密级：　　保密期限：

北京郵電大學

硕士研究生学位论文

题目：面向行业的信息融合原型系统的研究与实现

学　　号：106901

姓　　名：林哲

专　　业：计算机科学与技术

导　　师：吴国仕

学　　院：软件学院

2012 年 12 月

独创性（或创新性）声明

本人声明所呈交的论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京邮电大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名：_____ 日期：_____

关于论文使用授权的说明

本人完全了解北京邮电大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属北京邮电大学。学校有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存、汇编学位论文。

本学位论文不属于保密范围，适用本授权书。

本人签名：_____ 日期：_____

导师签名：_____ 日期：_____

面向行业的信息融合原型系统的研究与实现

摘 要

随着信息产业的不断飞速发展壮大，网络上的数据每天都在以惊人的速度不断的增长。用户越来越多的在查询中包含实体的信息，例如人名、机构名、地点，试图通过围绕实体来构建有意义的查询条件，从语义的方面查找到与这些实体相关的信息，而不仅限于通过关键词的文档搜索。基于文档级的索引的通用搜索引擎，例如谷歌、百度、雅虎等，都是基于关键词匹配的文档检索，在一定程度上已经开始不能满足互联网用户的搜索需要，人们期望以实体为中心的搜索系统的出现。

本文调研了上述搜索引擎的不足以及用户搜索的习惯，提出了基于实体模型的信息融合方法，通过机器学习构建面向行业的网页信息融合原型系统，以实体为中心将信息进行融合，目的在于利用实体的概念将信息以实体为中心集成起来，更方便于普通互联网用户有效的进行以实体为中心的搜索。

本文主要进行的研究工作如下：首先，基于百度百科，通过词条的抽取、分类、整理，得到一个基于 IT 行业领域的实体词典。其次，收集各大门户网站中的 IT 新闻文本以及 IT 行业中知名博客中的博文，通过网页抽取技术，整理并构建了面向行业的中文新闻领域的语料库。然后，通过机器学习的方法构建面向行业的网页信息融合原型系统，利用基于图的排序算法计算出文本与实体的相关度，在语义理解的基础上得到文本中实体的权重，并根据实体在所出现的文本的权重计算出实体间的关联度。最后，在上述研究基础上，完成一个以实体为中心的搜索系统原型。

本文在系统的实验中，使用已经构建好的基于中文新闻领域的语料库作为测试集，对该面向行业的信息融合原型系统进行了测试，实验结果表明，通过与人工标注的实体关联度进行对比，本文所构建的实体模型中，文本与实体的相关度以及实体间的关联度与人工标注的结果偏差大部分小于 0.1，计算结果与人们的认知结果基本吻合，具有较高的准确率。

关键字：信息融合 实体模型 面向行业 机器学习

RESEARCH AND IMPLEMENTATION OF INDUSTRY-ORIENTED INFORMATION INTEGRATION PROTOTYPE SYSTEM

ABSTRACT

The data available on the Internet is growing rapidly at a tremendous rate due to the rapid development in information industry and more search containing information about entities, such as names of persons, cooperation and place, are conducted by users. They are trying not only to conduct the search by keyword matching, but also by building search conditions from semantic analysis of those entities and related information.

The existed general document search engines, like Google, Baidu and Yahoo, are all using keyword match operations to fulfill users' need. However, these technologies are being found lack of satisfaction from Internet users and an entity centred search engine is in need.

This paper firstly investigated the disadvantages of the existed search engines and users' customs, purposed a method for information integration based on entity model, then build an industry-oriented information integrating prototype system with the help of machine learning algorithms to integrate information around entity concepts in order to make ordinary Internet users use this entity-based search engine more efficiency.

This paper has conducted the following research work: First, we made an entity dictionary based on extraction, classification and sorting entries from Baidu Baike. Second, we collected IT news and famous blogs of IT industry from portal sites and make industry-oriented Chinese news corpus after extracting and sorting those passages. Then, an industry-oriented web information integrated prototype system is made based on machine learning algorithms, which uses sorting algorithms form map to calculate the correlation between text and entity and result in entity weighting in texts on semantic bases; in addition, the correlations between entities are calculated based on the texts containing each entity and its weight. After all, an entity centred prototype search engine is made beyond the above research.

This paper contains experiments of industry-oriented information integrated prototype system made by using an existed Chinese news corpus as a test set. The results has shown that the model presented in this paper has a deviation of less than 0.1 about correlation between text and entity and between entities compared to hand-annotated results, which is anastomosed to human's cognitive and thus have a good accuracy.

KEY WORDS：Information Integration, Entity Model, Industry-oriented, Machine Learning

目 录

第一章 绪论

1.1 课题背景

近几十年来，计算机网络的飞速发展和信息化的推进，使得人类社会所积累的数据量已经超过了过去 5000 年的总和。随着互联网的蓬勃发展，网络也已经成为媒体传播的重要载体之一，其内容与形式也日益丰富。随着 Internet 的迅猛发展，与传统的信息资源相比，Web 已经成为全球传播与共享科研、教育、商业和社会信息等最为重要、最具潜力的巨大信息来源。

Web 上信息资源庞大性的特点，特别是近几年以来，互联网上的信息以惊人的速度在不断的膨胀，各种各样的网络信息混杂在一起使得 Web 上的信息资源不能被有效利用。在面对如此海量的信息时，一般的互联网用户要想充分利用这些信息资源，这就需要对 Internet 上的信息进行整理与归类，否则面对如此海量的信息时，用户很难快速找到自己感兴趣的信息。

实现网页数据共享，可以使更多的人、更充分地利用已有的网络资源，减少资料收集、数据采集等重复劳动和相应费用。然而，这些信息都存储或者发布在许多不同的数据源之中，为了更加有效地利用这些信息，需要从多个分布、异构和自治的数据源中集成数据。而且网络信息更新速度很快，各种新的事物、新的知识层出不穷，网络信息的这些突出的特点就要求网络信息分类体系具有有效跟踪信息动态发展的能力，以某种特征将信息通过算法加以整理。搜索引擎正是通过文档级别的索引加上巨大无比的计算能力，为信息需求者提供一个庞大的结果集，用户更快的查找到相关的信息。

当前的通用搜索引擎文档级别的索引因为提供的结果集太过于庞大而泛泛，不能精确满足用户的需求，这些缺点也越来越多的显现出来，谷歌、百度等通用搜索引擎“搜索框+关键词”的传统模式已经开始不足以满足网络用户对日常信息的搜索需要。

许多研究者已经对上述的这些问题进行了探索性的研究工作，提出了许多解决此类问题的方案与途径，其中就有研究者基于实体展开研究，也取得了一些进展。

1.2 研究意义

在众多的网络信息中，实体作为网络文本中承载信息的重要信息单位，负责

将网络上诸多信息的组织，例如某个公司召开了以某个科技发展为主题的会议等等。毋庸置疑，使用计算机技术对实体进行信息融合，将网络信息以实体为中心组织起来，对网络信息的利用与发展有着重大而深远的意义。

本文基于当前互联网的研究现状试图开展以实体为中心的信息集成与信息搜索方面的框架研究，试图使用信息融合方面的技术，设计一套完整的信息融合系统的原型，将一个行业中的新闻文本基于实体融合起来，使普通的互联网用户对信息可以进行实体相关的检索，从而提高用户对信息的利用率。

信息融合技术研究的是如何加工、综合来自于多个信息源的不同信息，并能将这些不同形式的信息进行相互补充、整合，使其信息量达到最大限度的发挥。使用信息融合技术对网络信息的取舍和集合划分后应用于检索系统，可以更加合理的将查询结果组织起来，这样更有利于网络信息的关联和有效利用。

1.3 国内外研究现状

信息融合技术始于 70 年代初，80 年代以后得到迅速发展，是一种综合利用多种信息资源，以获得对某一事物更客观、更本质认识的信息处理技术[4]，主要研究的是如何加工、综合来自于多个信息源的不同信息，并能将这些不同形式的信息进行相互补充、整合，使其信息量达到最大限度的发挥。信息融合技术通过对信息的取舍和集合划分后应用于检索系统，可以更加合理的将查询结果组织起来，将不同信息源的信息连接成为一个信息完备的有机整体，为用户提供各种信息资源查询服务。

搜索引擎是当前互联网用户使用最多、最频繁的web检索工具之一。通用搜索引擎是基于网络蜘蛛(crawler)的，即互联网用户日常所用到的百度[5]、Google[6]，这类搜索引擎，是一种较为浅层的网络资源集成技术。它由一个被称为网络蜘蛛的机器人程序自动的、根据html网页超链结构在互联网中爬行并发现信息并获取静态网页信息，抓取下来的网页信息交由索引器来建立基于文档级别的全文索引，最后根据用户输入的查询关键词查询结果返回给用户[53]。搜索引擎的优点是更新及时、搜索信息量大、维护费用低、自动化程度高[52]，但是其缺点也显而易见，即返回信息量太大、无关信息太多，这就要求进行资源查询的用户从结果中筛选所需要的信息。

近年来，网络信息爆炸式的增长，以及研究人员对搜索引擎研究的深入，促进了垂直搜索引擎、基于主题的搜索引擎以及个性化搜索引擎技术的发展。网络信息的整合开始朝着特定信息的类型、特定的主题与领域以及特定用户需求的方向发展。例如，INFOMINE[7]采用主题搜索引擎技术，通过广泛的采集、整合网络上学术信息的相关资源，建立了包括生物农业和医学、经济与贸易等9个学科

系列的网络信息资源数据库，形成了一个较为完善的学术资源集成检索系统，为科研人员提供学术资源集成检索服务。

基于RSS(Really Simple Syndication)的Web新闻主题聚合是当前信息处理领域中一个新兴的方向，具有很好的实用价值。RSS是一种Web内容联合格式(Web Content Syndication Format)，RSS规范描述了XML风格元素的一个简单子集，是一套用于描述Web内容的元数据规范，这些元素可以用来为网站内容创建一个或者多个汇总，并提供了一种新颖的Web内容联合机制，包括内容的整合者、内容的提供者和最终用户三个组成部分，然后被内容提供者发布在网站或其它媒介中并进行推广，最终由内容整合者以门户网站的形式进行展现，或者是内容的直接使用者使用独立的桌面工具或订阅服务最终进行使用[8]。与通用搜索引擎相比，RSS更适合于对那些已知网站的信息进行整合，例如，新闻RSS引擎NewIsFree[9]就是利用了RSS技术，通过每天24小时不间断的跟踪数千个新闻网站，为互联网用户提供数百个新闻频道浏览、检索、通知等各项相关的服务。

包胜华[48]在其博士论文中基于Web对实体信息搜索与挖掘进行了重点研究，其侧重点在于实体的搜索与挖掘，并没有在检索系统方面进行深入的研究，其成果主要是实体挖掘。陈永超[49]等人提出现了一种基于命名实体的搜索结果聚类算法，主要工作是将实体作为聚类的标签，将搜索出来的文档结果进行聚类。Tao Cheng[51]等人主要的关注点是实体属性的抽取，最终给出了一个实体实体搜索引擎的整体框架。

1.4 本文的研究工作

本文在现有的文本信息融合的基础上，结合新闻IT领域和中文文本的特点，主要针对以下几个方面的技术展开了研究工作：

1、网页内容抽取。一般的网页都是由html标记语言来描述，包含的信息量会很大，同时无用的信息也相对较大，在这步中只提取出对识别有实质作用的标签的内容，比如新闻的标题、正文以及用户的评论等，而把一些噪音数据，例如广告等数据过滤掉。门户网站中大多数的网页是一些类似与新闻与博客的文章，这些信息则需要进行分析后融合到实体单位上的数据，也是索引与搜索的相关数据，需要被精确的抽取出来。

2、实体词典库构建。实体抽取或识别技术是本文的基础，本文利用开源的词法分析系统对中文新闻文本进行实体的抽取，交由后续的信息融合模块使用。当前开源的词法分析系统大多基于统计的方法，对训练语料集的敏感度很高，所以大多的基于统计的开源的词法分析系统都提供了添加用户词典的功能来弥补这一缺陷。本文基于百度百科，通过词条的抽取、分类、整理，得到一个基于行

业领域的实体词典，实现实体词库的生成，减少了人工的分类工作。

3、实体信息融合。首先，收集各大门户网站中的 IT 新闻文本以及 IT 行业中知名博客中的博文，通过网页抽取技术，整理并构建了面向行业的中文新闻领域的语料库。然后，通过机器学习的方法构建面向行业的网页信息融合原型系统，利用基于图的排序算法计算文本与实体的相关度，在语义理解的基础上得到文本中实体的权重，并根据实体在所出现的文本的权重计算出实体间的关联度。最后构架了一个提供用户搜索的原型系统，用户可以搜索实体的相关新闻与相关实体关联信息。

1.5 本文的组织结构

本文主要针对面向行业的中文新闻领域的文本，开展了信息融合的相关研究工作，并实现了一个面向行业的信息融合原型系统。本文基于实体的信息融合方面的研究，使用网页信息抽取技术将网页中的文本信息抽取出来，并基于百度百科词条构建实体库，利用基于图排序算法最终实现了面向行业的信息融合原型系统，另外本文还对系统进行了精度测试与性能测试的相关实验，并对实验结果的具体细节展开分析。

本论文总共分为七个章节，每一章节的内容如下：

第一章是绪论，本章内容主要是介绍发研究背景、研究意义以及国内外在信息融合技术的相关研究现状。

第二章是背景知识介绍，论述本研究所应用的相关技术概念和原理，概括了当前中文处理领域的相关理论与技术，包括网页抽取、文本的分类技术和基于图排序技术。其中详细介绍了文本分析技术的流程，以及几个常用的分类算法模型，包括 k-近邻算法和支持向量机算法等。

第三章是信息融合系统研究，首先介绍了网页数据精确抽取的基本原理；然后详细讨论了实体词典的构建方法，主要包括百度百科网页信息抽取、词条特征提取以及基于 SVM 模型的词条分类等内容。

第四章是系统设计，从一个信息融合原型系统出发，分别介绍了本文研究的相关概念、问题模型。然后介绍了面向行业的信息融合系统原型的流程及架构设计及系统相关模块。最后详述了系统架构中各个包和类的功能与情况等。

第五章是信息融合系统的详细系统实现，详述系统架构中各个模块功能的具体实现及主要算法，包括网页的聚类、模板生成、新闻语料库的准备、SVM 分类器的建立以及实体关联模型的建立等。

第六章是实验与结果分析，概述了基于行业的信息融合原型系统的实验流程，并对实验结果展开了分析，主要包括相关测试精度指标的介绍、信息融合系

统的实验流程介绍，信息融合系统实验结果的分析和影响系统精度的相关因素的讨论等。

第七章是结论与未来研究方向，根据现有的知识与了解，提出本系统在未来有待深入探讨的地方与后续可能的改进方向。

第二章 相关理论与技术综述

2.1信息检索的经典模型

信息检索，通常是一个按照用户所提供的相应的查询请求对已有的数据信息，尤其是对文本数据信息进行相关处理，从而找到用户查询的信息的过程。研究人员在对信息检索技术的研究之中，已经提出了许多信息检索模型。纵观目前国内外信息检索模型方面的工作，研究者已经提出了许多模型来估计文档之间的相似度[12]。在针对当前搜索引擎的众多研究中，研究者也提出了许多利用元数据来提高文本相似度排序性能的模型，比如文档标题[13]、锚文本[14, 15, 16]以及用户查询日志[17]等。

2.1.1 布尔模型

布尔模型[27]模型中，每个词在一篇文档中是否出现，对应权值为 0 或 1，一个查询词就是一个布尔表达式，包括关键词以及逻辑运算符。通过布尔表达式，可以表达用户希望文档所具有的特征。

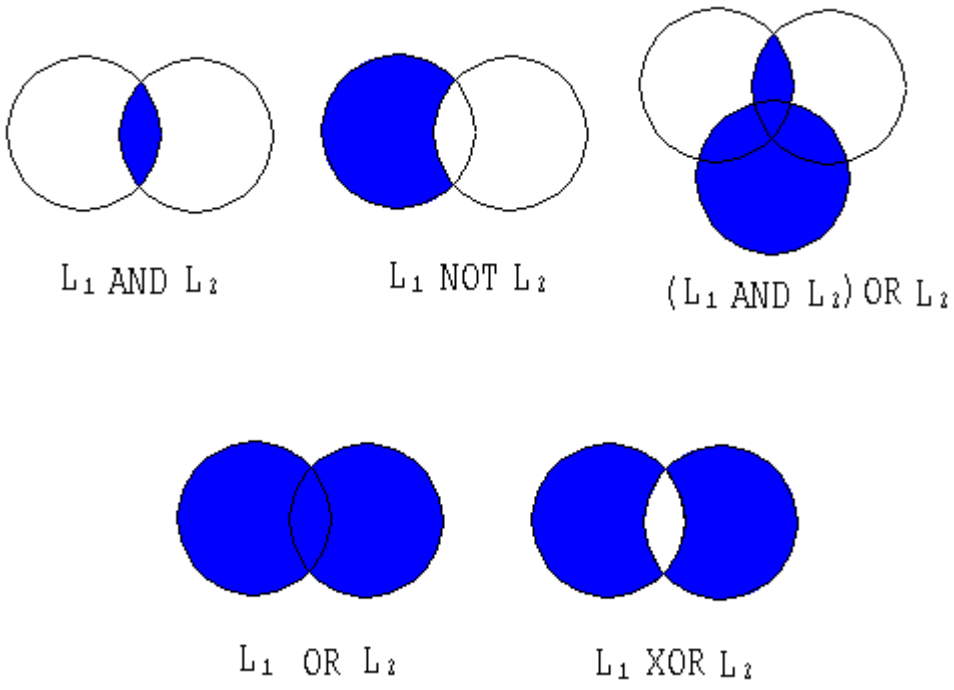

图2 - 1 布尔表达图解

布尔模型由于内部结构和形式十分简洁，如图 2-1 所示，布尔模型的就查找

使得查询词返回为“真”的文档，查询串以语义精确的布尔表达式的方式输入，使用与、或、非等简单的逻辑运算进行信息的检索操作，非常直观，易于用户使用与理解，因此在过去的几年引起了研究者的广泛关注。

