. Proceedings of the 2008 ACM SIGMOD international conference on Management of data: ACM, 2008: 1247-1250.
- [41]Suchanek FM, Kasneci G, Weikum G. Yago: A large ontology from wikipedia and wordnet[J]. Web Semantics: Science, Services and Agents on the World Wide Web, 2008, 6(3):203-217.
- [42]Fabian M, Gjergji K, Gerhard W. YAGO: A core of semantic knowledge unifying wordnet and wikipedia[C]. 16th International World Wide Web Conference, WWW, 2007: 697-706.
- [43]Singhal A. Introducing the knowledge graph: things, not strings[J]. Official Google Blog, May, 2012,
- [44]Dong X, Gabrilovich E, Heitz G, et al. Knowledge vault: A web-scale approach to probabilistic knowledge fusion[C]. Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining: ACM, 2014: 601-610.
- [45]Hachey B, Radford W, Nothman J, et al. Evaluating entity linking with wikipedia[J]. Artificial intelligence, 2013, 194:130-150.
- [46]Bekkerman R, McCallum A. Disambiguating web appearances of people in a social network[C]. Proceedings of the 14th international conference on World Wide Web: ACM, 2005: 463-470.
- [47]Zesch T, Gurevych I, Mühlhäuser M. Analyzing and accessing Wikipedia as a lexical semantic resource[J]. Data Structures for Linguistic Resources and Applications, 2007:197-205.
- [48]Mihalcea R, Csoma A. Wikify!: linking documents to encyclopedic knowledge[C]. Proceedings of the sixteenth ACM conference on Conference on information and knowledge management: ACM, 2007: 233-242.
- [49]Taneva B, Cheng T, Chakrabarti K, et al. Mining acronym expansions and their meanings using query click log[C]. Proceedings of the 22nd international conference on World Wide Web: International World Wide Web Conferences Steering Committee, 2013: 1261-1272.
- [50]Varma V BV, Kovelamudi S, et al. IIIT Hyderabad at TAC 2009[C]. Proceedings of the Second Text Analysis Conference(TAC 2009): TAC, 2009:
- [51]Chakrabarti K, Chaudhuri S, Cheng T, et al. A framework for robust discovery of entity synonyms[C]. Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining: ACM, 2012: 1384-1392.
- [52]Cheng T, Lauw HW, Paparizos S. Entity synonyms for structured web search[J]. Knowledge and Data Engineering, IEEE Transactions on, 2012, 24(10):1862-1875.
- [53]Gottipati S, Jiang J. Linking entities to a knowledge base with query expansion. Proceedings of the Conference on Empirical Methods in Natural Language Processing. Edinburgh, United Kingdom: Association for Computational Linguistics, 2011:804-813.
- [54]Guo Y, Qin B, Li Y, et al. Improving candidate generation for entity linking[M]. Natural Language

- Processing and Information Systems: Springer, 2013:225-236.
- [55]Deorowicz S, Ciura MG. Correcting spelling errors by modelling their causes[J]. International journal of applied mathematics and computer science, 2005, 15(2):275.
- [56]Jurafsky D, Martin JH. Speech & language processing[M]. Pearson Education India, 2000.
- [57]Chowdhury G. Introduction to modern information retrieval[M]. Facet publishing, 2010.
- [58]Russell S, Norvig P, Intelligence A. A modern approach[J]. Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs, 1995, 25
- [59]Langley P. Selection of relevant features in machine learning[M]. Defense Technical Information Center, 1994.
- [60]Gattani A, Lamba DS, Garera N, et al. Entity extraction, linking, classification, and tagging for social media: a wikipedia-based approach[J]. Proceedings of the VLDB Endowment, 2013, 6(11):1126-1137.
- [61]Ji H, Grishman R, Dang HT, et al. Overview of the TAC 2010 knowledge base population track[C]. Third Text Analysis Conference (TAC 2010), 2010:
- [62]Blei DM, Ng AY, Jordan MI. Latent dirichlet allocation[J]. the Journal of machine Learning research, 2003, 3:993-1022.
- [63]Kataria SS, Kumar KS, Rastogi RR, et al. Entity disambiguation with hierarchical topic models. Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. San Diego, California, USA: ACM, 2011:1037-1045.
- [64]Sen P. Collective context-aware topic models for entity disambiguation. Proceedings of the 21st international conference on World Wide Web. Lyon, France: ACM, 2012:729-738.
- [65]Xu Z, Ru L, Xiang L, et al. Discovering User Interest on Twitter with a Modified Author-Topic Model. Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Volume 01: IEEE Computer Society, 2011:422-429.
- [66]Cilibrasi RL, Vitanyi PM. The google similarity distance[J]. Knowledge and Data Engineering, IEEE Transactions on, 2007, 19(3):370-383.
- [67]Ceccarelli D, Lucchese C, Orlando S, et al. Learning relatedness measures for entity linking[C]. Proceedings of the 22nd ACM international conference on Conference on information & knowledge management: ACM, 2013: 139-148.
- [68]唐博蓉. 基于维基百科的命名实体消歧研究. 北京理工大学, 2011.
- [69]Witten I, Milne D. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links[C]. Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA, 2008: 25-30.
- [70]Herbrich R, Graepel T, Obermayer K. Large margin rank boundaries for ordinal regression[J]. Advances in neural information processing systems, 1999:115-132.

