基于同义实体识别的 Web 信息集成①

徐喆昊, 吴共庆, 胡学钢

(合肥工业大学 计算机系, 合肥 230009)

摘 要: 准确有效地集成海量 Web 信息, 是 Web 信息动态聚合、市场情报分析、舆情分析、商业智能等分析型应用的重要基础. 针对数据集成过程中不同实体指代同一实体的问题, 利用搜索引擎返回的页面摘要信息, 设计并实现了一种基于搜索引擎的同义实体识别算法 FSE, 并提出了一种基于同义实体识别的 Web 信息集成框架. 在医院信息集成测试数据集上的实验结果表明, FSE 算法效果优于基于 VarientDice、VarientCosine、VarientJaccard、VarientOverlap 相似度计算的同义实体识别算法.

关键词: Web 信息集成; 同义实体识别; 相似度计算; 搜索引擎

Web Information Integration Based on Synonymous Entities Recognition

XU Zhe-Hao, WU Gong-Qing, HU Xue-Gang

(Department of Computer Science, Hefei University of Technology, Hefei 230009, China)

Abstract: Integrating massive information on the Web accurately and effectively is the important basis of developing analytic applications, such as Web information dynamic aggregation tools, market information analysis tools, public opinion analysis tools, and business intelligence tools, etc. To solve the problem that different presentations refer to the same entity during the integrating process, this paper proposes an algorithm to recognize the synonymous entities by using the snippets from the search engine and a frame of Web information integration based on synonymous entities recognition. The experimental results on hospital information integration testing data sets show that the proposed method outperforms the synonymous entities recognition based on VarientDice, VarientCosine, VarientJaccard and VarientOverlap.

Key words: Web information integration; synonymous entities recognition; similarity computation; search engine

1 引言

随着互联网的飞速发展, 互联网已逐渐成为信息生产和消费的平台, 丰富的数据资源和方便的查询手段也促使其成为获取信息的重要途径. 由中国互联网协会、中国互联网络信息中心(CNNIC)联合组织编撰的《中国互联网发展报告》(2014)提出, “截至 2013 年 12 月, 中国网页数量为 1500 亿个, 相比 2012 年同期增长了 22.2%.”相比传统数据库中的数据, Web 数据具有多源、海量、异构、动态、不确定等特点, 没有特定的描述模型, 并且每个网站的数据都相互独立, 是一种半结构化的数据. 如何更加高效地管理和利用互联网上的 Web 数据资源, 提取其中用户真正需要的信息, 并以友好的方式展现给用户, 成为 Web 集成领域新的挑战.

通常 Web 数据集成过程可以分为领域模型构建、数据采集、数据抽取、数据融合等步骤[1]. 其中, 数据融合是数据集成过程中的关键步骤, 它决定了数据集成的最终质量. 在数据融合的过程中, 各个数据源对数据的组织形式和描述方式不尽相同, 例如存在别名、缩写、曾用名, 对于这些 Web 数据中存在的大量不同实体指代现实生活中同一实体的现象, 我们称之为同义实体现象. 这些数据的存在势必会影响最终服

① 基金项目:国家高技术研究发展计划(863)(2012AA011005);国家自然科学基金(61273297)
收稿时间:2015-01-20;收到修改稿时间:2015-03-04

务数据的质量和用户的体验, 因此需要识别并合并这部分数据. 同义实体识别是实体统一的一种, 指将现实世界的某一实体的不同描述识别出来并将其合并, 以便用户进一步使用. 通常, 来自不同数据源的同义实体可以相互补充、纠正, 因而经过同义实体识别后的数据, 信息量更为丰富、可信度更高. 准确地识别出数据中的同义实体, 对于去除冗余、提高数据集成度有着重要的作用.

在不同研究领域中, 同义实体识别问题的名称也有所不同, 常见的有: 记录链接[2]、重复检测[3,4]、实体解析[5-7]、名称匹配[8,9]、数据消重[10]、实例匹配[11]等. 传统的同义实体识别方法大致分为五类[4]: 1)基于概率模型的方法[12-14]. 此类方法使用概率化的方法来近似估计一组数据记录匹配的可能性. 2)基于监督及半监督的方法[15,16]. 此类方法利用手工标注的数据集和数据挖掘中常见的分类算法对实体构建分类回归树, 从而达到对实体进行识别的目的. 3)基于主动学习的方法[17]. 监督或半监督的方法需要大量人工标注数据集, 而基于主动学习的方法则只需要少量训练集. 通过主动学习算法, 系统可以自动检测模糊的匹配对, 从而找到同义实体. 4)基于距离的方法[5,18]. 此类方法利用命名实体的属性或关系, 将其转化为衡量实体距离的方式, 再划定阈值来确定两者是否为同义实体. 5)基于规则的方法[19]. 此类方法是基于距离方法的一种特例, 通过等式理论和部分规则来表达相关逻辑, 从而实现同义实体的识别. 传统的同义实体识别方法通常面向结构化数据, 而来自 Web 的数据相对复杂、混乱, 因此传统的同义实体识别技术很难满足加工 Web 数据的要求.