2.1.2 向量空间模型

Salton 等人在 20 世纪 70 年代提出的向量空间模型[20]（Vector Space Model）模型。向量空间模型以空间上的相似度表达语义的相似度，成功地应用于著名的文本检索系统 SMART[28]。向量空间模型应用的潜在前提是文档长度和相关度无必然联系，假设事先没有对向量的相关系数进行归一化处理，显然长篇幅文档的相关度要比短的高，因为它包含更多单词。

在向量空间中，任一文档 d_j 都可用一个向量 $\overline{d_i}$ 来表示，称之为文档向量。该向量在各个轴上的分量就是相应的各个特征项在各个文档中的权重，一般采用 TFIDF 方法计算。文本的向量 $\overline{d_j}$ 可以表示为：

$$\overline{d_j} = \sum_{i=1}^{n} w_{ij} \vec{t_i} \qquad \text{式（1-1）}$$

其中 w_{ij} 为特征项 t_i 在文档 d_j 中的权重。在向量空间模型中，所有的文档都被表示成空间中的向量，相应的，文档间的相似度可以通过计算向量间的相似度来获取，即文档间的相似度[54]表示为两个向量间夹角的余弦值，其表达式表示为：

$$Sim(d_i, d_j) = \frac{\sum_{i=1}^{n} w_{i1} * w_{i2}}{\sqrt{\sum_{i=1}^{n} {w_{i1}}^2 \sum_{i=1}^{n} {w_{i2}}^2}} \qquad \text{式（1-2）}$$

为了提高效率，在大量计算两两文档间相似度时，为降低计算量，一般先对文档进行向量进行单位化。

2.2 网页数据抽取技术

国外的网页信息抽取技术研究较早，成果较多，Web-Harvest[41]是一个 Java 开源 Web 数据抽取工具，它能够根据用户所写的配置文件收集指定的 Web 页面并从这些页面中提取有用的数据。在使用 Web-Harvest 的时候，用户必须针对每一个网页都编写一个配置文件，并且当原网页结构发生变化后，需要用户重新修改配置文件。因此，Web-Harvest 不适合用于庞大网页集的信息抽取。

RoadRunner[42]是基于包装器自动生成的网页数据抽取工具。它提出了一种

新颖的、通过比较 html 网页标签和文本的匹配和不匹配来生成包装器，从而抽取数据的算法。算法采用递归回溯的方法对网页的标签和文本进行比较。当标签或文本匹配时，则认为其实网页结构类信息；当标签或文本不匹配时，则认为其实网页内容类信息，并将其抽取出来。这种算法的优点是智能化的生产包装器，然而缺点也很明显，它的抽取精度相对较低，且当待比较的网页结构不相似时，其算法的时间复杂度极高。

在国内，网页信息抽取的相关工作起步较晚，与国外的研究成果相比，中文网页信息抽取的技术还处于探索的阶段。孙承杰[44]提出了一种基于统计的正文信息抽取方法，但对带有表格信息的网页未作考虑。李剑波[45]等人提出一种半结构化信息获取方法，将语义加入抽取规则中，进而抽取数据。

2.3 文本的分类技术

随着互联网技术的发展，人们通过网络获得的信息资源越来越多，这些信息资源来自各种不同的网站中。在网络这个庞大的信息资源库中，由于文本数据不是结构化的，用户在面对如此杂乱无章的信息时，网络信息的利用效率大大的降低了。因此采用自动化程度更高效率更好的数据处理方法，帮助人们更高效地进行文本分类将是未来的发展趋势。文本的自动分类技术则能降低网络的查询时间，使快速有效地获取文本信息成为可能，提高网络搜索质量。本节主要介绍了几种基于文本的分类技术的原理和方法，为使用文本分类方法提供一定的参考。

2.3.1 K-最近邻分类算法

K-最近邻[29]方法是一种基于向量空间模型，通过比较训练元组和测试元组的相似度来学习的分类方法。首先，将训练元组和测试元组看作是 n 维（若元组有 n 的属性）空间内的点，然后给定一条测试元组，搜索 n 维空间，找出与测试元组最相近的 k 个点（即训练元组），最后对类别的分值进行统计和排序，并把待分文档划分到那个得分最高的类别中去。KNN 中使用空间距离的方法度量两点间的关系，距离越大，表示两个点越不相似。距离的选择可采用欧几里得距离、曼哈顿距离或其它距离度量。但是因为欧几里得距离比较简单，易于实现，研究者大多会选择采用欧几里得距离。

KNN 算法的具体决策过程如图 2-3 所示：

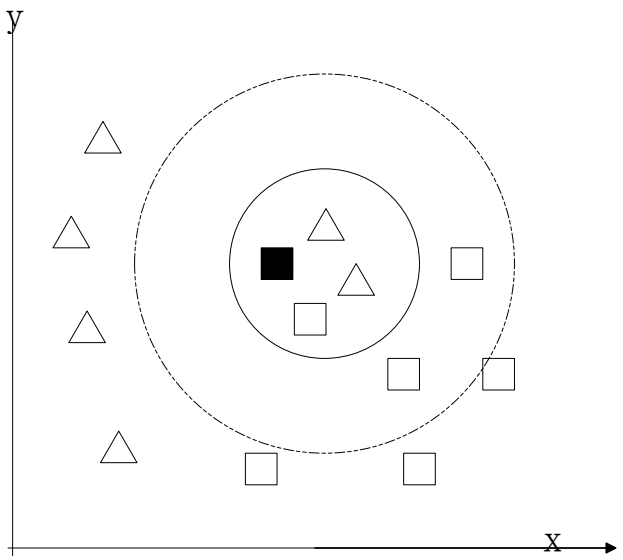

图 2 - 3 KNN 算法的决策过程

如图 2－3 所示，实心正方形代表一个待分类样本，要判断其所在类，首先要计算该待分类样本与已有样本的距离，找出其中与实心正方形距离最近的 K 个已知样本。如果 K=3，可得到在与实心正方形最临近的三个已分类样本中，实心正方形将被赋予与三角形所在类；如果 K=5，同理实心正方形则被判分到空心四方形所在的类。

从上述可以看出现，KNN 算法存在着不足，距离函数的确定和 K 值的确定是一个难点。最常用的距离函数有欧几里得距离、绝对距离、标准差和平方差等。而 K 值的设定一般是先确定一个初始值，再根据分类的结果进行试验与调整。

2.3.2 支持向量机分类算法

支持向量机[30]是 90 年代中期基于统计学基础上发展而来一种机器学习方法。因为引入了核函数，所以 SVM 基本不关乎维数的多少，即和样本的维数无关，在解决高维数据分类时，同样也可以达到很好的分类性能。其在解决非线性分类中表现出现很好的性能，在函数拟合等其他机器学习领域中得到了研究者们的广泛推广与应用。

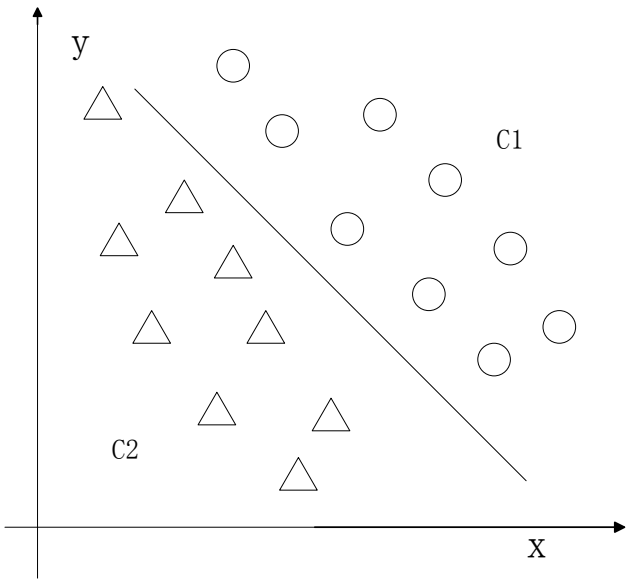

图 2 - 4 二维平面里二元分类问题举例

如上图 2-4 中有两个不同的类别，三角形 C2 和圆形 C1，它们的样本在二维平面中的分布如上图所示。分类的做法就是找到一个分类函数，将两者分开。如图中间这条直线可以将上述两类样本完全的分开，这条直线就就是这个二维平面中的一个分类函数。同理，在三维的空间里面这个分类函数就是一个面，四维、五维甚至更多维度依次类推。这种线性函数被统一称为超平面（Hyper Plane）。通常使用最大间隔法和最近点平分的方法来构造这个最优决策平面，得到一个超平面，称为“线性可分的支持向量机”。在支持向量机 SVM 模型中，通过使用核函数将特征参数投影到更高维度的空间里去。

在支持向量机中使用的核函数主要有四类：

线性核函数：

$$K(X_i, X_j) = X_i^T X_j \quad \text{式（1-3）}$$

多项式核函数：

$$K(X_i, X_j) = \gamma(X_i^T X_j + r)^d, \gamma > \quad \text{式（1-4）}$$

RBF 核函数：

$$K(X_i, X_j) = \exp(-\gamma \| X_i - X_j^2 \|) \quad \text{式（1-5）}$$

Sigmoid 核函数：

$$K(X_i, X_j) = \tanh(\gamma X_i^T X_j) \quad \text{式（1-6）}$$

其中，γ, r 和 d 均为核参数。

建议一般都是使用 RBF 核函数，虽然在实际应用中使用哪一种核函数取决对数据处理的要求，但是 RBF 核函数在实际问题中表现出了良好的性能，受到

研究者的青睐。

2.4 基于图排序算法的关键词抽取

关键词[1]提供了文档内容的概要信息，可以在整体上给信息使用者说明文档的概要信息。随着社会信息量的不断爆炸式的增长，起初人为的给出文档关键词的原始作法已经不足以满足人们的需要，并且变得不现实。于是在这样的背景下，关键词提取技术渐渐的成为了研究的热点，目前大部分的关键词提取算法都是基于机器学习的方法[1, 2]。

Salton 使用全局信息（图的结构）对节点进行排序的方法，研究文本的切分对于检索效果的影响，由此提出了文本关系图[117]的概念，它是一种描述文本之间关系的形式化模型。基于图的排序算法是决定图中点重要性的一种方法。Mihalcea 提出了基于图的排序的 TextRank[3]算法模型，被广泛使用在文本关键词抽取算法中。TextRank 算法模型把文本中的切分后的词通过类似基于图的排序算法，来确定各个词的重要性。

本文我们提出一种基于 TextRank 算法的信息融合的方法，根据文本中实体对文本的贡献度来判定实体与文本的相关度。

2.5 本章小结

本章主要介绍了与信息融合技术的相关技术与理论，其中包括了信息检索模型、文本分类技术、网页抽取技术等相关技术。其中，主要介绍了文本分类技术的操作流程，包括训练集的获取、特征模型的建立、特征选取、分类器的生成以及文本分类性能评估这五个步骤；各算法模型包括 KNN 和支持向量机（SVM）。最后介绍了基于文本语义的排序算法 TextRank 算法。上述这些理论内容与技术都是本文对面向行业信息融合原型系统展开研究与实现的理论基础与技术支撑。

第三章 面向行业的信息融合原型系统的研究

通用搜索引擎通过利用信息模型，能在很大程度上帮助用户过滤一些无用的网络信息并找到用户真正所需要的信息。但是，因为搜索引擎搜索粒度较大，所以返回的查询结果均比较单调，即这些搜索引擎通常返回网页供用户浏览，而忽略了网页中所蕴涵的丰富实体信息。许多的研究者也意识到了上述的问题，开始对网页的内容进行进一步的分析，并挖掘其中的实体信息以及实体与实体之间的联系。

本文也从实体的角度出发，开展了专注于面向领域的信息融合相关工作，进行以实体关联模型为基础的数据集成的研究，目的在于设计并实现一个运行良好并且结果较为理想的能够融合某个领域中新闻内容的系统，将从各个不同网站获取的网页数据信息，以实体为中心的实现方式融合起来，将网络数据以可视化的方式进行展示，便于用户的搜索。

3.1 系统研究步骤

实现基于实体关联模型的面向行业的信息融合系统，必须借助网页信息抽取、实体识别、文本分类等多个领域的技术。本文是在向量空间模型的基础上，对向量空间模型进行改进，以篇章中出现的实体做为篇章的一个维度，提出实体关联模型的方法来衡量文本之间的相关度。基于实体关联模型的信息融合系统的系统研究步骤主要如图3－1所示：

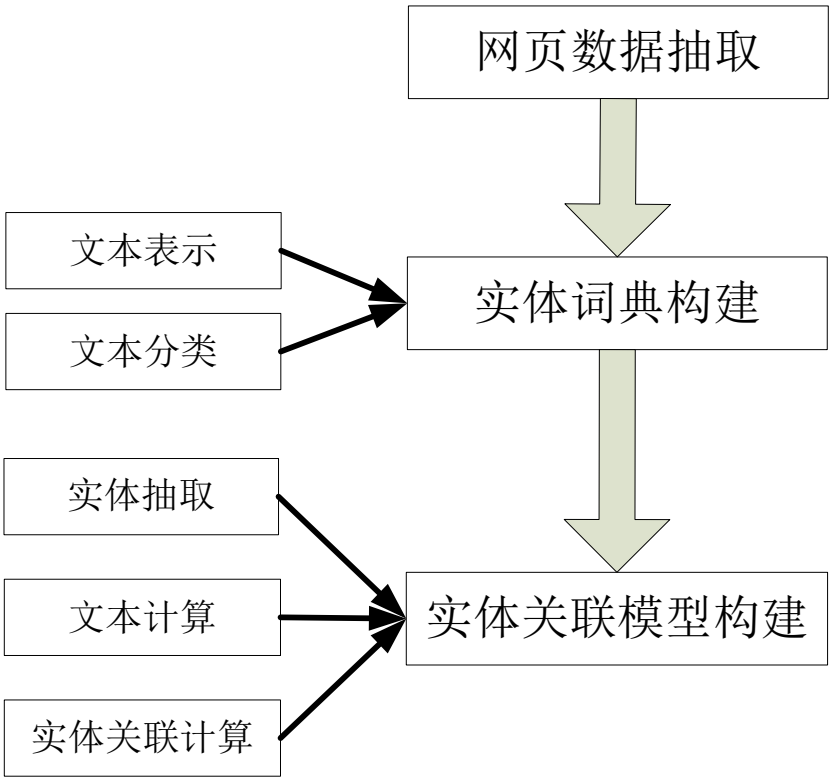

图3 - 1 系统研究步骤图

如图3-1所示，整个信息融合系统分为三个大的部分，分别为网页数据抽取、实体词典的构建与实体关联模型构建两部分，其中网页数据抽取基于简单树匹配与树对齐算法开展研究，实体词典的构建是基于百度百科的词条分类来完成，实体关联模型构建是通过TextRank算法与实体间的距离的计算来完成，各部分的详细研究内容将在本章接下来的部分展开。

3.2 网页数据精确抽取

刘斌[46]等人提出了一种简单匹配树算法，其中HTML标签树不包含节点之间的替换和层次之间的交叉操作。简单匹配算法的目标是找到两棵树间的最大匹配，即在A和B两棵树中查找一个最大匹配M，即一个拥有最多节点对的匹配，使得M中的节点a和b，有 $a \in A \wedge b \in B$，并且需要保证a的父节点pa与b的父节点pb满足 $pa \in A \wedge pb \in B$，这样的M就是一个拥有最多节点对的匹配，如图3-2所示：

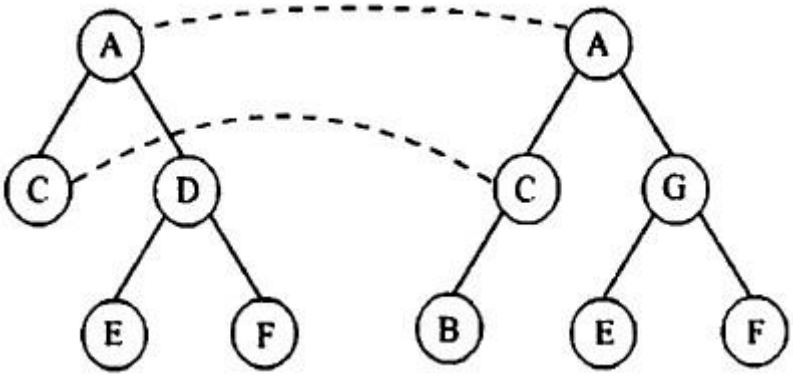

图3-2 树的最大节点匹配映射

STM算法的复杂度为O(n1*n2)，这里n1，n2分别是树A和B两棵树的大小。函数match(A，B)返回A，B节点的匹配权重。

```
1   STM(A,B)
2   begin
3      if A != B then
4         return 0
5      endif
6      let m be the number of children of node A
7      let n be the number of children of node B
8      M[i,0] = 0 for i=0,…,m
9      M[0,j] = 0 for j=0,…,n
10     for i = 1 to m
11        for j = 1 to n
12           W = STM(Ai,Bj)
13           M[i,j] = max(M[i,j-1], M[i-1,j], M[i-1,j-1]+W)
14        end
15     end
16     return M[i,j]+match(A,B)
17  end
```

