

清华大学

# 综合论文训练

题目：基于 WEB 的计算机领域新术  
语的自动检测

系 别：计算机科学与技术系

专 业：计算机科学与技术

姓 名：刘知远

指导教师：孙茂松教授

2006 年 6 月 26 日

指导教师评语:

基于Web m 信息域新词语自动检测是词语研究领域的  
的前沿研究,具有重要应用价值(在本文这方面进行了有益的探  
索,取得了较好结果,圆满完成了所交给的研究任务。论文  
写作有特色。

指导教师(签字) 张凤林

评阅教师评语:

刘知远的毕业设计论文抓住基于Web的计算机应用或新词语自动  
检测这一词语研究的前沿课题进行研究,选题具有应用价值。论文设  
计的算法包括预处理、即时新闻文本获取、N元词串统计、候选新词  
语发现和时间序列分析等几个部分,算法比较有效,有新意,取得  
了较好结果。

评阅教师(签字)

陈君平

答辩小组意见:

一致同意通过论文答辩。

组长(签字)

周强

成绩: 89

教学负责人(签字)

王


2006年6月22日



## 关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：学校有权保留学位论文的复印件，允许该论文被查阅和借阅；学校可以公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存该论文。

(涉密的学位论文在解密后应遵守此规定)

签 名：刘知远 导师签名： 日 期：2006.6.20

# 综合论文训练任务书

姓名 刘知远 学号 2002011678 班号 计22  
系别 计算机 同组姓名                      指导教师 孙茂松教授

## 一、课题名称:

基于 WEB 的计算机领域新术语的自动检测

## 二、论文主要内容及进度安排:

**主要内容** 这一技术的主要应用是及时发现外来计算机领域新术语并确定其中文名称,以规范国内的外来术语的命名,主要内容包括:

1. 抓取指定英文 IT 新闻网站历史上的数据,形成计算机领域背景语料库;
2. 定期检测指定 IT 类新闻网站,形成即时语料库;
3. 通过各语料库之间的比对,得到候选新术语;
4. 采用时间序列分析等技术,检测一段时间内候选新术语在每日获取的新闻文本中出现的频度,最终确定新术语。

### 进度安排

- 1~4 周: 调研前人进行的研究; 获取通用语料库, 计算机领域背景语料库, 并开始定期检测指定 IT 网站。
- 5~12 周: 处理语料库, 实现新术语的自动检测。
- 13~14 周: 调整参数, 使算法实现的召回率和准确率达到最优配置。
- 15~16 周: 撰写论文。

### 三、 论文主要要求（如主要指标）：

在新术语的召回率(Recall)保持较高水平的前提下，不断提高系统的准确率(Precision)，使系统达到可用水平。

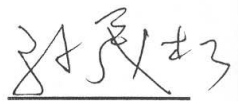
### 四、 主要参考文献：

[1] Thian-Huat Ong and Hsinchun Chen(1999). Updateable PAT-Tree Approach to Chinese Key Phrase Extraction using Mutual Information: A Linguistic Foundation for Knowledge Management, Appeared in Proceedings of the Second Asian Digital Library Conference, November 8-9, 1999, pp. 63-84.

[2] P.D. Turney. Mining the Web for Lexical Knowledge to Improve Keyphrase Extraction: Learning from Labeled and Unlabeled Data. August 13, 2002

[3] Peter D. Turney. Coherent Keyphrase Extraction via Web Mining, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03). August 9-15, 2003. Acapulco, Mexico. pp. 434-439.

指导教师签字



研究所教学负责人签字



2006年 3 月 26 日

## 中文摘要

本文主要介绍“基于 WEB 的计算机领域新术语的自动检测”算法的设计和实现。

随着计算机技术的迅猛发展，英语中每天都会出现大量的该领域的新术语，如何将这些新兴术语及时发现并纳入到汉语中来，是一个迫切而非常有意义的工作。该算法正是基于这一需求而设计实现的。

新术语一般具有以下两个特征：首先新术语是自某一时间点以来首次出现的词汇；另一方面，新术语必须受到普遍的认可，在一定时间内出现频率能够保持较高水平。该算法正是针对新术语的这两个基本特征设计的。其核心思想是，首先通过语料库的比对，找到“自某一时间点”以来在计算机类语料库中新出现的词语，即候选新术语，它们满足新术语的第一个特征；然后通过考查候选新术语在时间上的频度曲线，找到其中被广泛地应用，而非昙花一现的词语，确定为新术语。

该算法包括预处理，即时新闻文本获取，N 元词串统计，候选新术语发现和时间序列分析等几部分。“即时新闻文本获”部分采用 RSS 技术获取每日新闻构成即时新闻语料库，“N 元词串统计”部分统计通用语料库和计算机领域背景语料库的 N 元词串，然后“候选新术语发现”部分在即时新闻语料库中剔除以上已出现过的 N 元词串，得到候选新术语，最后“时间序列分析”部分通过评价函数根据候选新术语的频度曲线最终确定新术语。

该算法实现不是代替用户精确判定哪些是新术语，它的主要功能是为专家提供简短的新术语的名单，其中囊括绝大多数真正的新术语。专家只需要在该名单中检测选取真正的新术语，这样可以大大降低他们发现新术语的寻找范围和工作量。

**关键词：**自然语言处理，新术语，自动检测，统计，N 元词串，RSS





## ABSTRACT

This thesis introduces the algorithm “Web-Based Automatic Detection for IT New Terms” .

With the rapid development of computer science and technology, a large number of new terms in the field are emerging. It is a very meaningful work to detect for these new terms in time and translate them into Chinese. The algorithm is designed based on the urgent needs.