在现实世界中, 对于一些命名实体, 人们很难从字面上判断其是否为同义实体, 此时, 通常人们会选择利用搜索引擎来获取更多相关信息, 再判断其是否为同义实体. 受此启发, 我们尝试通过分析搜索引擎返回的页面摘要(Snippet)信息, 来判别命名实体是否为同义实体.

本文的贡献在于: 1)针对 Web 数据集成过程中出现的同义实体识别问题提出了一种基于搜索引擎页面摘要的同义实体识别方法; 2)设计并实现了一种基于同义实体识别的 Web 数据集成框架.

2　基于搜索引擎的同义实体识别

2.1 基于搜索引擎的相似度计算

本节中我们将阐述基于搜索引擎的命名实体间相似度计算的算法.

在搜索引擎中搜索命名实体(以下简称实体)A 时, 除了可以得到实体 A 的相关信息以外, 也可以得到其他与其同义或相关的实体的信息. 因此我们可以提出假设:

假设 1: 若实体 A 与实体 B 是同义实体, 则有 $A@DB>0$ 或 $B@DA>0$.

其中, DB 为搜索引擎返回的关于 B 的搜索页面摘要集合, DA 为搜索引擎返回的关于 A 的搜索页面摘要集合. $A@DB$ 表示在搜索引擎返回的关于 B 的搜索页面摘要中包含 A 的页面摘要个数. 因而, 假设 1 可以理解为: 对于实体 A 和实体 B, 若实体 A 与实体 B 为同义实体, 则实体 A 的搜索结果中必然存在实体 B 的相关内容或者实体 B 的搜索结果中必然存在实体 A 的相关内容.

从而, 我们可以利用公式 1 来度量实体 A 与实体 B 之间的同义关系强度.

$$R(A,B)=\frac{A@DB+B@DA}{N_{DB}+N_{DA}} \tag{1}$$

其中, N_{DB} 为在搜索引擎中检索实体 B 的结果数, N_{DA} 为在搜索引擎中搜索实体 A 的结果数. 即, 实体 A 与实体 B 的同义关系函数为 DB 中出现 A 的摘要个数与 DA 中出现 B 的摘要个数之和同 N_{DA}、N_{DB} 之和的比值. 当 DB 中 A 出现的摘要个数越多, 或 DA 中 B 出现的摘要个数越多时, 函数值越大, 即实体 A 与实体 B 之间的同义关系越强.

然而在研究中我们发现, 由于搜索引擎的自动纠错、联想等等功能, 部分实体 A 在实体 B 的检索结果页面中出现频率非常大, 而实体 B 在实体 A 的检索结果中基本不出现, 这对计算结果产生了一定的偏差. 因此我们引入了实体 A 与实体 B 的搜索平衡关系式:

设 X 为一命名实体, $P(X@DY)$ 表示 X 在 Y 的搜索结果页面摘要集合 DY 中出现的频率, 即:

$$P(X@DY)=\frac{X@DY}{N_{DY}} \tag{2}$$

其中, $X@DY$ 表示 X 在 Y 的搜索结果页面摘要集合 DY 中出现的页面摘要个数. N_{DY} 表示 DY 中所有页面摘要的

个数. 则实体 A 与实体 B 的搜索平衡关系式为:

$$B(A,B)=\frac{2*P(A@DB)*P(B@DA)}{P(A@DB)+P(B@DA)} \quad (3)$$

公式(3)的计算结果在[0,1]之间, 当实体 A 在实体 B 的搜索结果中出现频率较低或实体 B 在实体 A 的搜索结果中出现频率较低时, 公式 3 所得的值较低; 反之当实体 A 在实体 B 的搜索结果中出现频率较高且实体 B 在实体 A 的搜索结果中出现频率也较高时, 我们认为实体 A 和实体 B 的关系更紧密.

因此, 我们提出衡量实体 A 与实体 B 之间相似度计算公式:

$$Sim(A,B)=\sqrt{R(A,B)*B(A,B)} \quad (4)$$

从公式中可以看出, 当实体 A 的搜索结果中没有包含实体 B 或实体 B 的搜索结果中没有包含实体 A 时, $B(A,B)$的计算结果为 0, 致使 $Sim(A,B)$的值为 0, 即认为两者不相关.

2.2 基于搜索引擎的同义实体发现

利用公式 4 可以计算出两个命名实体 A、B 之间的相似度. 在进行多实体识别的过程中, 我们对所有实体进行两两比较. 由于在实体 A 搜索结果中出现频率最高的实体更可能是 A 的同义实体, 因此, 我们只需取与 A 相似度最高的实体进行判断. 但是, 即使两个实体有一定相似度, 也并非一定是同义实体, 所以我们需要设定阈值来最终判断 A、B 是否为同义实体. 为此, 我们提出了基于搜索引擎的同义实体发现算法 FSE(Find Synonymous Entities), 如图 1.