图 3-3 简单树匹配算法伪码

在简单树匹配算法中，当两棵树中出现多个可匹配的节点时，算法会返回第一个匹配的节点，如图3-4所示，A树和B树中的E节点是相互匹配的，算法取出的是B树中的第一个E节点。

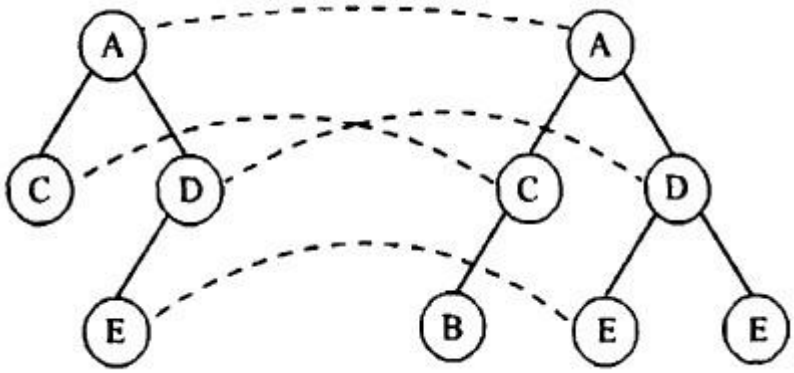

图 3-4 多种可能的最大匹配情况

使用简单树匹配算法计算后，可以得到两棵树的最大匹配节点，就可以全用式3-1所示公式求得两棵树的相似度：

$$Sim(A,B)=\frac{match(A,B)*2}{node(A)+node(B)} \qquad \text{式（3-1）}$$

本文中，网页数据精确抽取算法是原型系统的基础，在实体词典构建、新闻网页数据的抽取都使用了该算法进行数据的抽取，该算法的精确度直接影响了后续系统的性能。网页数据精确抽取算法的主要研究内容包括以下几个方面：

（1）通过对大量的网页集进行训练，生成网页模板，本工具可以同时处理大量不同结构的网页集，对于每一个类别（根据网页相似度进行聚类）的网页集会生成一个相应的网页模板。

（2）设计用户界面，用户通过模版选取感兴趣的信息，对其进行标记（语义信息）。在前面介绍诸多研究中，标记语义信息大多较为复杂，需要用户对网页结构进行分析，这就大大降低了软件的易用性，因此需要设计一种适用于更多非计算机专业的人群的交互界面，让用户可以通过简单的操作对模板上感兴趣的信息进行标记。

（3）利用用户标记过的模板，对指定的网页抽取出用户感兴趣的数据。对于待抽取的网页，根据该网页和模板库中的模板的相似度，选择对应（相似度最高）的模板，抽取出用户感兴趣的数据。

（4）抽取的信息可以按用户需要放到指定的文件（XML，HTML）或者用户数据库中。本工具抽取的数据可以作为数据挖掘、数据分析等其他系统的数据基础，因此需要为用户提供较为丰富的数据保存形式。

3.3 基于百度百科的实体词典

实体是句子中关键信息的承受者，句子所要表述的信息是围绕着实体展开的，最终都是为实体服务，例如，“谷歌公司专注于未来项目的开发”的关键信息是“谷歌”这个实体。

本文中，我们选用的是国科学研究院开发的ICTCLA开源分词系统，可以将句子中的字词的切分并标注出该词的词性，而且其性能表现基本上能够满足本文的实体抽取的需要，有一点美中不足的是该系统对机构名的识别不是太理想，远没有其人名识别的性能好，好在ICTCLA开源分词系统还提供了一个扩充用户词典的功能，即用户可以根据的需要增加词典来满足对系统分词性能进行优化。

Wiki是一种通过集体协作进而创造知识的平台，现在最大最全的是维基百科，维基百科（wikipedia）[18]是开放式的，并且是面向互联网的百科全书，这使其与与传统意义上的百科全书最大的不同之处。维基百科全书使用重定向页来标识两个不同的标识符所表述的意思是相同的，例如，“Google”和“谷歌”为两个不同的标识符，但是是同一个概念的不同表述而已，所以经过重定向后，会连接到相同的一个页面。在维基百科中，每篇文章介绍一个概念，也就是一个实体，这些实体是众多的维基编写者的集体智慧的结晶，也是当前社会所共知与认同的内容。而在中文领域中，由百度公司推出的一部内容开放、自由的网络百科全书百度百科[19]也在非常迅速的成长，其收录的内容包括知名人物、抽象概念、文学著作、汉语字词或特定主题的组合，例如：“云计算”、“李开复”、“谷歌公司”等。词条是百度百科所含内容的基础单位，例如：“刘德华”“2008年北京奥运会”。与维基百科相比百度百科更加贴近中文使用习惯，更适合于本文系统中用于词典补充的信息来源。

3.3.1 基于机器学习的百度百科词条分类

在本文的实体模型中，实体的识别需要加入一定的词典，用于提高实体抽取的精确度。在上两个小节中，我们简单的介绍了维基百科与百度百科的概念与其重大的利用价值，在这一小节中，我们将利用机器学习中分类的方法，对百度百科中提取出现的词条进行分类，即将其中的人名、公司机构名字、专有名词等进行分类，并将分类的结果以ICTCLA开源分词系统中用户词典的格式进行格式化，最终利用该词典进行实体的抽取。

本文中，我们需要以百度百科为基础，构建一个可以用于促进实体抽取系统精确度的实体词典，但是百度百科的词条数目非常巨大，到现在为止，已经有2869910个用户为百度百科贡献了5629957个词条，因此，必须借助计算机的力量

来分离出现对我们有用的词条，光靠人工的力量显然是力不从心。

在本文中，我们研究了百度百科的特点，选择一些更多包含有词条明显分类性质的文本内容，在很大程度上提高了分类的精确度。

图3 - 5 百度百科词条抽取内容

如图3-5所示，利用本文中的网页信息抽取方法，可以将页面中用红色方框标注出的内容，其中包括文本中的超链接、名片的列表等，并将这些内容做为分类的训练文本进行训练。

实验证明，以这个方法进行的词条分类可以达到很高的精确度，很好的满足我们分类的需求。许多研究者在使用机器学习的方法对文本进行分类时，通常主要选用朴素贝叶斯算法、决策树算法和支持向量机等机器学习分类算法。基于机器学习方法的百度百科词条分类过程如图3－6所示：

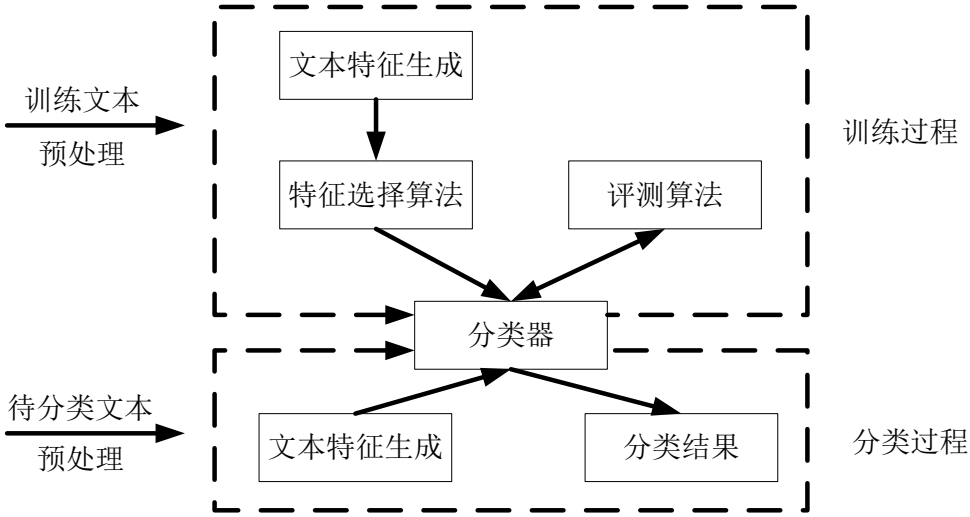

图3 - 6 机器学习方法对文本进行分类

因为其良好的分类效果，支持向量机在许多领域中得到了成功的应用，所以本文选用支持向量机分类方法对文本进行分类。

3.3.2 文本特征选择

将文本进行特征向量表示时，文本中出现的所有词条将就组成了原始的特征集。假如文本集很大的话，原始特征集会的维度会很高。在处理高维特征空间时，为了提高程序的执行效率和运行速度，需要进行特征降维。同时，在高维特征空间中，存在着一些在各个分类中都存在的特征，对分类的贡献很小，降维可以提高分类器的推广能力。

特征选择方法主要包括文档频率DF、互信息MI、信息增益IG和CHI统计等。本文在语料库和分类算法确定的情况，基于实验对以上四种特征抽取方法进行了分析和比较，得到的结果是文档频率法效果最佳，而且实现比较简单，所以本文中选用文档频率(DF)作为特征选择方法。

3.3.3 文本的表示

文本分类中使用较多的权重计算方法是 TF/IDF 模型。现在的搜索引擎对 TF/IDF 进行了不少细微的优化，使得相关性的度量更加准确。TF-IDF 权重计算方法十分简单，并且在实际应用中表现出现的非常良好的性能，这使用得 TF-IDF 权重成为在文本处理领域方面使用最为广泛的数值权重计算方法之一，深受广大研究者青睐。TF-IDF 模型的主要思想是：如果特征 i 在一篇文档 d 中出现的频率很高，但是在文档集 j 中却很少出现，则表明特征 i 具有很好的区分能力，适

合用来把文档 d 和文档集 j 中的其他文档区分开来。

目前最为常用的计算方法余弦相似度(cosine similarity)，其定义如下所示：

$$a_{ij} = f_{ij} \times \log(\frac{N}{n_i}) \quad \text{式（3-2）}$$

其中，f_{ij} 表示特征i在文档集j中出现的频率，N为文档集合中文档的总数，n_i 为特征i在文档集合j中所出现的总次数。在向量空间模型中，TF/IDF权重计算方法经常与余弦相似度一同使用，用来判断两个文档之间的相似性。

3.3.4 基于百度百科的分类建模

机器学习，顾名思义，需要经过对训练样本进行学习后对样本进行分类，属于有监督学习。机器学习中产生的各个分类算法模型各有优劣，总体来说，SVM 模型是性能最好的。本文通过基于统计的算法模型 SVM 算法，首先对经过预处理后有词条文本训练集进行训练，生成特征项，然后将待分类词条放入训练后的模型中进行分类，得到最终的词条分类结果。

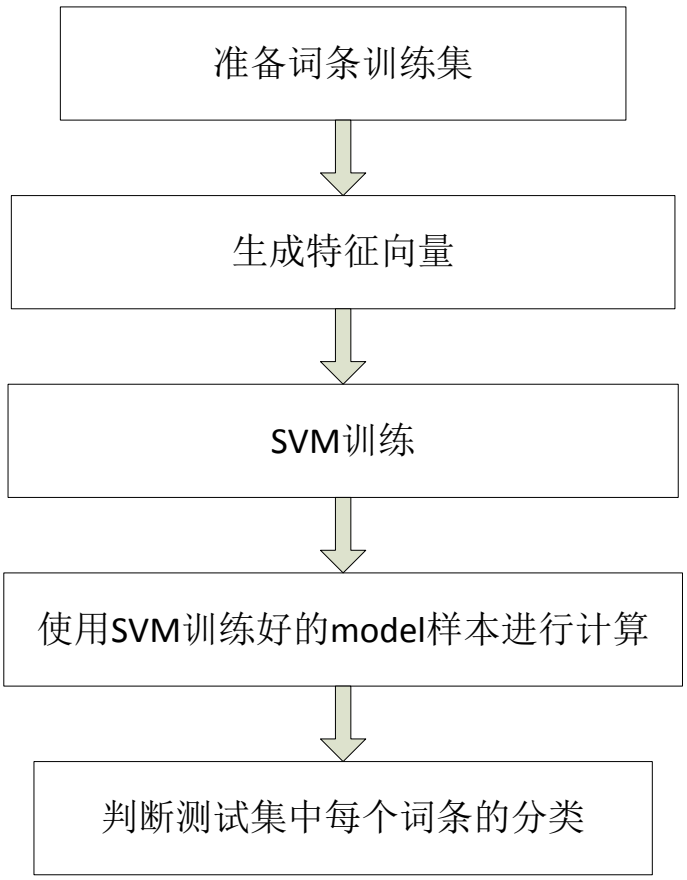

图 3 - 7 使用SVM算法对词条进行分类

整个 SVM 模型的百度词条分类流程如图 3-7 所示，其流程如下：

（1）词条训练集的准备。本文的 SVM 模型主要应用在基于百度百科的词条分类中，领域主要是百度百科中的 IT 行业中的知名人士、公司企业以及专有名词的词条文本，选择其的训练文本集。

（2）特征选择。对网页精确抽取得到的词条训练集进行文本预处理，然后通过人工的判断的方法，通过人工的判断的方法，将词条训练集按类别分成人名、公司机构名、专有名词以及其它四个大类。对每个词条文本使用向量空间模型进行表示，使用文档频度（DF）方法进行特征选择，归纳出每篇文章的特征向量。

（3）使用 LIBSVM 进行文本训练。在已有的特征向量的基础上，分别用 LIBSVM 训练，具体的训练算法实现将在实现部分给出。最终可以得到一个训练好的模型文件。

（4）使用 SVM 模型对待分类词条进行分类。批量循环的输入测试的百度百科词条文本数据，与训练文本一样也需要对先对文本进行预处理，然后对测试集进行特征提取，获得测试集中的特征信息，最后进行预测得出分类结果。

实验结果证明，基于 SVM 的机器学习模型精度一般在 70%到 85%左右，本文中用于进行词条分类的机器学习模型比这个结果稍微高出现了 4%到 10%左右，很大的原因取决于训练集质量的高低，以及特征向量维数的选取等因素。

3.4 实体关联模型

本小结内容主要基于实体关联模型的信息融合的相关研究，在紧接着的小结中将介绍实体关联模型的表示、实体与文本的相关度以及实体与实体之间的关联度的相关研究。这部分的研究内容都将作为系统的研究子模块分别在后期实现，这一小节，我们将详细的阐述基于实体的信息聚合的模型。

信息融合技术研究的是如何加工、综合来自于多个信息源的不同信息，并能将这些不同形式的信息进行相互补充、整合，使其信息量达到最大限度的发挥。信息融合技术通过对信息的取舍和集合划分后应用于检索系统，可以更加合理的将查询结果组织起来，方便于用户查询使用。实体关联模型的构建流程如下图 3-8 所示：

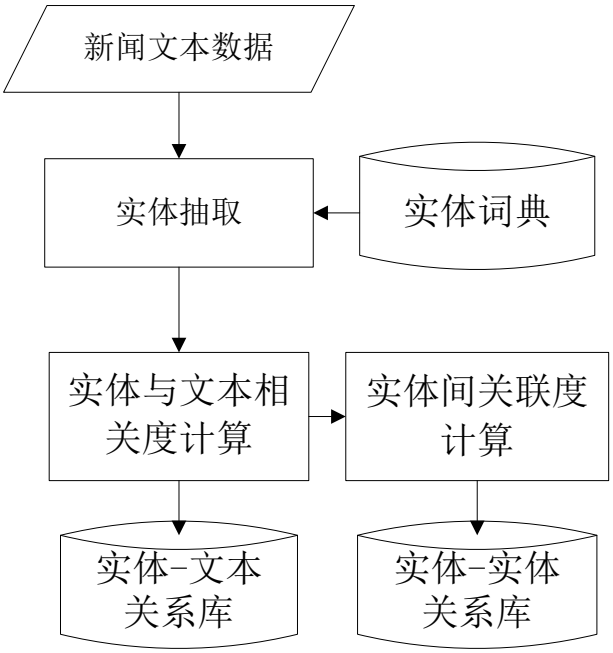

图 3 - 8 实体关联模型示意图

其中所用到的技术包括有网页信息抽取技术、实体抽取技术、基于图排序的文本关键词算法等，在这一小节中将一一进行阐述。

3.4.1 实体识别与抽取

在中文自然语言处理领域，中文信息处理的基础与关键是词法分析，但是因为在中文的句子中，没有英语句子那样明确的对字词进行划分，所以要进行后续的处理第一步必须将中文的句子进行字词的划分。

目前已有多种可用的开源分词工具，如中科院的开源分词系统ICTCLAS、MMSEG4J、SCWS分词系统、盘古分词、庖丁分词等。分词工具的性能好坏对情感分析的处理时间长短和结果准确率的高低都有显著的影响。