攻读硕士学位期间发表论文情况

1. 李禹恒, 宋俊, 黄宇, 付琨, 吴一戎, 陈昊. 基于微博文本的层次化实体链接方法. 吉林大学学报(工学版) (EI). 已录用.
2. 宋俊, 李禹恒, 黄宇, 陈昊, 付琨. 一种基于用户兴趣的微博实体链接方法. 计算机应用研究. 已录用.
3. Jun Song, Yu Huang, Xiang Qi, **Yuheng Li**, Feng Li, Kun Fu, Tinglei Huang. Discovering Hierarchical Topic Evolution in Time-Stamped Documents. Journal of the Association for Information Science and Technology (SCI). 已录用.

致谢

回首三年的研究生学习生活，收获颇丰，心存感恩。值本文付梓之际，在此向所有给予我学业上指导、生活上帮助、人生道路上启迪的恩师和挚友表达由衷的感谢和美好的祝愿。

感激母校中国科学院大学和中国科学院电子学研究所为我提供这样一个学习平台，能够与学界最优秀的研究员、教授对话，能够与业界最优秀的工程师和同窗共事。“高度决定视野，角度改变观念，尺度把握人生”，国科大丰富的师资与优越的科研条件培养了我广阔的视野，和一颗不甘平庸的心。

首先要感谢恩师吴一戎院士，三年前与吴老师的第一次促膝长谈至今记忆犹新。他让初入科院的我认识到自己已不再是一名普通的学生，而应该如一名科学的研究者一般，严谨、踏实而不失灵活创新：做学问首先要踏实，基本的理论要烂熟于心，才经得住推敲，相关研究要追根溯源，才能高屋建瓴；搞研究也不能死板，选题要明确其应用价值和前景，工作中要不时跳出来，从全局出发审视当前任务的价值和角度。他的教导不仅成为我做科研的首要准则，更深刻地影响了我为人处事的态度，令我受益终生。

衷心感谢恩师付琨研究员，入室两年多的日子里，他不仅在学业上对我悉心指导，更在实验室项目中委我以重任。他鼓励年轻人创新，开放而包容，从他身上我不仅能感受到作为一名学者应有的对待科研的态度，更学习到如何做一名令人尊敬的团队领袖。

感谢黄宇师兄对我的悉心指导，从大四毕设到外场开发再到毕业论文，一路并肩走来，他总是能给予我最中肯的教导和帮助，并如兄弟一般照顾着我。我们在一起聊工作、谈人生、讨论学术，这些言传身教让我终身受用！衷心祝愿他身体早日康复，工作顺利，阖家幸福。

感谢许光銮主任、王宏琦研究员、黄廷磊研究员、高鑫研究员、尤红建研究员、郭智研究员等诸位老师在学位论文各关键阶段对我提出宝贵的意见和指导。同时也感谢葛蕴萍、闫国刚、齐嗣芳、尹锋岩、巩敏、邹颖、张然、沈元春等老师在研究生学习期间给我提供的无微不至的关怀，感谢研究生处卢葱葱老师、王永老师、王

毓萍老师、林飞宇老师以及所长秘书梁伟老师长期的辛勤工作以及在我学习、申请过程中给予的莫大的支持和帮助。

感谢孙显、李以福、李清广、郑国芹、袁文龙、方继飞、李磊、刘昆、宋晶晶、曹玮、郑友华、李毓、韩记伟、李志强、项天远、杨志峰、张天翔、王磊、王义成、陈娟、傅兴玉、张道兵、闫梦龙、陈克明、梁霄、王洋等师兄师姐在实验室项目工作上的指点和帮助，从他们身上我学到了很多知识和经验。

感谢与我同窗三年的同学们，祝齐翔、郑欣慰、刘格、张亚森、张姿、宋俊、田璟、张万层、吴舟婷、王峰、谭洪、关欣、孙康、林雪、侯天怡、张源奔、刘晓燕、黄晓海、熊文昌学长以及李斌、陈丽勇、高君、王福来、张晓、肖新耀、傅翀、李琳、吴亮、张浩龙、张波、朱雪莹等同学工作顺利，祝林宜琛、董文强在异国他乡学业有成。

感谢吴斌、王陈园、毕思泽、刁文辉、张跃、吴凡、窦方正、韩啸宇、彤博辉、罗珞珈、陈板桥、郭岩、曲景影、康丽萍、张文凯等师兄师姐、师弟师妹们陪伴我度过充实愉快的研究生生活，并祝顺利完成学业。

感谢侯天怡、吴斌、张跃、袁文龙、吕磊、李磊、白云鹏、李峰、唐侃、张翰墨、宋晶晶、傅翀、张晓、折小强等羽毛球队的队友，和他们一起驰骋赛场的日子是我科院三年生活中难忘一笔。

感谢所有在求学生涯中关心和帮助过我的老师、同学和朋友，虽然没有一一列出他们的名字，但他们曾经给予的帮助，我将永远铭记在心，感激不尽。

最后，特别感谢我的父母和家人，谢谢他们对我求学路上无条件的支持以及生活上无微不至的关怀。学海无涯，迈出国科大的校门，我将会在更广阔的世界中探寻知识、体验人生，但无论走到哪里，回望故乡，他们都是我最深沉的眷恋。

李禹恒

2015年5月于北京