Algorithm FSE

输入: 命名实体 A, 命名实体集合 S, 阈值 τ
输出: 命名实体 B

```
 1:  DA←ExtractSnippets(A,n);
 2:  max=0, sEntity=null;
 3:  for each x in S {
 4:      Dx=ExtractSnippets(x,n);
 5:      r=R(A,x,DA,Dx,n);
 6:      b=B(A,x,DA,Dx);
 7:      simility=Sim(r,b);
 8:      if(simility>max))
 9:          max=simility;
10:          sEntity=x;
11:      }
12:  }
13:  if (max>τ)
14:      return sEntity;
15:  else
16:      return null
```

图 1 同义实体识别算法 FSE

首先我们利用搜索引擎搜索命名实体 A, 抽取 A 的搜索结果前 n 条页面摘要 DA. 设相似度最大值 max 为 0, 候选实体 $sEntity$ 为空. 遍历命名实体集合, 依次将命名实体 A 与其他命名实体 x 进行比较, 抽取 x 的搜索结果前 n 条页面摘要 Dx, 利用公式(2)、(3)、(4), 计算 A 与 x 的相似度, 当相似度大于 max 时, 令 max 为当前相似度, 同时记录 $sEntity$ 为当前实体 x. 当遍历完所有命名实体后, 取相似度最高的命名实体 $sEntity$, 若 $sEntity$ 与 A 的相似度大于预设阈值 τ, 则认为 $sEntity$ 与 A 为同义实体, 若 $sEntity$ 与 A 相似度不大于 τ 则认为 $sEntity$ 与 A 不是同义实体, 并返回空, 表示未找到 A 的同义实体.

3 基于同义实体识别的Web信息集成系统

相比传统数据集成, Web 数据集成面临的问题更加明显: 1)数据源独立性强、异构性明显. 大多数研究面向结构化数据记录, 缺乏对 Web 数据随意、多样等特点的考虑, 使得这些方法很难在 Web 背景下使用; 2)自治性强, Web 数据的变更频繁, 与集成系统之间并没有直接交互, 一旦发生变更集成系统无法及时更新; 3)冗余量大. 由于数据来源不同, 数据源对数据的描述方式和组织形式不同, 其中势必包含大量冗余信息, 若不对其进行识别处理, 则会大大加重集成系统负担, 更会对集成结果产生影响.

为了解决上述问题, 我们提出了基于同义实体识别的 Web 信息集成系统框架, 旨在对互联网相关的特定资源进行加工整合, 将网络上无结构或半结构化的信息加工融合为结构化数据, 构建独立的信息资源库, 并在此基础上, 可为用户提供信息检索、数据分析、数据挖掘等服务.

3.1 设计架构

本系统设计的基本思想为: 采集领域相关的 Web 服务网站数据, 并对数据进行加工、融合, 包括填补数据缺省信息、处理矛盾内容、合并冗余条目, 形成高质量数据, 为用户提供数据资源与访问接口.

常见的数据集成方式主要包括模式集成法与数据仓库法. 模式集成法大多用于查询集成系统中, 在构建集成系统时, 会将各个数据源的数据视图集成为全局模式, 用户可以直接在全局模式上提交查询请求, 而不必理会各个数据源具体的操作. 查询请求则通过各个不同数据源的包装器转化为针对各个数据源的本

地查询模式. 数据仓库法则是在用户与数据源之间建立中间层, 即将各个数据集的数据复制到同一处, 例如数据仓库, 而用户则像访问单一数据库一样访问数据仓库. 模式集成法获取数据的实时性好, 适合更新频繁的数据源, 但是其数据较为离散, 难以形成完整体系, 不适合后续的分析和挖掘工作, 对于构建市场分析、商业智能等分析型应用来说, 显得捉襟见肘.

系统主要由资源层、融合层、服务层构成, 如图 2.

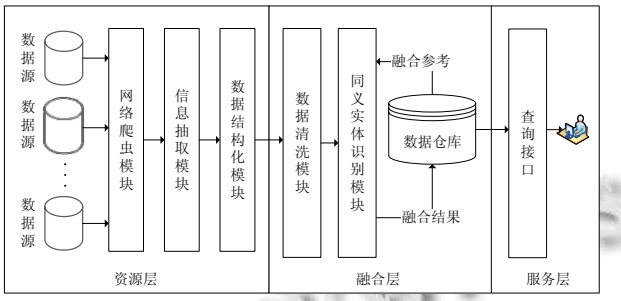

图 2　Web 信息集成系统架构图

资源层主要负责对系统所需的数据资源进行采集和重组织, 完成数据从半结构化到结构化的操作. 首先, 网络爬虫模块从 Web 数据源中采集 Web 页面. 采集的 Web 页面将递交给信息抽取模块, 提取我们需要的信息. 信息抽取模块将我们需要的内容从 HTML 中抽取出来, 随后交给数据结构化模块. 数据结构化模块根据用户定制的系数据模型, 将 Web 页面中的半结构化信息转变为结构化的待融合数据, 随后提交给融合层.