使用分词工具对句子进行切分处理后，可以得到每个切分后的词的词性，从而抽取出现句子中的实体。我们在比较多个分词系统后，确定使用由中科院开发的ICTCLA开源分词系统[23]，该系统基于隐马尔可夫模型和N-最短路径的策略，在理论上具有一定的优势，经过一系列不同的测试，它有一个很好的结果，尤其对中文人名识别方面精确度可以达到98%以上，而且ICTCLA有不同的平台版本容易扩展，同时还可以使用用户自定义词典的功能，扩展起来十分方便。在实体识别方面，本文在ICTCLA分词系统的基础上，我们使用添加词典的方法，用于弥补ICTCLA分词系统对机构名称识别精度相对较低的缺点，实体词典的构建方法将会要下一小节中具体描述。

3.4.2 实体关联模型表示

目前文本表示通常采用向量空间模型[20]。本文基于空间向量模型，引入实体的概念模型。一个文档可以看成是 n 维空间中的一个向量。实体关联模型 M 主要由四部分组成，即

$$M=(D,E,R,F) \quad \text{式（3-7）}$$

其中，E 为所要融合的对象实体集合，这个集合就是模型的实体集合，在本文中需要通过实体抽取算法抽取出来，可表示为：

$$E=\{e_0,e_1,e_2,\cdots,e_n\}; \quad \text{式（3-8）}$$

D 为所要进行信息融合的文本数据集，在本文中，文本数据通过网页抽取算法，将新闻、博客等文本抽取的文本抽取组成，即：

$$D=\{d_0,d_1,d_2,\cdots,d_n\}; \quad \text{式（3-9）}$$

R 是 D 中的值与 E 中一个或多个 e 的相关度集合，

$$R=\{r_0,r_1,r_2,\cdots,r_n\}; \quad \text{式（3-10）}$$

映射函数 F，将模型中集合 D 的每一个值中的映射到 E 中，并计算 d_i 与 e_j 的相关值 r_{ij}。

如式（3-7）中给出的表达式，我们在进行信息的融合过程中，需要映射函数 F，将文本 d_i 与文本 d_i 中的实体 $\{e_{i0},e_{i1},e_{i2},\cdots,e_{in}\}$ 的相关度计算出来，这一过程可表示为：

$$R=F(D,E); \quad \text{式（3-11）}$$

若 R 为一个二元式，即

$$R(p)=\{(e_0,r_0),(e_1,r_1),(e_2,r_2),\cdots,(e_n,r_n)\}; \quad \text{式（3-12）}$$

于是，通过（1）（2）（3）式，我们可以定义搜索函数 S：

$$D_s=S(e); \quad \text{式（3-13）}$$

其中 e 是用户输入的实体，D_s 是经过函数计算后得到的文本集合，并且可以通过 D_s 中元素与 e 的相关度进行排列。

3.4.3 基于TextRank的文本与实体相关度

通常在一个文档中，大部分的词语都是围绕同一个主题展开阐述的。在这些词语构成一个语义集中 W 中，在语义上与 w_i 相关的词语越多，w_i 越有可能表达文档的主题，即与文档的关联度越大，同时与 w_i 在语义上有关联的词语 w_j 与文档的关联性越大，说明 w_i 与文档的关联性也越大。基于图的排序算法的基本方法是通过“引用”与“被引用”的关系，使用文本的全局信息，即图的结构，来对节点排序，从而决定图中每一个点在文档中的重要性。

我们可以将 w_i 看成是一个文档中出现的实体，基于上述这两个性质，本文引入基于 TextRank 算法，用于计算文档中各个实体与文档的关联度。

TexRank 模型是基于图排序的，可以用一个带权有向图 G（V,E）来表示，由点集合 V 和边集合 E 组成，E 是 V*V 的子集，图中两点 i，j 之间边的权重为 Wji。对于一个给定的点 vi，In(Vi)为指向该点的点集合，Out(Vi)为点 vi 指向的点集合点 vi 的分数定义为：

$$WS(V_i)=(1-d)+d*\sum_{V_j\in In(V_i)}\frac{w_{ji}}{\sum_{V_k\in Out(V_i)}w_{jk}}WS(V_j) \qquad \text{式（3-14）}$$

其中，d 为阻尼因数，取值范围为 0 到 1，代表从图中某一特定点指向其他任意点的概率。该算法被广泛应用于关键词的抽取应用中，本文我们提出一种基于 TextRank 算法的信息融合的方法，根据文本中实体对文本的贡献度来判定实体对文本的相关度。

3.4.4实体间关联度计算

一个实体 E 可以看成是 n 维空间中的一个向量，在两个实体 E_1 和 E_2 之间的关联度 $R(E_1,E_2)$ 指两个实体的相关程度，能够反映出实体间的语义相似性[56]。实体关联程度是由文本关联度决定的，因此，需要根据文本关联度集合中计算出现两个实体间的关联度期望。

设文档 D 的数量为 M，实体 E 的数量为 N。采用向量空间模型对类别进行表示，则任意一个文档 d_i 的词向量为：

$$d_i=\{(e_1,weight_1),(e_2,weight_2),\cdots,(e_N,weight_N)\} \qquad \text{式（3-15）}$$

对上述结果进行倒排索引，则词典中任意一个实体 e_i 的文档向量为：

$$e_i=\{(d_1,weight_1),(d_2,weight_2),\cdots,(d_M,weight_M)\} \qquad \text{式（3-16）}$$

对于任意 2 个实体 e_1 和 e_2，根据两者的文档向量采用向量夹角余弦公式计算实体间的相关度，具体计算如下：

$$Relevancy(e_1,e_2)=\frac{d_1\times d_2}{|d_1|\times|d_2|} \qquad \text{式（3-17）}$$

显然，我们在上一步可以得到每一个实体 e 的文档向量表示，利用式（3-18）可以得到每一对实体之间的关联度 $Relevancy(e_1,e_2)$。

3.5 本章小结

本章从研究的角度，阐述了本文中提出的基于实体关联模型的信息融合系统

原型的关键技术，主要包括网页数据精确抽取技术的研究、实体关联模型建立中词典分类算法、文本与实体的相关度算法以及实体间关联度算法，这些技术是本文信息融合系统的的基础，是本文所提出的信息融合系统的精华所在。

第四章 面向行业的信息融合原型系统的设计

上一章分别从原型系统的功能的角度出发，对系统所运用的一些关键技术进行了分析，本章将在上一章的基础上对系统进行设计。设计阶段将着重解决实现程序模块设计问题，即如何把开发的系统划分成若干个模块；决定各模块的接口，即模块之间的相互关系；以及确定模块之间的传递关系。

4.1 原型系统总体设计

信息融合是将互联网上来自于多个信息源的信息通过加工、综合，并将这些不同形式的信息进行相互补充、整合，使其信息量达到最大限度的发挥。如果在全行业上做一个又大又全的系统，那是一个十分巨大的工程，所以本文选择了 IT 行业——这样一个相对较小的领域进行原型系统的研究工作。

4.2 原型系统流程

面向行业的信息融合系统是基于实体关联模型建立的，信息融合的过程大致是将文档进行分词处理，对识别出来的实体进行相关度分析，再对实体的关联度进行加权求值的过程，将文档进行一个从面到点的处理，即文档到实体的表示，再由点到面的处理，即再从实体到文档的筛选过程。

本文所设计的面向行业的信息融合原型系统考虑到已有的以文档级别为基础的信息检索模型，大多数只是根据关键词加权处理的简单方式进行分析处理，而未考虑到现实世界中信息组织是以实体为中心开展的事实，结合使用TextRank算法与实体抽取方法，重点考虑了以实体为融合中心的相关处理算法并予以实现，再与实体间的关联算法模块进行集成，构成了面向行业的信息融合原型系统，具体处理流程如图4－1所示：

本文所设计的系统总体设计图如图 4－1 所示：

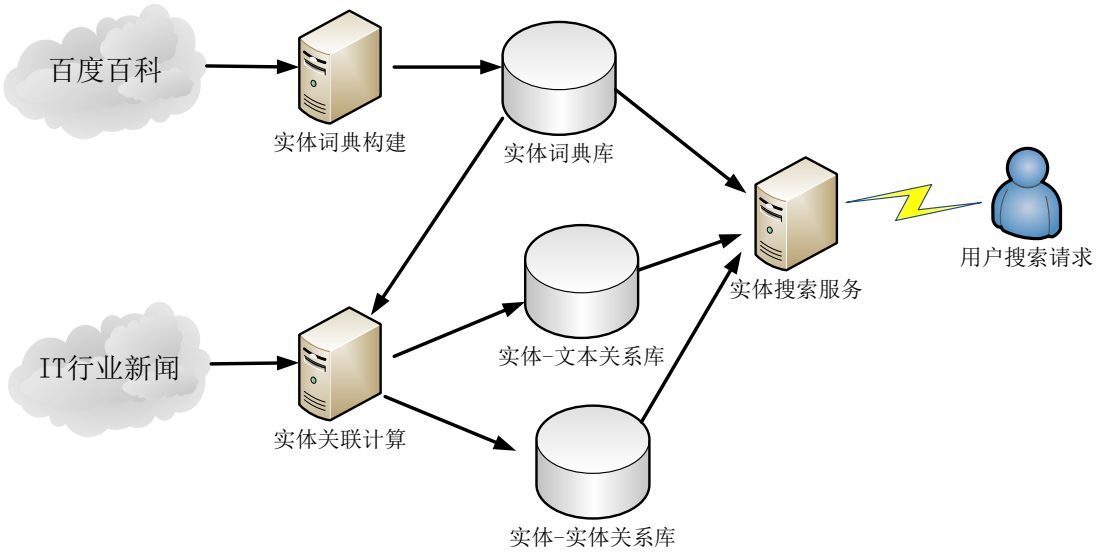

图 4 - 1 面向行业的信息融合原型系统总体示意图

如图4-1所示，信息融合系统原型可以分为三个大的部分，网页数据的抽取、实体词典的构建以及实体关联模型的构建，其中网页抽取是其它两个部分的公共部分，是其它两个部分的基础，为其它部分提供文档数据。

4.2.1 实体词典构建流程

实体词典的构建是本文中一个十分重要的部分，实体词库为实体抽取提供相应的用户词典，在很大程度上提高了实体抽取的精度，为本文最核心的实体关联模型算法提供足够的实体数据支持。

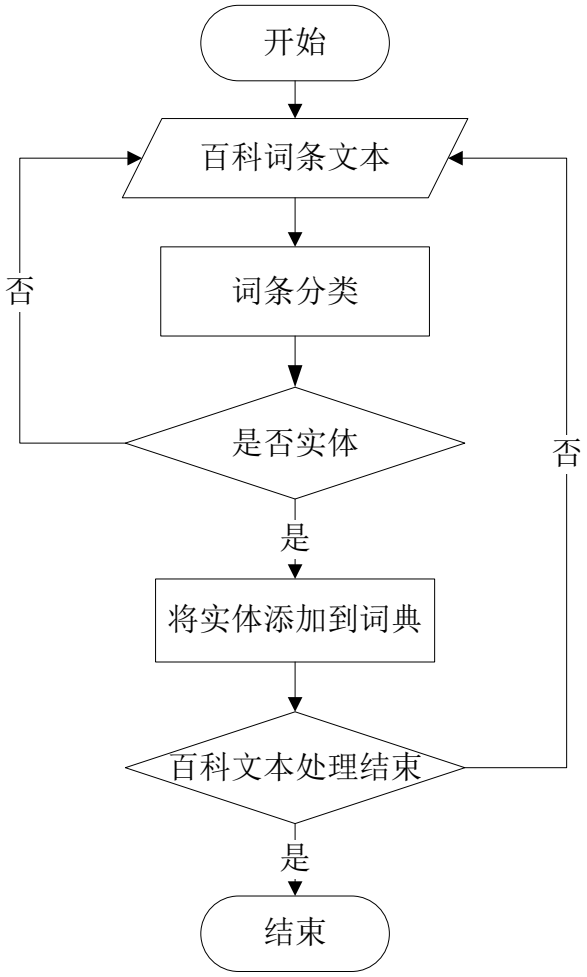

图4 - 2 实体词典构建流程

如上图 4—2 所示，该图实体词典流程图所示，首先，将通过数据抽取后得到的百度百科的词条内容，利用支持向量机为分类算法对其进行分类处理，如果词条被分到相关的实体分类中，则将其添加到实体词典中，以备实体抽取时使用，直到将百度百科的词条分类完毕。

4.2.2实体关联模型构建流程

实体关联模型的构建，是本文中十分重要的部分，是本文的核心。本文通过实体抽取的技术，将抽取出来的实体经过算法的计算构建一个实体关联模型。其构建流程如图4-3所示：

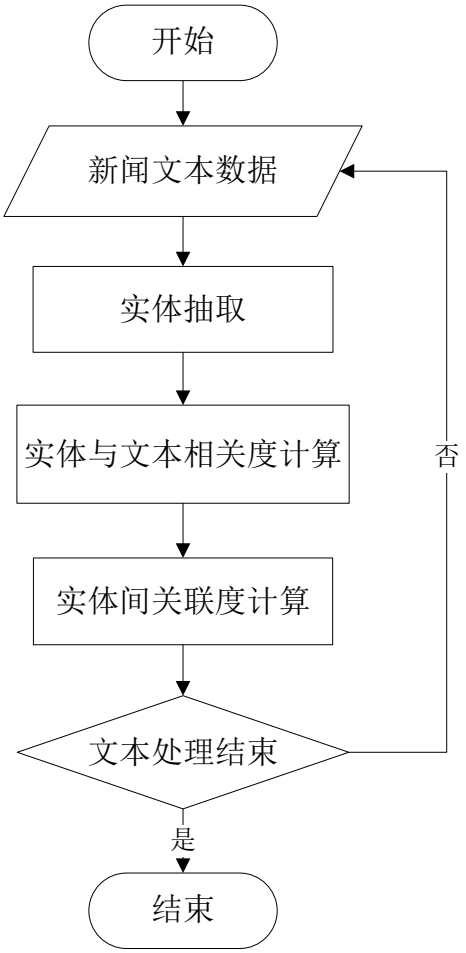

图4 - 3 实体关联模型构建流程

如图 4-3 所示的实体构建流程图，系统对进行新闻文本处理。其中包括实体抽取、实体与文本相关度计算以及实体间关联度计算。文档进行实体抽取之后，将文档转化为文档中实体的贡献度计算，即利用空间中实体的相互间的引用关系计算出实体的贡献度值，最后通过所有出现过某个实体的文档中的贡献度，计算出实体间的关联度。

最后，结合各模块分析所得实体与文本关系库和实体与实体关系库，最终将实体进行索引后，便可以向外部的搜索引擎提供服务。

4.3 原型系统架构

信息融合技术[4]始于 70 年代初，80 年代以后得到迅速发展，是一种综合利用多种信息资源，以获得对某一事物更客观、更本质认识的信息处理技术。

信息融合技术研究的是如何加工、综合来自于多个信息源的不同信息，并能将这些不同形式的信息进行相互补充、整合，使其信息量达到最大限度的发挥。信息融合技术通过对信息的取舍和集合划分后应用于检索系统，可以更加合理的将查询结果组织起来，使来源于不同信息源的信息可以连接为一个有机的整体，

这样可以方便用户查询到更为完整、准确、及时有效而且简洁、明了的信息。

本文所设计的系统架构图如图 4－4 所示：

图4 - 4 面向行业的信息融合原型系统架构图

最下面一层是数据层，包括前面章节中介绍过的新闻语料库、实体词典库、用于存储实体与文档之间关系的实体与文本关系库以及用于存储实体与实体之间关系的实体与实体关系库。

数据层的上一层是处理层，这是本系统最为核心的一块内容，主要包括网页抽取模块、实体词典构建模块以及实体关联模型构建模块。网页抽取模块主要进行网页数据的精确抽取，并对抽取出来的文本进行数据的清理，为其它两个模块提供基础的文档数据；词典构建模块主要是利用机器学习的方法对从百度百科抽取出来的词条进行实体类别的分类，为实体关联模块提供用户词典的补充，提高实体抽取的精度；实体关联模型构建模块是本系统的核心，模块首先通过 TextRank 算法计算出文档与实体的相关度；然后，在空间中将实体表示成文档的向量，通过计算两实体间的空间向量夹角将实体间的关联度计算出来；最后，分别将计算出来的结果存储到实体与文本关系库和实体与实体关系库中。

服务层建立在处理层之上，主要是为系统提供实体搜索的服务。对实体库中的实体进行索引的建立后，向外部提供搜索接口与数据，完成内部数据与界面的

交互工作。

最上面一层是显示层，主要是为系统提供数据显示的功能，同时接收外部查询条件的输入，提供系统与用户的交互功能，最终完成整个原型的工作流程

4.4 原型系统接口设计

面向行业的信息融合系统原型中，根据模型之间低耦合，小模块尽可能多复用的接口设计原则进行接口的设计。

各个模块封装的 API 分别如下：

（1）模板生成算法的接口，输入参数是所要进行模板生成的文件夹与网页相似度，接口描述如下所示：

```
/**
 * 模板生成方法(同步)
 * @param pageFolder 网页文件夹
 * @param outputFolder    模板生成后输出的位置
 * @param simIndex        相似度
 */
public void generateTemplate(File input, File output, double simIndex);
```

（2）抽取数据算法接口，输入参数是模板文件与所要进行抽取的网页文件，返回抽取结果集合，接口描述如下所示：

```
/**
 * 传入文件数组，抽取数据(同步)
 *
 * @param templateFile 模板文件
 * @param pages 网页数组
 * @return 抽取结果类集合
 */
public ResultCollection extractData(File template, File[] pages);
```

（3）文本实体的权重抽取算法，即实体与文本关系算法接口，输入参数是所要进行计算的文本与抽取实体个数，返回实体与实体权重的 map，接口描述如下所示：

```
/**
 * 文本中实体的权重抽取算法
 * 基于图排序算法的TextRank算法实现
 *
 * @param text输入的新闻文本
 * @param num 抽取实体个数
```

```
     * @return      实体与实体权重的Map
     */
    public Map<Object,Integer> extract(String text, int num);
```

（4）实体抽取算法接口，输入参数是所要进行实体抽取的新闻文本，返回实体与实体类型的对应关系，接口描述如下所示：

```
    /**
     * 实体抽取
     * 返回实体与实体类型的Map
     *
     * @param sInput        输入的文本
     * @return              返回实体与实体类型的Map
     */
  public Map<String,String>tag(String sInput);
```

4.5 本章小结

本章在第三章面向行业的信息融合原型系统的研究的基础之上，对系统进行了整体架构的设计。首先，本章阐述了本系统的相关设计概述，包括了设计面向行业的信息融合原型系统的动机和意义，并对本系统进行了简单的系统分析。然后，本章大致介绍了面向行业的信息融合原型系统的流程，包括了通过数据抽取后得到的百度百科的词条内容，利用支持向量机为分类算法对其进行分类处理；进行新闻文本处理，包括实体抽取、实体与文本相关度计算以及实体间关联度计算。最后，从整体上阐述了本系统的向外部提供的接口。

第五章 面向行业的信息融合原型系统的实现

本文所设计与实现的信息融合系统是面向IT行业的，信息的数据源是整个互联网，但是本文讨论的工具是在网页抓取的基础之上，而不关心所需的网页是如何获取的，只针对网络爬虫从互联网上爬取下来的网页为基础进行后续的处理。本文中，在开发面向行业的信息融合原型系统时，使用Java[26]作为主要编程语言。

5.1 网页信息抽取模块实现

网页信息抽取模块是本文原型系统的基础模块，实现目的是进行网页信息的精确抽取，为其它模块提供基础的文本数据。本节中将从实现的角度进行模块的介绍。

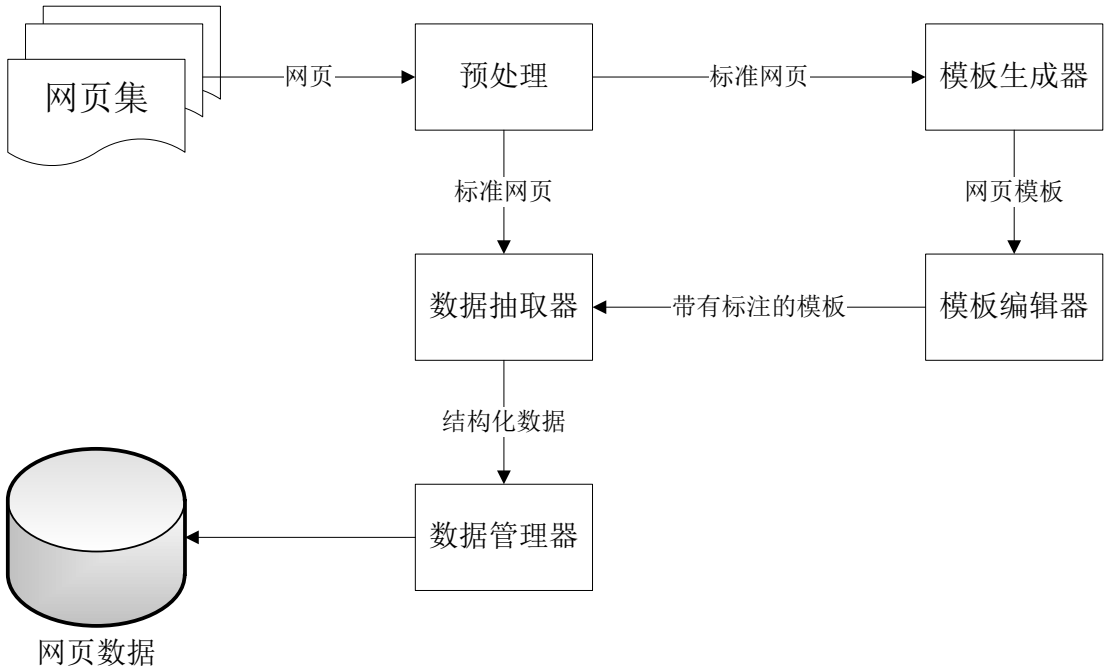

图5 - 1 网页精确数据抽取架构图

如图5-1所示，模块主要由以下几个部分组成：网页预处理功能模块，模板生成功能模块，模板编辑器，数据抽取功能模块和数据管理功能模块。

5.1.1 网页预处理

通常互联网上的HTML页面都是不规则的，非结构化的页面，不能直接转化成一个DOM树。如果我们需要访问或者抽取里面的内容的话，我们需要分析

HTML页面，去除垃圾，将其先转化成XHTML格式，XHTML的规范式如下：

表 5-1 XHTML的规范式

1)“<”和“>”只能用来包含网页中的标签，出现这两个符号须用<和>代替。
2）所有标记必须匹配。即每个开始标签都对应一个结束标签。
3）所有标签的属性值都必须放在引号中。如 。
4)所有标签必须是正确嵌套的。如<a>………是不正确的嵌套。

另外，WEB中很多的网页都会存在标签上的错误，例如空标签、script错误等。对于上述这些问题，本文实现了HtmlCleaner模块予以解决，它可以将HTML转化成XHTML，并能清除网页中的明显错误。HtmlCleaner[21]是一个开源的Html文档解析器。HtmlCleaner能够安全的解析和转换web上的HTML到标准的XML，重新排序每个元素，然后生成结构良好(Well-Formed)的XML文档。然后，用户可以提供自定义tag和规则组来进行过滤和匹配。它被设计的小，快速，灵活而且独立。HtmlCleaner也可用在Java代码中，当命令行工具或Ant任务。解析后编程轻量级文档对象，能够很容易的被转换到DOM或者JDom标准文档，或者通过各种方式(压缩，打印)连续输出XML。

5.1.2 层次聚类算法

层次聚类算法是一种发展比较早、应用广泛的聚类方法，分别有分解型层次聚类法和聚结型层次聚类法。本文使用一个聚结型层次聚类算法——使用代表点的聚类法(CURE)。

使用代表点的聚类法首先把每个单独的数据对象作为一个簇，每一步距离最近的簇对首先被合并，直到簇的个数为 K，算法结束。常用的距离公式如式 5-1：

$$\text{最小距离：} d_{\min}(C_i,C_j)=\min_{p\in C_i,p'\in C_j}|p-p'|$$

$$\text{最大距离：} d_{\max}(C_i,C_j)=\max_{p\in C_i,p'\in C_j}|p-p'| \qquad \text{式（5-1）}$$

$$\text{平均距离：} d_{\text{avg}}(C_i,C_j)=\frac{1}{n_i n_j}\sum_{p\in C_i}\sum_{p'\in C_j}|p-p'|$$

5.1.3 网页聚类

在本文中，使用代表点的聚类法进行网页聚类，在聚类实验中网页的数目大概是 500-1000，在这个复杂度上，可以采用类 CURE 算法，在本文的网页聚类实验中，距离定义为两个网页生成的 DOM 树的相似度，而计算两个类距离的时候，本文引入了部分树对齐算法（Partial Tree Alignment）来将同一类的点对齐

成一个点。部分树对齐算法时在[50]中被提出以在数据抽取中对多棵树对齐。它也可以用于对齐多个字符串。

该算法由一棵种子树（Seed Tree）开始逐渐增长，在逐渐增长的过程中对齐多棵树。其中种子树用T_s来表示，首先选定数据域数量最大的树作为种子树，随后对每一个T_i（i≠s），该算法为T_i中的每一个节点在T_s中找一个匹配的节点，如果对一个节点v_i无法找到匹配，则算法将尝试通过将v_i插入T_s来扩展种子树。扩充后的种子树T_s随后被用于后续的匹配。插入只有在v_i的一个位置在T_s中能够被唯一确定时再进行。否则，它会被留着不匹配。所以这个对齐是部分的（Partial）。它代表一种最少承诺方法（Least Commitment Approach）。早期不确定的承诺可能会给之后的匹配带来不良效果。由于最后对齐完的种子树可以直接转化为模板，本文在下一节模板生成中会在对其再进行详细介绍。

因为一个网页集合中簇的个数事先并不知道，CURE 算法当簇的个数为 K 时终止，所以在这里不可行的。因此采用新的终止条件，即当任意两个簇之间的距离都大于给定的距离闭值点时算法终止。整个聚类过程如图 5-2 所示：

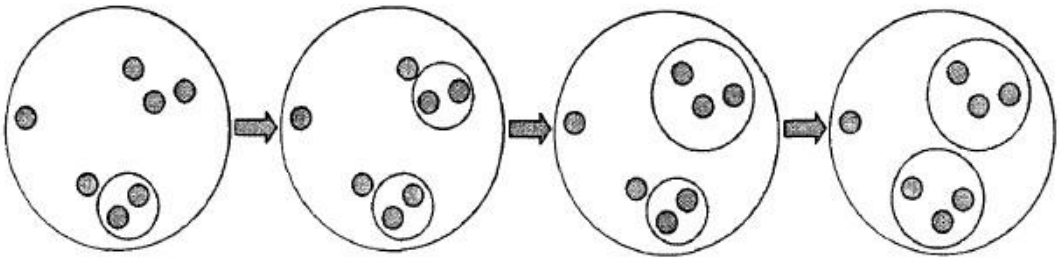

图5-2网页聚类算法过程

网页聚类是抽取模块的基础，后续的模板生成与数据抽取都是在网页聚类算法的基础上进行的进一步的处理。

5.1.4 改进简单树匹配算法

针对原始的简单树匹配算法，本文提出了如下改进，重新定义了两个节点间的关系：

E(equal): 标签的名字和属性都相同；

D(different)：标签的名字不同

S(similar)：其他情况

并且提出了启发式的方法，利用数据项里通常用“：”“-”等符号来分隔数据名称和数据值，来对标签加入语义属性，从而增加标签属性信息。

```
Algorithm: Extended_STM(A, B)
1. if the roots of the two trees A and B are of relation D
2.    return 0;
3. else
4.    m := the number of first-level sub-trees of A;
5.    n := the number of first-level sub-trees of B;
6.    Initialization: M[i,0] := 0 for i = 0, …, m;
7.                    M[0, j] := 0 for j = 0, …, n;
8.    low = 1;
9.    for i = 1 to m do
10.      for k = 1 to low do
11.         m[i,k] = m[i-1,k];
12.      endfor;
13.      j = low;
14.      while j < m do
15.         M[i,j] := max(M[i,j-1], M[i-1, j], M[i-1, j-1]+W[i, j]);
16.            where W[i,j] = Extended_STM(Ai, Bj)
17.             j++;
18.             if the roots of the two trees Ai and Bj
19.                are of relation E
20.             then
21.                low = j;
22.                break;
23.             endif;
24.       endwhile;
25.       for k = j to m do
26.          m[i,k] = m[i,k-1];
27.       endfor;
28.    endfor;
29. endif;
30. return (M[m, n]+1)
```

图 5-3 扩展的简单树匹配算法

如图 5-3 所示拓展的简单树匹配大体上思路和简单树匹配一样，但当两个节点(i，j)关系被判断为 E 时，则会将两棵树对于(i，j)进行一个“绑定”，即认为匹配的节点中一定存在节点对(i，j)。原始的 STM 算法只考虑标签，在遇到存在多种可能的最大匹配情况就会出现错误。

5.1.5 模板生成

模板生成是在网页聚类的结果上，对属于同一类的网页进行共性分析，从而生成这类网页的最佳抽取模板。需要指出的是，生成的模板并不一定能表达用于生成这个模板的聚类集合中的全部网页。因为尽管聚类集合中的网页的相似度较

高，但有些网页中仍会存在部分内容不是共有的，这些内容出现的次数很少，因此不认为它们属于模板。

对于网页聚类后的每一个网页簇，都会生成一个对应的抽取模板，根据上一节讨论的网页聚类算法，聚类的过程就是不断地对其种子树进行扩充，当算法终止时，生成多少个簇，就有多少棵对应的种子树，因此定义该种子树对应的网页就是该簇中所有网页的模板。因此，网页聚类过程其实就是模板生成的过程，本质上其实是多棵 HTML DOM 树的对齐过程。图 5-3 给出了基于两棵树部分对齐的多棵树对齐的完整算法。S 是输入的树集合。S 是输入的树集合，用图 5-4 中一个简单的例子来解释这个算法。其中 S 包含三棵树。

Algorithm PartialTreeAlignment(S)
1. Sort trees in S in descending order according to the number of data items that are not aligned;
2. T_s = the first tree (which is the largest) and delete it from S;
3. *flag* = false; $R = \varnothing$; I = false;
4. **while** ($S \neq \varnothing$)
5. T_i = select and delete next tree from S;
6. Simple_Tree_Matching(T_s, T_i);
7. L = alignTrees(T_s, T_i); // based on the result from line 6
8. **if** T_i is not completely aligned with T_s **then**
9. I = InsertIntoSeed(T_s, T_i);
10. **if** not all unaligned items in T_i are inserted into T_s **then**
11. Insert T_i into R;
12. **endif**;
13. **endif**;
14. **if** (L has new alignment) or (I is true) **then**
15. *flag* = true
16. **endif**;
17. **if** $S = \varnothing$ and *flag* = true **then**
18. $S = R$; $R = \varnothing$;
19. *flag* = false; I = false
20. **endif**;
21. **endwhile;**
22. Output data fields from each T_i to the data table based on the alignment results.

图 5-4 部分树对齐算法

第 1 至 2 行（图 5-4）找出拥有最多数据项的树。它被当做种子树 T_s 使用。在图 3-9 中，种子树是第一棵树（略去了 T_1 左边的许多节点）。第 3 行初始化 R，它用于存储每一轮迭代中没有和 T_s 完全对齐的树，第 6 行进行树匹配。

第 7 行通过追踪第六行的矩阵结果找出所有匹配上的节点对。这函数与用编

辑距离对齐两个字符串相似。第 8 行和第 9 行尝试将未匹配的节点插入到T_s中，也就是上面讨论过的部分树对齐。在图 5-4 中，T_2中的节点 w、c、k 和 g 没有一个可以被插入到T_s中，因为找不到唯一的位置。于是它不会被通过 if-语句（5-3 第 9 行中 InsertIntoSeed()返回 false）。第 13 至 14 行将T_2插入 R，它是一个需要重新匹配的树列表，因为一些数据项没有被对齐，也没有被插入到T_s中。在图 3-10 中，当下一轮迭代中匹配T_3和T_s时，所有未匹配的节点 c、h 和 k 都可以被插入到T_s中（第 9 行）。由于有一些插入，重新匹配 R 中的那些树。第 10 行和第 11 行将 R 中的树放到 S 中并重新初始化 R。T_3将不会被插入到 R 中（第 13 行）。

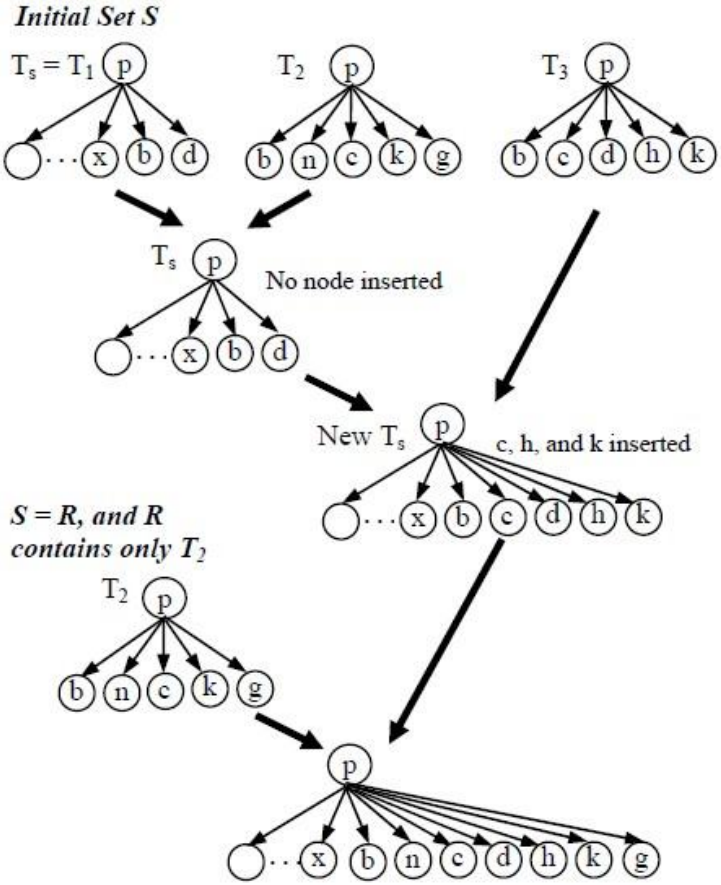

图 5-5 有两轮迭代的迭代树对齐

在图 5-5 中，T_2是 R 中唯一将在下一轮和新的T_s匹配的树。现在，T_2中的每一个节点都可以被匹配或者插入，于是这个过程就完成了。图 5-4 中第 18 行根据产生出来的对齐每一棵树输出数据项。注意如果在算法完成后仍然有未匹配的含数据的节点（比如 R≠ø），那么每一个未匹配的数据项将单独占据一列。图 5-6 展示了图 5-5 中树的数据表。用“1”来表示一个数据项。