融合层主要负责数据的清洗、加工以及融合. 融合层构建于数据仓库之上, 所有的操作在数据仓库中进行. 在融合层中, 数据首先进行初步清洗, 完成数据格式的统一, 其次进行融合, 去除冗余、补全残缺、修正矛盾. 命名实体识别模块便工作在此层, 用于识别哪些数据指代同义实体, 继而对其进行融合, 提高数据集成度. 识别后的数据会形成同义实体映射表, 存放在数据仓库中, 以便后续的融合操作遇到时能快速识别.

服务层主要提供数据的对外服务业务. 用户可以通过调用数据库连接直接访问融合后的数据资源, 也可以自由定制 Web 服务, 向服务层提交数据请求, 获取系统返回的 XML、JSON 等结构化数据. 服务层需要保证良好的可扩展性, 以便用户进行数据分析、挖掘和应用开发.

3.2 资源层:

资源层以互联网上特定资源相关的网站为获取对象. 这些网站相互独立, 其网页内容、页面风格、模板各不相同.

利用开源的网络爬虫, 我们可以构建爬虫集群. 由主节点分配、调度爬取任务, 从节点负责爬取工作的具体执行. 在主节点上, 需要架设定时器模块, 以便周期性地对网站页面进行爬取, 保证数据及时更新.

由于各个网站页面风格不同, 因而对于不同的网站, 若要提取其中用户需要的信息, 则需要定制个性化的抽取策略. 常见的信息抽取方法有很多种, 包括基于视觉的、基于统计规律的、基于机器学习的、基于模板等方法. 在数据源相对稳定的情况下, 为了让抽取效果较好, 我们选用基于网站模板的抽取策略. 因为每个网站的风格都是相对统一的, 页面的结构也大体相同, 因此利用网页抽取模板我们可以方便准确地抽取需要的数据信息. 我们针对每个网站设计抽取模板, 利用 DOM 树解析爬取的 HTML 文件, 考察需要抽取的数据的相关标签信息, 将其标签层次规律和属性特征设计成规则, 通过这种规则, 可以方便地对该网站相同结构的页面进行抽取.

抽取出信息以后, 为了能够使数据顺利进入数据仓库, 我们需要对数据进行结构化. 根据用户建立的数据模型, 将抽取出的数据转化为结构化数据, 以数据记录的形式传递给融合层进行下一步操作.

3.3 融合层

融合层为系统的核心功能部分, 负责对数据的进一步清洗、加工和融合, 目的是为了获取精确的、完整的、一致的、有效的、唯一的数据. 从 Web 采集来的数据充满噪音和错误, 因此首先在融合前, 我们需要针对这些数据进行大致的清洗. 主要步骤包括:

1)数据格式的清洗. 包括数据类型的转化、格式统一等.

2)数据表与基础数据间关联的清洗. 基础数据指标准化的数据, 这些数据在各个领域都是相对一致的, 例如省市数据等. 通过与基础数据关联可以发现数据中的错误, 并且尽可能地填补数据中空缺.

3)明显错误数据的清洗. 对于乱码等明显有误的数据可以直接予以清除, 防止影响后续融合等操作.

数据清洗可以解决一些明显错误, 但是对于潜在语义层面上的问题却效果不佳, 因此我们需要进一步的融合操作来获取更高质量的数据. 本文引入了同义实体识别技术以解决该问题.

同义实体识别的主要方法是通过计算命名实体之间的相似度来进行判定, 根据给定相似度计算算法, 计算两个实体之间的相似度, 划定相似度阈值, 当满足阈值后, 即认为两者为同义实体. 因此同义实体的识别本质上也可以认为是一种基于命名实体间相似程度衡量的识别.

因此, 融合层中的数据融合操作主要通过对同义的命名实体进行识别, 针对新进数据, 识别其同义实体并进行融合, 以达到去重、补全、修正的目的.

图 3 为系统数据融合模块的流程图, 融合的主要步骤如下:

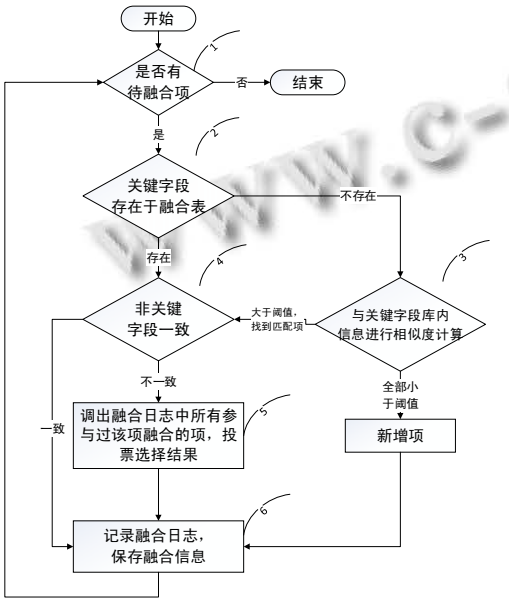

图 3 数据融合流程图

步骤 1: 首先判断是否有待融合数据, 若有待融合数据则对待融合数据进行融合操作, 执行步骤 2, 若无, 则结束融合操作.

步骤 2: 由用户确定待融合数据的关键字段. 关键字段中的关键词是融合的参照标准, 它是系统判断待融合数据是否需要融合、与谁进行融合的评判依据. 它类似于主键, 在融合表(完成融合后的表)中具有唯一性. 确定待融合数据关键字段后, 我们需要判断待融合数据中的关键词是否已经存在于融合表的关键字段中. 若不存在则执行步骤 3; 若已存在, 则称此融合表中与待融合数据关键词相同的数据为融合对象数据, 执行步骤 4 对非关键字段进行融合操作.

步骤 3: 将待融合数据关键词与融合表中相关字段的关键词进行比较, 我们可以得到关键词与融合表中其他关键词的相似度. 通过设置阈值, 我们可以进行判断, 若关键词与融合表中相似度最高的关键词的相似度值高于阈值, 则认为两者为同义关键词, 并与此关键词所属数据项进行融合, 执行步骤 4; 若关键词与融合表中相似度最高的关键词的相似度值仍低于阈值则认为该关键词不同于融合表中的其他关键词, 作为新数据插入融合表中, 执行步骤 6 记录融合日志.

步骤 4: 当我们选定了融合对象数据后, 需进一步比较其与待融合数据的非关键字段是否一致, 若一致则不对融合数据进行修改, 直接执行步骤 6 记录融合日志; 若不一致, 则执行步骤 5.

步骤 5: 读取融合日志, 取出之前参与融合的数据项进行比较, 通过投票等策略决定是维持融合表中数据不变还是修改为待融合数据中的数据, 随后执行步骤 6 记录融合日志.

步骤 6: 记录融合过程, 包括待融合数据的 ID 信息、融合后数据的 ID 信息、融合方法、融合时间等, 对于参与了融合的关键词可以加入系统的词典, 以便后续融合操作. 返回步骤 1, 继续检查是否有待融合数据.

3.4 服务层

服务层负责对用户提供服务, 是用户获取数据资源的接口. 系统可以通过 Web Service、HTML 页面、数据库接口等方式为用户提供融合后的数据查询服务. 用户也可以利用融合后的数据资源, 在其上进行数据分析、数据挖掘等二次开发.

4 实验与评估

同义实体的识别效果直接决定了数据融合的最终效果. 因而, 本文提出的同义实体识别算法 FSE 是 Web 信息集成系统的核心部分, 本节将重点介绍 FSE 算法实验的详细内容, 包括数据集、实验评价标准、实验流程和实验结果. 实验过程中的硬件配置为: Intel(R) Core(TM) i5-3470S CPU @ 2.90GHZ, 2.90GHZ, 8GB 内存, 操作系统为 Windows 7 64bit.

4.1 实验数据集

实验数据集来源于国家 863 计划课题“多源异构数据集成与挖掘的关键技术”的示范应用系统 “普适医疗系统”(http://210.45.241.181/health/)收集的数据. 该系统针对医疗领域, 利用网络爬虫获取 Web 数据, 并运用结构化提取、数据集成和分析管理等关键技术, 构建海量医疗信息的结构化管理. 项目实施过程中,

我们利用课题组研制的 JAVA 开源爬虫 Web Collector(Github 地址: https:// github.com/CrawlScript/WebCollector)作为信息采集引擎对好大夫在线、39 健康网、全民健康网等医疗网站进行医院信息采集, 使用 JSOUP 解析 Web 页面, 并抽取相关信息, 构建原始数据库.

在本文实验部分, 我们选取了医院数据集中所有北京医院的数据, 作为命名实体数据集, 所有的数据全部来自于真实世界的各个网站. 其中命名实体共有 317 个, 通过人工比对, 整理出其中异义的实体(不同的实体)共有 176 个.

我们选用国内应用最多的百度搜索引擎获取命名实体的搜索结果, 利用 JAVA 开源轻量级网页爬取工具 HttpClient 来采集搜索结果页面, 并使用 JSoup 解析搜索结果页面. 通过设计页面摘要的特征路径, 我们可以从页面中提取命名实体搜索结果的页面摘要, 从而生成所有命名实体的页面摘要集合.

4.2 评价标准

本文参考了文本检索领域常用的精度、召回率和 *F* 值标准对同义实体识别算法的效果进行评估.

F 值标准对同义实体识别算法的效果进行评估.