该算法的复杂度为 O(k^2n^2)，这里 k 是 S 中树的数量，而 n 是每棵树的大小（假设所有树大小都相似）。然而，实际上该算法几乎总是只遍历 S 一遍（即

R=ø）。

	…	x	b	n	c	d	h	k	g
T_1	…	1	1			1			
T_2			1	1	1			1	1
T_3			1		1	1	1	1	

图 5-6 最终数据表 (“1”表示一个数据项)

事实上，为了使该算法完备，一个递归调用被加到图 5-4 第 17 行后面以处理 R≠ø时的情况，即仅对 R 中的那些树进一步对齐。下列三行可以被加入：

18. if R≠øthen

19. PartialTreeAlignment(R)

20. endif

这三行照顾到了有些数据项没有被对齐也没有被插入的情形。通过这个完备的算法：首先，即使没有对种子树进行对齐或插入，递归也会终止的，因为种子树在每一次递归中被删除，从而 R 变得越来越小。第二，该算法能够从数据中找出多个模板。每一次递归中的种子树都代表一个不同的模板。

5.1.6 模板标注

语义标记是一个用户交互的过程，工具通过可视化界面的方式引导用户对网页标签进行语义标记，并将标记的语义信息保存在该标签的semantic属性中。如图5-7所示：

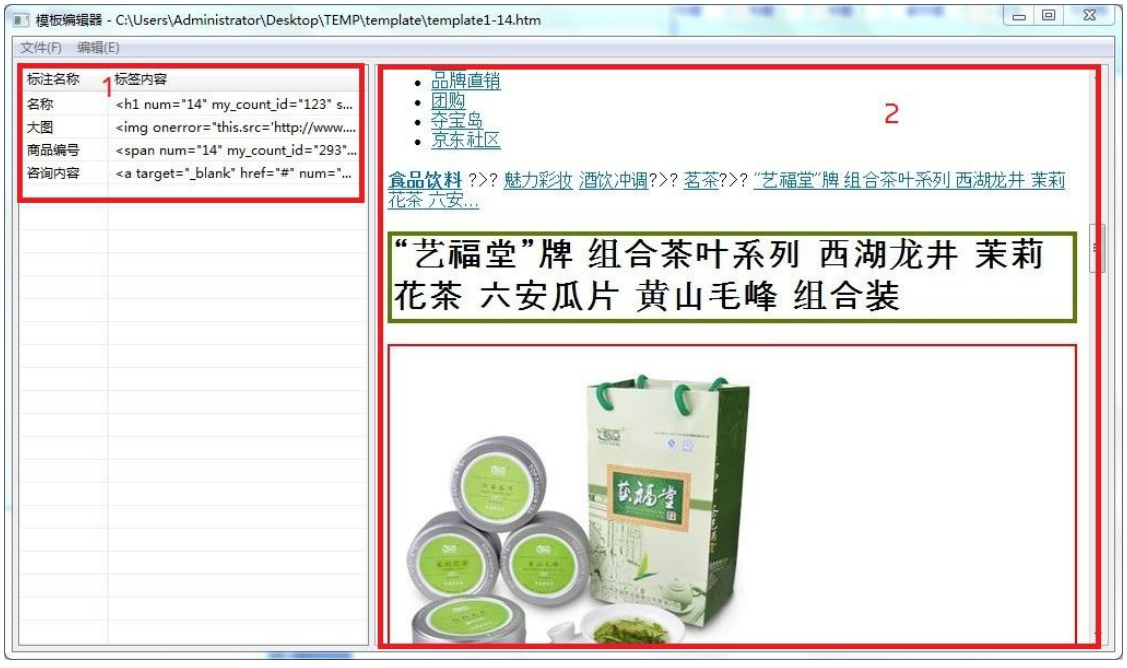

图 5-7 模板编辑界面

图5-7中即区域1是显示用户的标注标签，其中列出现的是用户已经标注的标签；区域2是用户的编辑区，用户可以在所要选择标注的标签上双击选中标签后，将标注的信息添加到标注标签列表中。

5.1.7 数据抽取

数据抽取是一个相对简单的过程。利用用户标记过的模板，对指定的网页抽取出用户感兴趣的数据。抽取的信息可以按需要放到指定的文件（XML，TXT）或者用户数据库中，本文抽取的数据作为实体词典构建与实体关联模型的基础数据。

5.2 实体词典构建模块实现

本文在前面的篇章中已经阐述了实体词典的作用与重要意义。尽可能广泛地覆盖相关领域、尽可能多的包含每个实体的不同表述，是本文构建实体词典的最高目标，因为实体词典的覆盖度越高，实体抽取的结果就会越准确性，一个尽可能完备的词典能显著提高实体抽取的结果就会越准确率，同时对后续的各个流程的处理就越准确，所以说实体词典的作用不可小觑，可以说是非常的重要。

因为中文的博大精深，清朝《康熙字典》收字47035个，当代的《汉语大字典》(2010年版)收字60370个，由这些字组成的词语更是数不胜数，而且新词又在不断的增加，更麻烦的是相同词语在不同语境中可能代表不同的语义，所以目前还没有一个完整的中文词典能够涵盖所有领域。本文中，我们提出了基于百科百科的词条分类的方法构建实体词典，在很大程度上减少了人工收集实体的工作量。

5.2.1 百度百科数据抽取

利用百度百科的网页数据，我们使用网页精确信息抽取工具，可以很方便的实现词条信息的抽取任务。

图 5-7 百科百科数据抽取

本文主要选择了百科条目中一些更能代表词条词性的属性进行抽取，如图5-7中红色线框部分，词条的名片是一个词条介绍的一个摘要，本文再进一步的将文本中的超链接抽取出来做为分类的特征，同时还有一些介绍的抬头，也是一个词条词性最具有代表性的信息之一。

5.2.2 构建文本空间向量

本文中，在实体词典的分类算法中，我们要将百度百科上的文本进行分类，要做的第一步就是要将文本向量化，然后在向量空间中对文本进行研究。本文中，我们使用WVTool开源软件，对文本进行处理，实现文本向量化。WVTool是一款开源软件，它主要做文本词频方面的处理，对于实现文本向量化非常有用。它支持对文本、半结构化内容（Html、XML）的向量化处理功能。

对于一篇文章来说，它是由很多的词来组成；而对于一个数据集来说，它是由一个包含该数据集中的所有文章的词组成的。现在我们要对不同文档进行一个相似度的比较，将文档用数字化的东西表示出来，即向量的模式。那么把数据集中的所有词作为向量的某一维，如果某一个词在该文档里出现我们标示它为1，否则标示为0。通过这样的一个过程我们可以得到这样的一个向量(0,1,1,1,……)，那么对于数据集中的所有文档，我们就可以用一个向量空间来表示了，对于表示

好的文档，我们在向量的最后一位或者最前一位标示出它的类别信息，这样一个包含类别信息的向量空间就表示出来了。

本文中，我们通过TF-IDF的形式对每一个词进行加权，因为每个词它在表示文档是它的权重应该是不同的，这个权重是通过该词的词频和在某个文档中出现的次数来决定的，源代码如下所示，使用WVTool可以很快速的创建文本文档的向量空间表示，并使用TF-IDF对向量空间模型中的词权进行调整，其详细代码如下所示：

```
        //初始化一个WVTool对象
        WVTool wvt = new WVTool(false);
        //初始化一个configuration对象,并将文本权重设置为TFIDF
        WVTConfiguration config = new WVTConfiguration();
config.setConfigurationRule(WVTConfiguration.STEP_VECTOR_CREATION,
new WVTConfigurationFact(new TFIDF()));
        //实现分词功能
        Tokenizer tk = new ChineseTokenizer();
        //定义一个只有4个类别的输入集合，即整个样本集合仅有2个类别
        WVTFileInputList list = new WVTFileInputList(2);
        //添加训练分类，第1个参数是类别放置的文件夹，最生一个参数是类别
        list.addEntry(new WVTDocumentInfo("./0", "txt", "", "", 0));
        list.addEntry(new WVTDocumentInfo("./1", "txt", "", "", 1));
        //生成wordList
        WVTWordList wordList = wvt.createWordList(list, config);
         //对wordList中DocumentFrequency做出一个限制，即DF在1<n<1000之间
        wordList.pruneByFrequency(1, 1000);
         //生成向量空间
        wvt.createVectors(list, config, wordList);
```

本文中，为了提高分词的准确度，我们使用中科院的分词系统来实现中文处理的类WVTTokenizer, TokenEnumeration这两个接口，通过包装中文分词工具来实现WVTool中的中文处理功能，使得分类效果更加精确。

5.2.3 分类训练

本文中使用 Salton 于 1969 年提出的向量空间模型来对文本进行表示，在进行分类之前需要分词处理，同样的，文中使用的 ICTCLAS 开源分词系统来实现。SVM 分类器通常具有较高的分类精度，总的来说，SVM 能够较好的解决小样本，非线性，高维数识别和局部极小点等问题。

目前使用得最多的支持向量机的软件工具主要有 LIBSVM 和 SVMLight。本文使用的是开源的 LibSVM[22]实现 SVM 分类。LIBSVM 是台湾大学林智仁(Chih-Jen Lin)博士等开发设计的通用 SVM 软件包。LIBSVM 使用的一般步骤如

表 5-1 所示：

表 5-1 LIBSVM 使用步骤

步骤	方法
步骤 1	按照 LIBSVM 软件包所要求的格式准备数据集
步骤 2	对数据进行简单的缩放操作；
步骤 3	虑选用 RBF 核函数 $K(X_i, X_j) = \exp(-\gamma \| X_i - X_j \|^2), \gamma > 0$
步骤 4	采用交叉验证选择最佳参数 C 与 g ；
步骤 5	采用最佳参数 C 与 g 对整个训练集进行训练获取支持向量机模型；
步骤 6	利用获取的模型进行测试与预测。

在对文本进行分类之前，我们必须先进行文本分类的训练，其代码如下

```
Public void svm_problem train(List<Integer> types, svm_parameter param){
     problem = new svm_problem();
     problem.l = types.size();
     double[] y = new double[problem.l];
     for(int i=0; i<problem.l; i++) {
          y[i] = types.get(0);
          types.remove(0);
     }
     problem.y = y;
     this.types = null;
     svm_node[][] x = new svm_node[problem.l][];
     for(int i=0; i<problem.l; i++) {
         x[i] = values.get(0);
        values.remove(0);
    }
    problem.x = x;
    this.values = null;
    SVMService.train(problem, param);
}
```

其中 List<Integer> types 是从构建出来的空间向量的相关值，svm_parameter param 是对 SVM 的控制参数。

本文的训练语料是从百度百科的网页中抽取出来的，按照本文词典的几个不同分类，我们相应的将语料进行分类，分别为人名、机构名、专有名词以及其它类别。其训练语料的分布如下图所示：

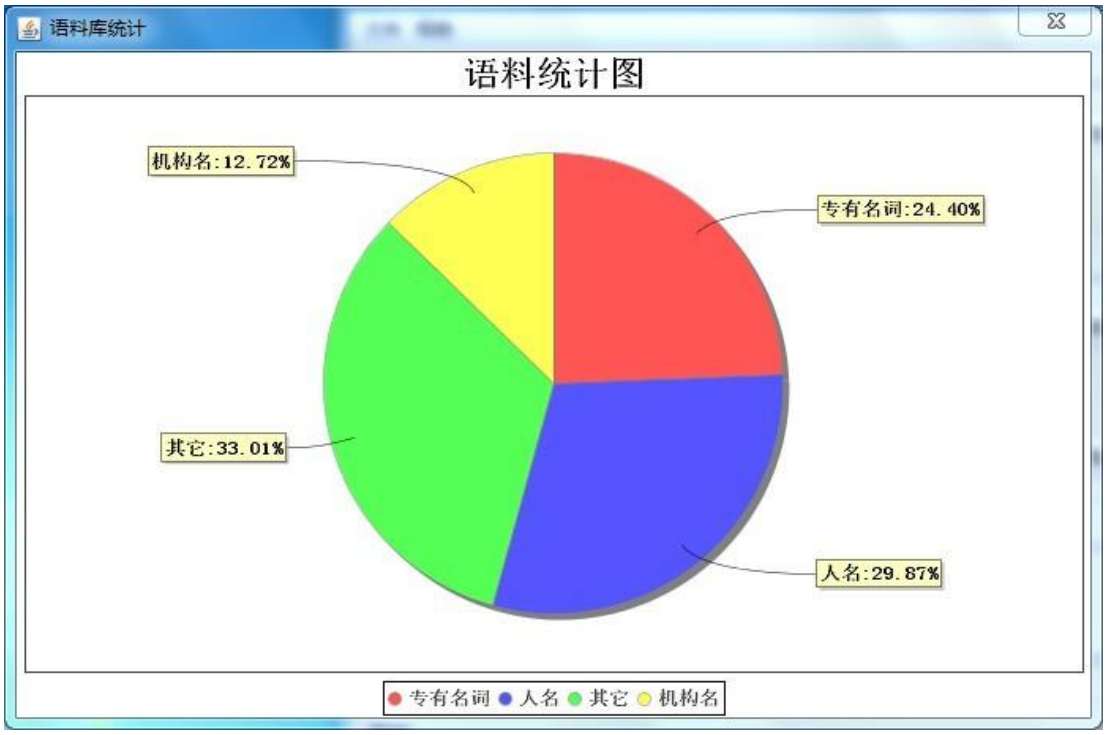

图 5-8 百科百科语料库统计

如图 5-8 所示，本文从百度词条中，人工的将 2000 多条词条进行分类，主要分成人名、机构名、专有名词三个实体词典所需的分类，剩余的归入其它分类中，最后使用上述的 SVM 文本分类器对词条进行分类。

5.2.4 分类预测

在上一小节中，我们已经完成了模型的训练，到此我们便可以训练的模型进行百科词条的分类，其分类代码如下：

```
public double predict(String doc) {
    WVTWordVector vector = null;
    WVTDocumentInfo info = new WVTDocumentInfo("", "", "", "");
    vector = wvtool.createVector(doc, info, wvtConfig, wordList);
    svm_node[] nodes = new svm_node[vector.getValues().length];
    for(int i=0; i<nodes.length; i++) {
        nodes[i] = new svm_node();
        nodes[i].index = i;
        nodes[i].value = vector.getValues()[i];
    }
    return svm.svm_predict(model, nodes);
}
```

参数doc是传入的文本，程序文本转换成空间里的一个文本的向量，在已经训练完成的模型中使用svm分类算法进行分类，最终返回一个double型的分类值，

与分类值最近的分类即是词条对应的分类。

5.3 实体关联模块实现

信息融合系统的具体实现除了之前章节中介绍的网页精确信息抽取、实体词典和 IT 新闻语料库的收集之外，最重要的部分是实体关联模型的实现，其中包括了实体的抽取、文本与实体的关系以及实体间的关联度的计算，通过算法的计算后，这些信息将会被存入数据库中，以备后续的操作所调用。

5.3.1 新闻类语料的准备

因为没有现成的IT类新闻语料库支持，本文通过使用网络爬虫系统在网络上进行爬取，获取IT类新闻领域的文本作为语料库，来源主要是各大主流网站中关于IT行业的新闻报道，最终收集到的关于IT领域的新闻网页203318个，其中包含新浪、网易、搜狐、腾讯等门户网站，以及36氪、伯乐在线、月光博客等国内知名的IT行业博客，同时将其网页的URL记录下来，以备系统的后续处理与更新工作。同样的，本文利用网页精确信息抽取工具，将抓取下来的网页进行内容的抽取，主要包含标题、内容、信息来源等相关信息。

5.3.2 分词模块调用

ICTCLAS 开源分词系统基于层叠隐马尔科夫模型与 N-最短路径粗切分[25]的基础上进行切分标注结果。该系统对于使用其它语言开发的使用者提供了相应的概率词典和一套完整的动态连接库，开发者可以直接在自己的系统中调用ICTCLAS，而不必清楚汉语词法的分析过程，只需要在分词和词性标注的基础上进行调用即可完成自己应用的开发需要。同时因为使用 C++作为开发语言其运行速度非常快，而且系统的准确率也很高。

因为信息融合系统是使用 Java 语言开发的，在使用 Java 调用 ICTCLAS 开源分词系统时，并不能直接使用 Java 调用现成的 ICTCLAS.dll 动态链接库文件，所以分词模块的调用采用 JNI 技术，实现了 ICTCLAS 分词模块的无缝链接。在信息融合系统中首先定义一个 CWSTagger 类，其代码示例如下所示：

```
public class CWSTagger {
    public static native boolean ICTCLAS_Init(byte[] path,int encoding);
    //encoding=0,设置为GBK；encoding=1设置为UTF8,encoding=2设置为BIG5
    public static native boolean  ICTCLAS_Exit();
    public native   int ICTCLAS_ImportUserDict(byte[] sPath);
    public native float ICTCLAS_GetUniProb(byte[] sWord);
```

```
    public native boolean ICTCLAS_IsWord(byte[] sWord);
    public native byte[] ICTCLAS_ParagraphProcess(byte[] sSrc,int
bPOSTagged);
    public native  byte[] nativeProcAPara(byte[] src);
    static {
        System.loadLibrary("ICTCLAS");
    }
}
```