设 S_e 为算法计算出的同义实体关系对集合, S_l 为标准同义实体关系对集合. 则我们定义精度、召回率和 *F* 值如下:

$$P=\frac{|S_e \text{ I } S_l|}{|S_e|} \tag{5}$$

$$R=\frac{|S_e \text{ I } S_l|}{|S_l|} \tag{6}$$

$$F=\frac{2\times R\times P}{P+R} \tag{7}$$

精度(*P*): 计算正确的同义实体关系对占计算出的同义实体关系对的比例.

召回率(*R*): 计算正确的同义实体关系对占标准同义实体关系对的比例.

F 值(*F*): 精度与召回率的综合指标.

4.3 实验结果

为了比较 FSE 算法的识别效果, 我们选取了比较算法 VariantDice、VariantCosine、VariantJaccard、VariantOverlap. 这 4 个算法是由 Chen H H[20]提出的利用搜索引擎获取的搜索页面摘要计算文本之间相似度的算法.

VariantDice:

$$Sim(X,Y)=\begin{cases}0 & if\ f(Y@X)=0\ or\ f(X@Y)=0\\ \dfrac{f(X@DY)+f(X@DY)}{f(X)+f(Y)} & \text{Otherwise}\end{cases} \tag{8}$$

VariantCosine:

$$Sim(X,Y)=\frac{\min(f(X@DY),f(X@DY))}{\sqrt{f(X)+f(Y)}} \tag{9}$$

VariantJaccard:

$$Sim(X,Y)=\frac{\min(f(X@DY),f(X@DY))}{f(X)+f(Y)-\max(f(Y@DX,f(X@Y))} \tag{10}$$

VariantOverlap

$$Sim(X,Y)=\frac{\min(f(X@DY),f(X@DY))}{\min(f(X),f(Y))} \tag{11}$$

其中, *f*(*X*)表示在 *X* 的前 *N* 个搜索结果页面摘要中 *X* 出现的次数, *f*(*Y*)表示在 *Y* 的前 *N* 个搜索结果页面摘要中 *Y* 出现的次数. *f*(*X*@*Y*)表示在 *Y* 的前 *N* 个搜索结果页面摘要中 *X* 出现的次数, *f*(*Y*@*X*)表示在 *X* 的前 *N* 个搜索结果页面摘要中 *Y* 出现的次数.

在识别同义实体时, 我们分别利用这四种算法来计算命名实体的相似度, 手工调整阈值, 将大于阈值的实体视为同义实体, 并将 FSE 算法与这四种算法的识别结果进行比较.

表 1 显示了分别使用 100 条、200 条、300 条、400 条、500 条页面摘要时, VariantDice、VariantCosine、VariantJaccard、VariantOverlap 与 FSE 算法在实验数据集上比较的结果. 每个部分的最高值使用黑色粗体标出.

表 1　对比实验结果

指标	摘要条数 / 算法	100	200	300	400	500
P	VariantDice	94.71%	91.21%	86.99%	87.35%	86.69%
	VariantCosine	95.07%	91.89%	86.72%	87.87%	86.02%
	VariantJaccard	91.34%	**96.04%**	**94.29%**	**92.96%**	**91.94%**
	VariantOverlap	95.07%	91.89%	86.72%	87.87%	86.02%
	FSE	**95.58%**	94.40%	93.86%	92.27%	91.42%
R	VariantDice	91.10%	92.37%	**91.85%**	91.85%	**92.27%**
	VariantCosine	85.78%	89.47%	91.27%	91.30%	90.63%
	VariantJaccard	90.95%	85.46%	86.84%	87.22%	86.22%
	VariantOverlap	85.78%	89.47%	91.27%	91.30%	90.63%
	FSE	**91.53%**	**92.80%**	**91.85%**	**92.27%**	91.81%
F	VariantDice	92.87%	91.79%	89.35%	89.54%	89.40%
	VariantCosine	90.19%	90.67%	88.94%	89.55%	88.26%
	VariantJaccard	91.14%	90.44%	90.41%	90.00%	88.99%
	VariantOverlap	90.19%	90.67%	88.94%	89.55%	88.26%
	FSE	**93.51%**	**93.59%**	**92.84%**	**92.27%**	**91.61%**

从表 1 中我们可以看出, 当使用 100 条页面摘要时, FSE 有着比其他算法更高的精度、召回率和 F 值. 当使用 200 条页面摘要时, FSE 算法的召回率高于其他算法中最高的 VariantDice 算法 0.43%, 高于最低的 VariantJaccard 算法 7.34%; 而 F 值达到了数据集最高的 93.59%, 高于其他数据集中最高的 VariantDice 算法 1.8%, 高于最低的 VariantJaccard 算法 3.15%. 当页面摘要条数达到 500 条时. FSE 算法各项指标依然稳定在 90% 以上, 同时 F 值高于其他算法中最高的 VariantDice 算法 2.24%, 高于其他算法中最低算法 VariantOverlap 和 VariantCosine 的 3.38%.

从表 1 中我们还可以看出 FSE 算法的精度、召回率和 F 值都在 90%以上, 正如我们期望, FSE 算法有着良好的稳定性. 同时, FSE 算法的 F 值与召回率在所有算法中都是最高的. 其在 100 至 500 条页面摘要上的平均精度比其他算法的平均值高 2.24%; 召回率比其他算法的平均值高 2.24%; F 值比其他算法的平均值高 2.31%.

图 4 为各算法使用 100-500 条页面摘要数的 F 值比较. 由于 VarientCosine 和 VarientOverlap 的 F 值完全相同, 所以两条曲线在图中重合. 从图中还可以看出, 当使用 200 条短摘录时, FSE 算法达到最高值, 另外我们还可以看出 FSE 算法的 F 值曲线要明高于其他算法. 当使用 300 条以上的页面摘要时, 所有算法的 F 值都开始下降, 但是 FSE 下降的速率较低, 因此相比其他算法, 它有更好的稳定性.

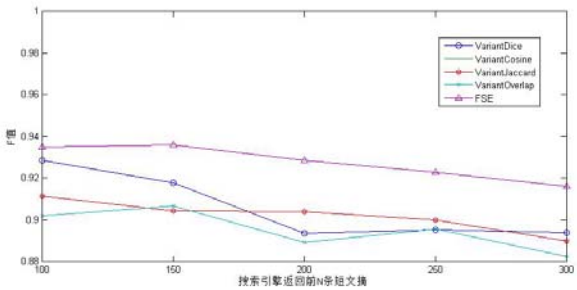

图 4 各算法使用不同页面摘要数的 F 值比较

图 5 为各算法使用 100-500 条页面摘要的时间(毫秒)比较. 从图中可以看出, 在实际数据集上, FSE 算法所用时间约为其他四种对比算法的 1.5 倍, 这是由于平衡关系式的运算计算量较大, 导致了数据处理效率的降低. 虽然 FSE 耗时较多, 但是可以认为 FSE 与其他几种对比算法在时间开销上处于同一数量级. 综上, FSE 算法更适合用于离线分析.

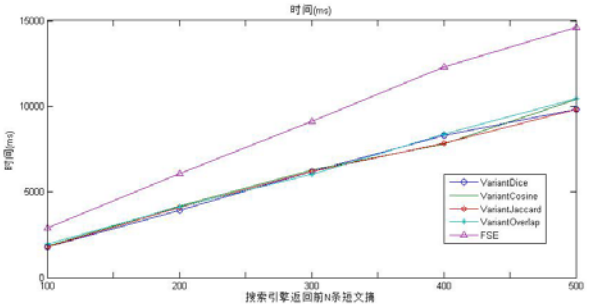

图 5 各算法使用不同页面摘要数的时间比较

4.4 讨论

从图 4 中可以看出, 当使用页面摘要数增加时, 各算法的 F 值下降. 这是因为随着页面摘要个数的增加, 搜索引擎的返回结果会引入更多的联想内容, 而这部分引入的数据都由搜索引擎算法决定. 例如, 对于一些信息量较少的医院命名实体, 搜索引擎会提供许多相关或相似名称的医院信息. 对于用户来说, 搜索引擎这一行为可以提高用户体验, 提高搜索的命中率, 但是对于识别算法来说, 它增加了噪音数据, 导致产生错误的判断结果. 因此对于此类同义实体识别算法而言, 使用的搜索摘要数量大并不一定意味着更精确, 相反, 少量的、排序靠前的数据对于命名实体识别任务而言可能更为有用.

表一显示, FSE 有着较高的 F 值. 这是因为 FSE 对于部分具有包含关系的命名实体有着较好的识别效果, 例如, “北京邮电医院”、“北京协和医院”和“北京协和医院(西院)”这三家医院实体. 通过搜索引擎返回的结果我们了解到, “北京邮电医院”现更名为“北京协和医院(西院)”. 理论上来说, 由于“北京邮电医院”与“北京协和医院(西院)”指代的是同一个命名实体, 因此应该得到最高的相似度. 但是我们发现, 除 FSE 外的相似度计算算法在计算“北京邮电医院”与“北京协和医院”时, 反而会得到更高的值. “北京邮电医院”虽然是“北京协和医院”的分院, 但是其地址、电话、医生、科室等等信息都不相同, 在数据融合时, 应视为两个不同的命名实体, 因此这种现象显然是错误的. 经过我们对实验数据的仔细分析, 发现“北京协和医院(西院)”中包含完整的“北京协和医院”字符串, “北京邮电医院”的搜索结果页面摘要中出现的“北京协和医院”其实有部分是“北京协和医院(西院)”, 但相似度计算过程中, 这部分内容也被计入“北京协和医院”的频率中,

致使“北京协和医院”得到比“北京协和医院(西院)”更高的相似度. 而采用 FSE 算法后, 因为“北京邮电医院”在“北京协和医院”的搜索结果中出现的次数比在“北京协和医院(西院)”少, 因而平衡关系式得到了一个较低的值, 从而降低了与“北京协和医院”的相似度, 使得我们可以获取正确的结果.