其中，遵循 JNI 规范的中间层实现对开源分词系统接口函数的封装，当信息融合系统调用分词系统时，通过 system.loadLibrary 方法加载动态链接库。

以下代码是 CWSTagger 类的调用示例，对字符串进行分词处理，前得到分词后各个各个词的词性。

```
CWSTagger tagger= new CWSTagger();
String argu = ".";
System.out.println("ICTCLAS_Init");
if (tagger.ICTCLAS_Init(argu.getBytes("GB2312"),0) == false){
    System.out.println("Init Fail!");
    return;
}
String sInput = "这是一个测试句子。";
// 设置词性标注集 2 北大二级标注集
tagger.ICTCLAS_SetPOSmap(2);
byte nativeBytes[] =
tagger.ICTCLAS_ParagraphProcess(sInput.getBytes("GB2312"), 1);
String nativeStr = new String(nativeBytes, 0, nativeBytes.length,
"GB2312");
System.out.println("分词结果： " + nativeStr);
```

根据上述的调用过程，其输出结果为：

```
这/rzv 是/vshi 一个/mq 测试/vn 句子/n 。/wj
```

句子已经被分成一个个字或词，每个字或词后跟着词性，用“/”分开，由此可以对分词后结果再进行一次处理，便可以得出每个词以及词所对应的词性。

5.3.3 实体抽取

在上一小节中已经介绍了如何在 Java 程序中调用 ICTCLAS 分词模块进行文本的词性的分析，通过分词模块的处理后，我们就可以进行实体的抽取。例如下列文本，其中包含的实体有：Linux、简锦源、操作系统等。

```
日前/t ，/wd 香港/ns Linux/nz 商会/n 会长/n 简锦源/nr 在/p 广州/ns 信息/n 产
业/n 周/qt 上/f 指出/v ，/wd 由于/c 手机/n 、/wn 超/v 小型/b 笔记本/n 等/udeng
移动/vn 互联网/n 终端/n 的/ude1 出现/vn ，/wd 这种/r 移动/vn 终端/n 设备/n 采
```

```
用/v Linux/nz 平台/n 作为/v 操作系统/nz 已经/d 成为/v IT/x 业/ng 界/n 的/ude1
一/m 种/q 发展/vn 趋势/n 。/wj
```

其中，nr 指代的是人名；org 指代机构名；nz 指代专有名词，通过实体词典的添加，我们实体的抽取的性能可以在很大程度上得到提高，其实现代码如下：

```
List<Word> list = new ArrayList<Word>();
List<String> listTemp = new ArrayList<String>();
listTemp = Arrays.asList(str.split("\\s+"));
String wordRaw;
String word;
String tagger;
int length= listTemp.size();
for(int i = 0; i < length; i++){
    wordRaw = listTemp.get(i);
    if(wordRaw.lastIndexOf("/") < 0){
        continue;
    }
    word = wordRaw.substring(0, wordRaw.lastIndexOf("/"));
    tagger = wordRaw.substring(wordRaw.lastIndexOf("/")+1);
    if(!isStopWord(word)){

        Word newWord = new Word(word, tagger);
        list.add(newWord);
    }
}
```

其中 Word 是抽取词类，包含词与词的词性两个属性，我们可以通过词性过滤的处理就可以将实体抽取出来。

5.3.4 实体与文本相关度计算

文本经过TextRank算法计算后，我们构建了Web对象之间的关系图。通过这个对象关系图进行链接分析，我们可以计算得到一个Web对象的重要性，即web实体与文本的贡献度。这是在传统的网络图不可能获得的信息。

文本经过TextRank算法计算后，我们构建了Web对象之间的关系图。通过这个对象关系图进行链接分析，我们可以计算得到一个Web对象的重要性，即web实体与文本的贡献度。这是在传统的网络图不可能获得的信息。有如下一段文本：

全球 200 万名 Linux 开发者终于等到了这一天，是时候对微软说“不”了，因为“云计算”时代即将来临，以及廉价的、超小型笔记本电脑正在快速普及。日前，香港 Linux 商会会长简锦源在广州信息产业周上指出，由于手机、超小型笔记本等移动互联网终端的出现，这种移动终端设备采用 Linux 平台作为操作系统已经成为 IT 业界的一种发展趋势。简锦源指出，

在过去几年里，以 Linux 为代表在开放源码软件的使用率正在不断上升，已经开始挑战微软的桌面平台和办公软件等等产品。简锦源透露，开放源码的软件在服务器端占有率已经超过 50%，而在中国新的电脑中预装和捆绑 Linux 的，每年都超过 200 万台。

根据TextRank算法，我们可以得出各个词在整个文本中的得分，通过归一化后，我们可以得每个词对应的一个取值为[0,1]的数值，即每个词对这一文本的贡献度。

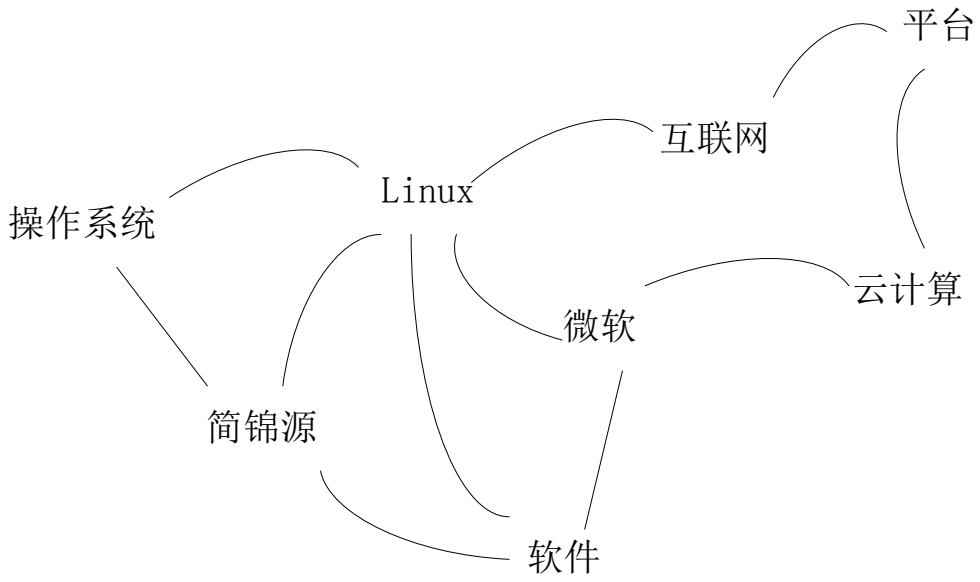

图 5-9TextRank 算法中实体引用

在 IT 行业中，软件、平台等是比较重要的名词，所以将其定义为实体，如上图 5-9 所示，给出了文本中实体的部分引用的关系。

5.3.5 实体间关联度计算

在上一小节中，我们提出了基于图排序算法 TextRank 构建实体间的引用关系，根据实体的引用与被引用的关系计算出现实体在文本中的贡献度，即实体与文本的相关度，第二章中，我们已经推导出了实体间的关联度计算的式（11），所以我们便可以根据下面的算法计算出现实体间的关联度：

```
18  Relevancy(A,B)
19  begin
20     if A ∧ B = ∅
21        return 0
22     endif
23     let m be the number of artile
24     V[0,i] = A[i]    for i=0,…,m
25     V[1,i] = B[i]    for j=0,…,m
```

```
26      for i = 0 to m
27          M += A[i]* B[i]
28          a += A[i]*A[i]
29          b += B[i]* B[i]
30      end
31      return M/ √a * √b
32  end
```

从上面所示的算法中，可以计算出现每对实体之间的关联度的期望值，对应的就是这对实体在众多的文章中所存在的关系。

5.4 系统界面

用户在系统主页的搜索框中，输入所要搜索的实体后进行搜索操作，系统就会从实体库中检索相应的实体，并将实体秘文本关系库和实体间关系库中搜索出相关的信息显示在检索结果界面上。如下图 5-10 所示，为检索结果界面：

图 5 - 10 百度百科词条特征

其中标注 1 是用户所输入的实体，标注 2 是从实体间关系库中取出的与搜索实体关联关系值最大的相关实体，标注 3 是从实体与文本关系库中与实体相关度值最大的相关新闻。

5.5 本章小结

本章详细介绍了面向行业的信息融合系统原型系统的实现，从系统的构建方面详细的介绍了整个原型系统的实现过程，主要工作包括网页精确抽取的实现、实体词典的构建、新闻语料库的准备以及实体关联模型的实现。

首先，因为网页数据的抽取是本文的一个基础，所以本章先从网页精确信息

抽取的实现开始，介绍了网页信息的抽取的算法实现，其中包括网页的聚类、模板生成等算法的实现。

然后，本章介绍了本系统所使用的实体词典的构建方法，基于百度百科抽取出来的数据，使用支持向量机的分类器进行词条词性的分类；接着介绍了面向 IT 行业的新闻语料的准备过程，同样使用了网页精确数据的抽取的方法来实现。

最后，简单介绍了实体关联模型的实现，简单介绍了分词模块的调用和实体的抽取，同时详细的介绍了实体与文本相关度、实体间关联度的计算这两个实体关联模型中的核心思想的实现。

第六章 实验结果分析

本章为了证明原型系统中各算法的精确度，进一步对系统运行结果进行说明，分别进行了网页抽取、词条分类算法、实体与文本相关度以及实体间关联度的实验，并对实验结果进行分析。

6.1 测试的性能指标

本文采用传统的性能指标参数对文中所使用的算法进行性能方面的评估，其中包括查全率（Precision）、召回率（Recall）性能指标，查全率和召回率分别在两个不同的方面反映了算法的性能，两者必须综合考虑，下面将给出这些指标的数学表达式。

准确率在本文中用来测试本文所使用算法的准备度，即是被分类的文本中分类正确的文本所占的比率，其表达式如式 6-1：

$$准确率(precision)=\frac{分类正确的文本数}{实际分类的文本数} \qquad 式（6\text{-}1）$$

召回率在本文中用来测试本文所使用算法的查全率，是人工分类结果应有的文本中分类正确的文本所占的比率如式 6-2：

$$召回率(\text{recall})=\frac{分类正确的文本数}{应有的文本数} \qquad 式（6\text{-}2）$$

6.2 对比实验及结果

本文通过使用网络爬虫系统在网络上进行爬取，获取 IT 类新闻领域的文本作为语料库，建立了一个较大规模的语料库，来源主要是各大主流网站中关于 IT 行业的新闻报道，最终收集到的关于 IT 领域的新闻网页 203318 个，其中包含新浪、网易、搜狐、腾讯等门户网站，以及 36 氪、伯乐在线、月光博客等国内知名的 IT 行业博客。

6.2.1 网页抽取性能评估

网页信息抽取算法作为本文的基础，其精确度直接影响到后续对文本数据的处理精度，所以网页信息抽取算法的性能尤为重要，从以上算法可以流程可以知

道本文使用的网页信息抽取算法主要分为模板生成、模板标注以及数据抽取三个重要步骤。

在本次测试中，本文从 360buy.com、36kr.com，sina.com 等大型网站采集了大概 3000 个真实的网页作为实验样本，并且从每个网页中抽取了多个数据项，例如对于电子商务网站，抽取了名称、价格、描述；对于新闻网站，抽取了标题、作者、日期、内容等。

表 6-1 网页抽取实验结果

网页名称	准确率（%）	召回率（%）
新浪新闻页	100	97.2
36kr 博客页	100	98.5
163 新闻页	99.38	93.12
搜狐新闻页	94.07	91.01
腾讯新闻页	95.81	90.8
sina 日志页	96.52	91.36
sohu 博客首页	94.09	94.13
sohu 日志页	95.58	90.9
163 微博	94.61	90.09
sina 微博	96.19	92.0
sohu 微博	95.28	91.72
百度百科	98.3	96.8

从实验结果表 6-1，可以看到对于 B2C 电子商务等一些网站，该方法表现得比较出色，原因在于这些网页都具有比较统一的模板。而对于一些新闻类别的网站，准确率和查全率有所下降，通过分析这些新闻网站的网页结构发现新闻的文本内容一般都具有不同的分段、大小并且会在不同地方嵌入图片，这些都会影响网页的结构，也就是说这些网页不能很好的聚合在一起生成统一的模板，不过值得欣慰都是实验结果显示的效果基本上能够满足需求。

6.2.2 词条分类性能评估

对于词条分类来说，其实质上就是对词条中的文本进行分类，通常用查全率（Precision）、召回率（Recall）和 F1 这几个指标对分类系统的精度好坏进行评估。在本次测试中，我们从百度百科的网页抽取数据中选择了 1000 个词条，利用已经训练好的模型，选择不同的文本权重进行词条分类测试，其结果如下表 6-2 所示，词条分类算法查全率和召回率都可以达到 80%以上，比当前文本分类算法的总体精确度稍高，因为中文处理的复杂性词典分类的结果的精确度相对较低，但是基本上可以满足系统的使用要求。

表 6 - 2 词条分类实验结果

特征权重	Precision (%)	Recall (%)	F1(%)
熵权重	90.65	80.77	86.30
词频	91.3	81.82	86.30
TDITF	92.83	84.91	88.12

6.2.3 文本与实体相关度性能评估

文本与实体之间的相关度是本文中的重点，直接关系到整个信息融合系统的性能，对后续实体间关联度的计算的精确度也至关重要。在进行该算法的测试时，我们从 203318 篇新闻语料中随机的挑选 50 篇文章作为测试样本，对算法进行精确度的测试实验。

首先使用本文中的实体抽取算法将篇章中的实体抽取出来，使用文本的基于图排序算法的文本与实体相关度算法进行计算，得到文本中各个实体与文本之间的关联度。然后，10 位标注人员分别对 50 个文本中的抽取出来的实体人工的进行相关度标注。标注方法是标注人员根据自身的认识与经验，借助与互联网等工具对文本与实体的相关度给出分值，分值的范围为 0 到 1 之间，标注完成后进行平均，得到文本与实体的相关度值。最后，将标注的相关度值与计算出的相关度进行比较。

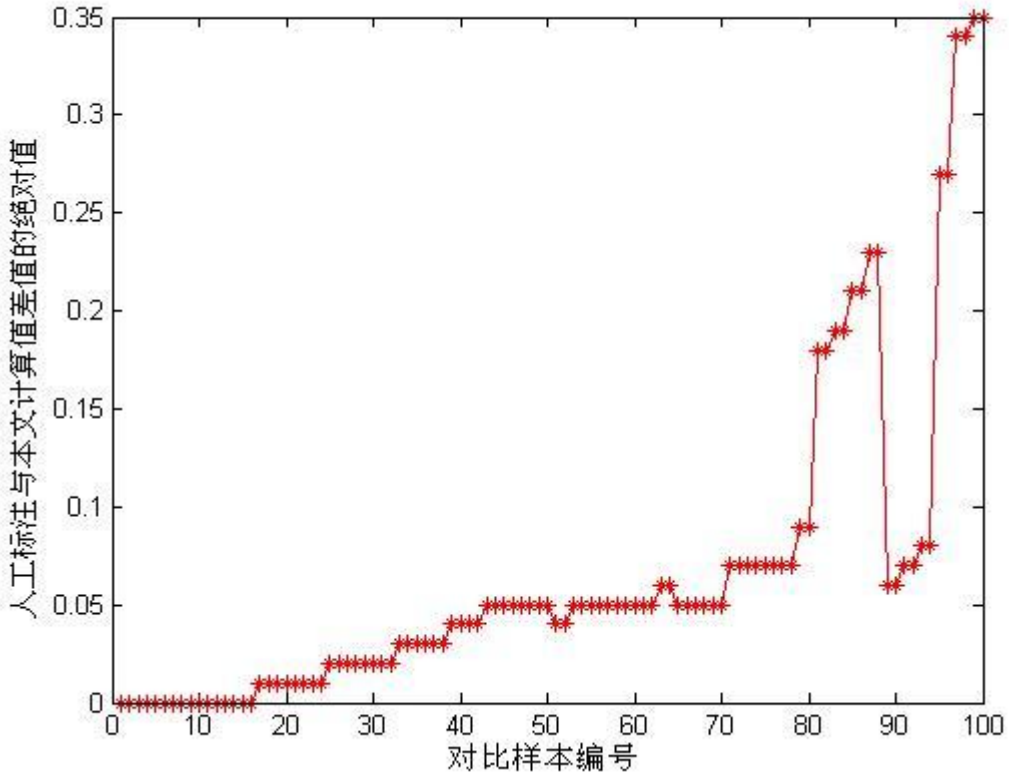

图 6-1 人工标注与本文算法的数据对比

如图 6-1 所示，我们测试数据中随机选出 100 个实体，图中表示的是人工标注与算法计算出来的文本与实体的相关度值两者差的绝对值，从图中我们可以看出，大部分的差值集中在 0.15 以下，只有不到十分之一的值高于 0.2 且最大的差值在 0.35 左右，算法的计算值基本上与人们的认知相吻合。

6.2.4 实体间关联度性能评估

实体间的关联度是在文本与实体相关度的基础上，经过向量运算得出的。在实体间关联度实验中，我们从 84639 对实体中随机挑选出现 200 对（400 个）实体进行算法精确度的测试。

与上一小节的实验相类似，首先从实体间关系数据库中随机选取 200 对实体关联度的数据，然后，10 位标注人员分别对实体间的关联度进行标注，标注方法是标注人员对实体间的关系，借助与互联网等工具对实体关联度给出分值，分值的范围为 0 到 1 之间，标注完成后进行平均，得到文本与实体的相关度值。最后，将标注的相关度值与计算出的关联度进行比较。

我们计算了几个常见的命名实体，如比尔盖茨和微软、苹果公司和乔布斯、Windows 与微软、Android 和 Google 的关联度，计算结果如表 6-3 所示。

表 6-3 标注值与本文算法的关联度对照表

实体	*实体*	*标注值*	*关联度*	*实体*	*实体*	*标注值*	*关联度*
盖茨	微软	0.94	0.98	乔布斯	苹果	0.91	0.85
Windows	微软	0.93	0.9	Android	Google	0.89	0.95

如图 6-2 所示，横轴表示实体对序号，竖轴表示人工标注与算法计算的关联度值之间差值的绝对值。

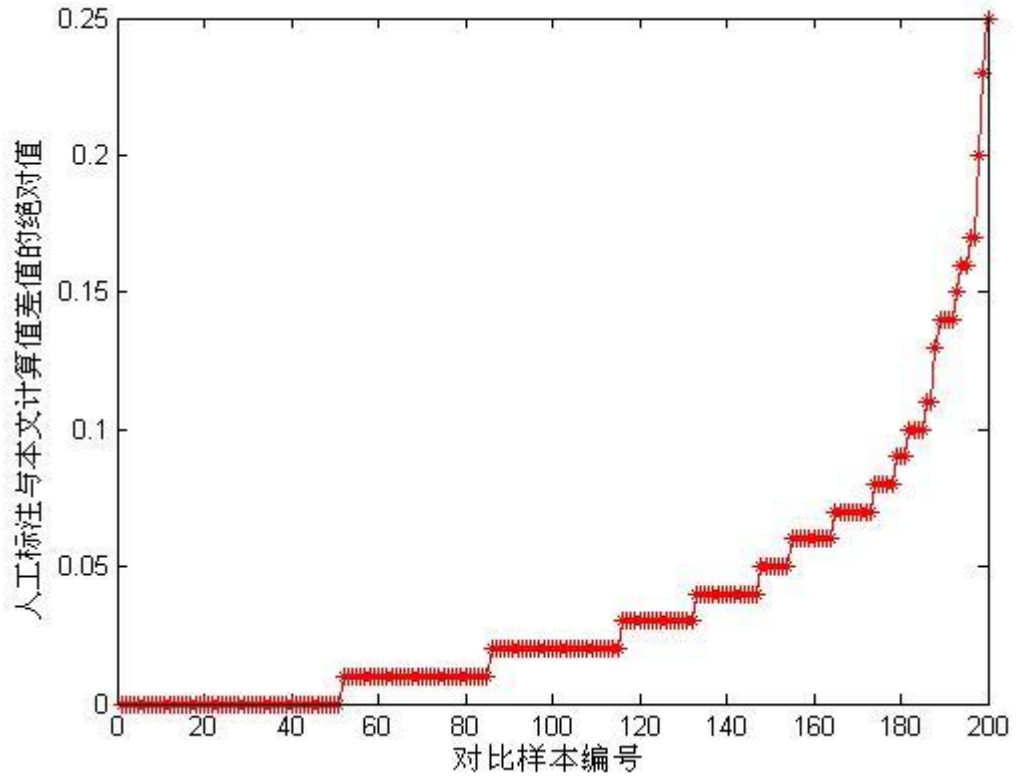

图 6-2　人工标注结果与本文算法的数据对比

由图 6-2，我们可以看出现，大部分的差值集中在 0.1 以下，只有一些零星的孤点值高于 0.2 以上。其原因与网页内容的噪音有关，将在下一节中详细的进行阐述，不过值得欣慰是实验结果基本上与现实中的实体关联度相吻合。

6.3 实验及结果分析

通过上述的实验结果，我们可以看出，本文所介绍的基于实体关联模型的信息融合系统中的各个算法都达到了较为不错的性能，网页抽取算法查全率和召回率都可以达到 94%以上，已经可以满足网页数据抽取的要求，很少会出现抽取错误的情况出现；词条分类算法查全率和召回率都可以达到 80%以上，因为中文处理的复杂性词典分类的结果的精确度相对较低，后期再人工的进行分类后基本上

可以满足系统的使用要求。

在对实体关联模型的测试中，因为没有什么约定的标准进行衡量，本文采用的是完全人工的方法进行，并对人工与本文所实现算法的结果进行比较的方法来实现。从上述的实验中，我们也可以看出实体与文本的相关度以及实体的关联度算法所得出的值与人工进行的标注值偏差也不是很大，基本上都小于 0.1，所以本文所提供的系统结果体现出了优秀的精准度性能。

在实体与文本的相关度以及实体的关联度算法所得出的值与人工进行的标注值的对比中，出现了一些偏差很大的点，主要是出现在文本的噪音上，例如在新浪新闻中，会反复的出现“新浪网记者报道”这些干扰性的噪音，虽然在文本与实体相关度中没有十分突显，但是在实体关联度计算是这些偏差进一步得到了放大，所以就会出现了实体关联度出现一些零星的孤点，其偏差很大。

6.4 本章小结

本章在面向行业的信息融合原型系统的研究、设计与实现的基础上进行了相应的测试实验，并对实验结果进行了分析。主要包括：

首先，介绍了系统实验的数据来源以及测试的指标。

其次，介绍了系统的实验进行及对实验结果展开了分析。包括对本文所使用的网页抽取方法、词条分类方法、文本与实体相关度计算以及实体间关联度计算进行了实验，并给出了实验结果。实验结果证明本文给出的基于实体关联模型的面向行业的信息融合系统可以较好的进行信息融合，表现出了良好的系统性能。

最后，对系统各种算法性能不高的原因展开分析，详细分析了某些因素对实验结果的造成的不利影响，同时还提了了也优化系统性能的大致方法。

第七章 结束语

7.1 论文工作总结

随着信息产业的不断飞速的发展壮大，网络上的数据每天都在以惊人的速度不断的增长，越来越多的用户倾向于在在搜索中查找一些实体或者对象，而不仅限于文档的搜索，我们越来越多的在查询中包含命名实体，例如人名、机构名、地点等。我们总是试图通过围绕实体来构建对我们有意义的查询条件。但是，现在基于文档级别的索引的通用搜索引擎，例如谷歌、百度、雅虎等，已经远远不能满足人们越来越广泛的搜索任务，与此同时，万维网中被人们频繁提及的各种网络实体信息的挖掘与搜索工作才刚刚起步。

信息融合技术是一种综合利用多种信息资源，以获得对某一事物更客观、更本质认识的信息处理技术，主要研究的是如何加工、综合来自于多个信息源的不同信息，并能将这些不同形式的信息进行相互补充、整合，使其信息量达到最大限度的发挥。信息融合技术通过对信息的取舍和集合划分后应用于检索系统，可以更加合理的将查询结果组织起来，使来源于不同信息源的信息可以连接为一个有机的整体，这样可以方便用户查询到更为完整、准确、及时有效而且简洁、明了的信息。

本文主要进行的研究工作如下：首先，本文收集各大网站中某个行业的新闻文本，通过网页抽取技术，整理并构建面向某一行业的中文新闻领域的语料库；其次，研究了上述搜索引擎的不足与用户搜索的习惯的问题，基于百度百科，通过词条的抽取、整理、分类，得到一个基于某一行业领域的词典，通过机器学习的方法构建面向行业的网页信息融合原型系统，以实体为中心对信息进行融合，目的在于利用实体的概念将信息以实体为中心集成起来，更方便于普通互联网用户有效的利用网络资源，通过信息的加权处理，将文本信息融合到网络信息实体上，实现以网络信息实体为中心，基于某个行业领域的信息融合。最后，将文本与实体在处理模块进行计算，在语义理解的基础上将文本中实体进行的相关度进行加权，得到文本中实体的相关度，并根据实体所出现的文本的关系计算出实体间的相关度。

通过实验及实验结果的分析，本文在系统的实验中，使用已经构建好的基于中文新闻领域的语料库作为测试集，对该面向行业的信息融合原型系统进行了测试，实验结果表明，通过与人工标注的实体关联度进行对比，本文所构建的实

体关联模型中，文本与实体的相关度以及实体间的关联度与人工标注的结果偏差小于 0.1，计算结果与人们认知结果基本吻合，具有较高的准确率。

7.2 问题和展望

本文对面向行业的信息融合原型系统进行了较为深入的研究，并在本文中的实现部分给出了相应的算法的实现。但仍然发现许多可以改进与完善的地方，概括为以下几点：

首先，实体抽取算法的完善与优化。本文考虑了信息融合系统会受到分词系统的性能的影响较大从而提出了基于百度百科的词条分类对分词系统进行补充，实体词典大大提高模型的准确度，但仍然存在一词多义等中文现象对结果的影响，这些问题将是今后的工作研究的侧重点。其次，对实体关联模型的研究。本文所介绍的信息融合系统只考虑了将文本以实体作为融合的维度，是一个简单的实体抽取与实体信息之间的计算过程。但中文文本中往往表述了其它的一些相关的信息，所以只将实体作为与文本相关度的计算方式可能会遗漏一些重要的信息，这是系统研究的更进一步的方向。生成多维度的实体融合模型意义重大，它能使中文领域的信息融合更为全面、透彻与立体化。最后，对于文本中的一些噪音的处理方面，算法还不能做到很完美，需要在今后的工作中进一步的研究与完善。

由于本文作者的水平有限，还存在许多问题需要去研究与深入。本文所述难免有不当之处，算法可能还存在一定的缺陷，恳请各位专家、同行给予指出，在此本人向您表示衷心的感谢。

参考文献

[1] Foster I, Kesselman C, Nick J, et al. The physiology of the grid: An open grid services architecture for distributed systems integration. http://www.globus.org/research/papers/ogsa.pdf,2002

[2] UDDI: The UDDI Technical White Paper. http://www.uddi.org, 2000

[3] R.Mihalcea, P.Tarau. Textrank: Bringing Order into Texts. Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing(EMNLP 2004)2004,:404-411.

[4] 高琦.文本挖掘与信息融合技术在检索系统中的应用[J]. 图书馆学刊. 2003(03)

[5] Baidu. http://www.baidu.com

[6] Google. http://www.google.com

[7] INFOMINE: scholarly internet resource collections, [2005-03-09]. http://infomine.uer.edu/.

[8] RSS 2.0 specification. [2005-07-15]. http://blogd.law.harvard.edu/tech/rss.

[9] http://www.newsisfree.com

[10] L.R. Rabiner. A tutorial on hidden markov models and selected applications inspeech recognition. In Proceedings of the IEEE, pages 257–286, 1989

[11] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proc. 18th International Conf. on Machine Learning, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001

[12] G. Salton, M. J. McGill. Introduction to Modern Information Retrieval. McGraw-Hill, New York, 1983.

[13] Y. Hu, G. Xin, R. Song, G. Hu, S. Shi, Y. Cao, and H. Li. Title extraction from bodies of html documents and its application to web page retrieval. In Proc. of SIGIR 2005, pages 250–257, 2005

[14] N. Craswell, D. Hawking, and S. Robertson. Effective site finding using link anchor information. In Proc. of SIGIR 2001, pages 250–257, New Orleans, 2001

[15] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems, 30(1-7):107–117, 1998.

[16] T. Westerveld., W. Kraaij, and D. Hiemstra. Retrieving web pages using content, links, urls and anchors. In Proc. of TREC10, 2002.

[17] G.-R. Xue, H.-J. Zeng, Z. Chen, Y. Yu, W.-Y., W. Xi Ma, and W. Fan. Optimizing web search using web click through data. In Proc. of CIKM 2005, pages 118–126, 2005

[18] http://www.wikipedia.org/

[19] http://baike.baidu.com

[20] G. Salton, A. Wong, and C. S. Yang, A Vector Space Model for Automatic Indexing[J]. Communications of the ACM, 1975,18(11):613–620.

[21] http://htmlcleaner.sourceforge.net

[22] Chang C. Lin C J. LIBSVM: a library for support vector machines. Dept. of

Computer Science and Information Engineering, National Taiwan University 2001.

[23] 刘群,张华平,俞鸿魁等.基于层叠隐马模型的汉语词法分析[J].计算机研究与发展,2004,41(8):1421-1429.

[24] 夏天,樊孝忠,刘林等.利用 JNI 实现 ICTCLAS 系统的 Java 调用[J].计算机应用,2004,24(z2):177-178,182.

[25] 张华平,刘群.基于 N-最短路径方法的中文词语粗分模型[J].中文信息学报,2002,16(5):1-7.

[26] HORSTMANN CS, CORNELL G. Java2 核心技术 卷 2: 高级特性[M]. 北京: 机械工业出版社, 2000.

[27] 刘奕群,马少平,洪涛,刘子正. 搜索引擎技术基础: 清华大学出版社. 2010 年 7 月第 1 版:第七章.

[28] Buckley C. Implementation of the SMART information retrieval system. Cornell University, Tech Rep: TR85-686,1985

[29] Belur V. Dasarathy, ed. (1991). Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques. ISBN 0-8186-8930-7.

[30] [100] 张学工.关于统计学习理论与支持向量机[J].自动化学报,2000,26(1):32-42.

[31] 徐冰,郭绍忠,黄永忠.,基于朴素贝叶斯分类算法的活跃网络结构挖掘[J].计算机应用.2007.6

[32] C.S. Campbell, P. Maglio, A. Cozzi, and B. Dom. Expertise identification using email commu-nications. In Proceedings of ACM 12th Conference on Information and Knowledge Management,pages 528–531, 2003.

[33] M. Hearst. Automatic acquisition of hyponyms from large text corpora. In Proceedings of the 14th International Conference on Computational Linguistics, 1992

[34] E. Charniak and M. Berland. Finding parts in very large corpora. In proceeding of the 37th Annual Meeting of the ACL, 1999

[35] P. Cimino, S. Handschuh, and S. Staab. Towrds the self-annotating web. In Proceedings of World Wide Web (WWW-04), 2004.

[36] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A Popcscu, T. Shaked, S. Soderland, and S. Weld. Web-scale information extraction in knowitall(preliminary results). In Proceedings of WWW 2004,pages 100–110, 2004.

[37] U. Hahn and K. Schnattinger. Towards text knowledge engineering. In Proceedings of the 15th National Conference on Artificial Intelligence and the 10th Conference on Innovative Application of Artificial Intelligence, pages 524–531, 1998

[38] B. Liu and C. Chin. Mining topic-specific concepts and definitions on the web. In Proceedings of WWW 2003, pages 251–260, 2003.

[39] H. Fang, L. Zhou, and C Zhai. Language models for expert finding-uiuc trec 2006 enterprise track experiments. In Proc. of TREC 2006, 2006.

[40] D. Fensel, C. Bussler, Y. Ding, and B. Omelayenko. The web service modeling framework WSMF. In Proceedings of Electronic Commerce Research and Applications, 2002.

[41] http://web-harvest.sourceforge.net/

[42] http://www.dia.uniroma3.it/db/roadRunner/

[43] Zhang Y M, Zhou J F. 2000. A Trainable Method for Extraction Chinese Entity

Names and Their Relations. In Proceeding of the Second Chinese Language Processing Workshop, Hong Kong, 66~72.

[44] 孙承杰, 关毅. 2004. 基于统计的网页正文信息抽取方法的研究. 中文信息学报, 18(5): 17~22.

[45] 李剑波, 李小华, 董树明, 杨科华. 2006. 一种基于 XML 的 Web 信息抽取方法. 情报杂志, 8:49~51.

[46] Y. Zhai, and B. Liu. Web Data Extraction Based on Partial Tree Alignment. In Proceedings of the 14th international conference on World Wide Web, pp. 1614-1628, 2005.

[47] Salton G, Singhal A, Buckley C, et al. Automatic Text Decomposition Using Text Segments and Text Themes[C].In: Proceedings of the seventh ACM Conference on Hypertext. NY: ACM New York,1996:53-65.

[48] 包胜华.基于 Web 的实体信息搜索与挖掘研究[D].上海交通大学,2008.

[49] 陈永超,刘贵全. 一种基于命名实体的搜索结果聚类算法[J]. 计算机工程,2009,07:46-48.

[50] http://www.dia.uniroma3.it/db/roadRunner/

[51] T. Cheng, X. Yan and K. Chang. Supporting entity search: a large-scale prototype search engine[A]. ACM, 2007: 1144-1146

[52] 王治江.面向领域的垂直搜索系统研究与实现[D].大连理工大学,2008.

[53] 刘占山.基于 XML 搜索引擎的研究[D].吉林大学,2007

[54] 刘治华. 面向主题的文档摘要技术研究[D].北方工业大学,2011

[55] 芮璋现,肖海波. 支持向量机(SVM)及其应用[J]. 福建电脑,2007,04:110+192.

[56] 寇月,申德荣,李冬,聂铁铮. 一种基于语义及统计分析的 Deep Web 实体识别机制[J]. 软件学报,2008,02:194-208.

攻读学位期间发表的学术论文

[1] 林哲，吴国仕．一种基于实体模型的信息融合方法的研究[OL]．[2012-12-27]. 中国科技论文在线，http://www.paper.edu.cn/releasepaper/content/201212-957