5 总结与展望

本文提出了一种三层式 Web 信息集成融合的框架, 包括: 资源层、融合层和服务层. 针对在 Web 信息融合的过程中遇到的命名实体无法判别是否为同义实体的情况, 提出了一种基于搜索引擎的同义实体识别算法.

然而, 我们的算法仍存在改进空间. 在命名实体识别过程中, 我们对数据集中的命名实体采用两两比较的方法来计算相似度, 这无疑是一种计算量庞大并且时间复杂度较高的方法, 对于非常庞大的数据集来说, 会非常耗费资源. 在后续的研究中, 希望能找到一种方法, 能够快速筛选出候选的同义实体, 在不降低识别效果的前提下, 减少计算量.

参考文献

1 Yan Z, Li Q, Zhang S, Peng Z, Dong Y, Ding Y, Zhang Y, Xu X. MI-WDIS: web data integration system for market intelligence. Proc. of the 19th ACM International Conference on Information and Knowledge Management. ACM. 2010. 1957–1958.

2 Christen P. A survey of indexing techniques for scalable record linkage and deduplication. IEEE Trans. on Knowledge and Data Engineering, 2012, 24(9): 1537–1555.

3 Draisbach U, Naumann F, Szott S, Wonneberg O. Adaptive windows for duplicate detection. Proc. of 28th International Conference on Data Engineering. IEEE. 2012. 1073–1083.

4 Elmagarmid AK, Ipeirotis PG, Verykios VS. Duplicate record detection: A survey. IEEE Trans. on Knowledge and Data Engineering, 2007, 19(1): 1–16.

5 Bhattacharya I, Getoor L. Collective entity resolution in relational data. ACM Trans. on Knowledge Discovery from Data (TKDD), 2007, 1(1): 5.

6 Christen P, Gayler R, Hawking D. Similarity-aware indexing for real-time entity resolution. Proc. of the 18th ACM Conference on Information and Knowledge Management. ACM. 2009. 1565–1568.

7 Singla P, Domingos P. Entity resolution with markov logic. Proc. of 6th Int. Conf. on Data Mining. IEEE. 2006. 572–582.

8 Christen P. A comparison of personal name matching: Techniques and practical issues. Proc. of 6th IEEE Int. Conf. on Data Mining. IEEE. 2006. 290–294.

9 Liu J, Lei KH, Liu JY, et al. Ranking-based name matching for author disambiguation in bibliographic data. Proc. of the 2013 KDD Cup 2013 Workshop. ACM. 2013. 8.

10 Jiang Y, Lin C, Meng W, Yu C, Cohen AM, Smalheiser NR. Rule-based deduplication of article records from bibliographic databases. Database, 2014, 2014: bat086.

11 Castano S, Ferrara A, Montanelli S, et al. Ontology and instance matching. Knowledge-Driven Multimedia Information Extraction and Ontology Evolution. Springer Berlin Heidelberg, 2011: 167–195.

12 Newcombe HB, Kennedy JM, Axford SJ, James AP. Automatic linkage of vital records. Proc. of the Workshop on Exact Matching Methodologies, Arlington, Virginia, 1985.

13 Singla P, Domingos P. Entity resolution with markov logic. Proc. of 6th Int. Conf. on Data Mining. IEEE. 2006. 572–582.

14 Wang X, Sun A, Kardes H, et al. Probabilistic estimates of attribute statistics and match likelihood for people entity resolution. Proc. of IEEE Int. Conf. on Big Data. IEEE. 2014. 92–99.

15 On BW. Data Cleaning Techniques by means of Entity Resolution [Ph.D. Thesis]. PA, USA: The Pennsylvania State University, 2007.

16 Bhattacharya I, Getoor L. A Latent Dirichlet Model for Unsupervised Entity Resolution. Proc. of the 6th SIAM Int. Conf. on Data Mining. SIAM. 2006, 124. 47.

17 Sarawagi S, Bhamidipaty A. Interactive deduplication using active learning. Proc. of the 8th ACM SIGKDD int. conf. on Knowledge discovery and data mining. ACM. 2002. 269–278.

18 Köpcke H, Thor A, Rahm E. Evaluation of entity resolution approaches on real-world match problems. Proc. of the VLDB Endowment. 2010, 3(1-2). 484–493.

19 Whang SE, Garcia-Molina H. Entity resolution with evolving rules. Proc. of the VLDB Endowment, 2010, 3(1-2): 1326–1337.

20 Chen HH, Lin MS, Wei YC. Novel association measures using web search with double checking. Proc. of the 21st Int. Conf. on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics. Association for Computational Linguistics. 2006. 1009–1016.