Generally new terms have two characteristics. First, new terms should emerge for the first time since some time. Secondly, new terms should be universally recognized and used widely. The algorithm detect for new terms based on their two characteristics. First of all, we compare different corpus built based on time to find candidates which meet the first feature. Then we adopt time series analysis to check the frequencies of these candidates for final new terms which meet the second feature.

The algorithm is designed to comprise the following modules, Preprocess Module, News Capture Module, Statistical Module, Candidates Detection Module, and Time Series Analysis Module. In “News Capture Module” we adopt RSS technology to capture daily news. And in “Statistical Module” we compute the frequency of N-gram words in General Corpus and Computer Background Corpus. Then the N-gram words are removed from the Just-in-Time Corpus in “Candidates Detection Module” and candidates are sorted out. These candidates are evaluated with some function based on their frequency during some time in “Time Series Analysis Module” . And finally new terms are determined and submitted.

The system is primarily to provide a list of new terms for experts. They can easily check and determine which are real new terms just within the list. The algorithm will significantly reduce their seeking range and workload for new terms.

**Key words:** Natural Language Processing, New Term, Automatic Detection, statistics, N-gram, RSS



# 目 录

中文摘要 .....	I
ABSTRACT .....	III
第一章 引言 .....	1
1.1 背景介绍及需求 .....	1
1.2 前人的工作 .....	2
1.2.1 信息抽取 .....	3
1.2.2 关键词语提取 .....	4
1.2.3 中文新词语发现和中文分词 .....	4
1.3 毕业设计的主要任务 .....	5
第二章 算法思想简介 .....	7
第三章 预处理 .....	11
3.1 生成背景语料库 .....	11
3.1.1 抓取网页 .....	12
3.1.2 提取正文 .....	14
3.2 语料库统计数据 .....	14
第四章 即时新闻文本获取 .....	17
4.1 RSS 介绍 .....	17
4.1.1 背景知识 .....	17
4.1.2 RSS规范 .....	19
4.2 从 RSS 获取新闻网页 .....	24
第五章 N 元词串统计 .....	29
5.1 流程介绍 .....	29
5.2 流程实现 .....	30
5.2.1 划分句子 .....	30
5.2.2 统计数据的存储结构选取 .....	31
5.3 统计结果 .....	32

第六章 候选新术语发现 .....	35
6.1 孤岛词串发现算法设计 .....	35
6.2 存储形式与数据结构 .....	40
6.2.1 XML .....	41
6.2.2 通过 XML 存储候选新术语信息 .....	43
第七章 时间序列分析 .....	47
7.1 时间序列分析的必要性 .....	47
7.2 算法设计 .....	50
7.2.1 聚合数据 .....	50
7.2.2 评价数据 .....	50
7.3 结果查看 .....	51
第八章 实验结果及分析 .....	55
8.1 准确率分析 .....	55
8.2 粒度和分析点数目对结果的影响 .....	58
第九章 结论和展望 .....	61
9.1 结论 .....	61
9.2 存在问题 .....	61
9.2.1 广告问题 .....	61
9.2.2 新闻评论问题 .....	62
9.3 未来工作 .....	63
插图索引 .....	I
表格索引 .....	III
公式索引 .....	V
参考文献 .....	VII
致 谢 .....	XI
声 明 .....	XIII

附录 A 外文资料调研阅读报告 .....	XV
A.1 Information Extraction .....	XV
A.1.1 Definition .....	XV
A.1.2 The Overall Flow .....	XV
A.1.3 Pattern Matching and Structure Building .....	XVI
A.1.4 Lexical Analysis .....	XVI
A.1.5 Name Recognition .....	XVI
A.1.6 Syntactic Structure .....	XVII
A.1.7 Scenario Pattern Matching .....	XVII
A.1.8 Co-reference Analysis .....	XVIII
A.1.9 Inferencing and Event Merging .....	XVIII
A.2 Phrase Extraction .....	XVIII
A.2.1 Dictionary approach .....	XVIII
A.2.2 Linguistic approach .....	XIX
A.2.3 Statistical approach .....	XIX
A.3 Stemming .....	XX
A.4 N-gram and Bi-gram .....	XXI
A.5 TF*IDF Method .....	XXIII
附录 B 实验数据列表 .....	XXV



# 第一章 引言

本章主要介绍“基于 WEB 的计算机领域新术语的自动检测”算法的背景和需求,前人的工作和研究,以及毕业设计的主要任务。

## 1.1 背景介绍及需求

随着社会与科学技术的飞速发展,日常语言中的新词语不断涌现。在英语世界,以收录新词语而闻名的 Barnhart Dictionary of New English,其出版刊物 The Barnhart Dictionary Companion 称,每年可以提供约1,500个新词语和新语义<sup>[1,2]</sup>。在中国,语言文字工作委员会专家曾做的一个保守统计,中国自改革开放以来20年平均每年产生800多个新词语<sup>[3]</sup>。因此新词语在中国也成为研究的热点。

汉语中的新词语,有相当大的比例来自英语。尤其是随着信息技术 (Information Technology) 的飞速发展,英语世界每天都会产生大量的计算机领域特有的新术语,是汉语新词语的重要来源。总的来讲,计算机领域的新术语一般来自新技术、新思想、新产品。根据参考文献[4],英语计算机领域的新词汇主要可以细分为以下几类:

1. 公司名称、产品/设备名称及型号、广告用词,如 HP、DELL、Yahoo!、FreeBSD 等;
2. 计算机领域新技术、新思想用词,如 Web2.0、Blog、RSS 等;
3. 专用动词,如zip/unzip、compile 等;
4. 文件扩展名,如 doc、jpg、pdf 等;
5. 通用标准或协议用词,如 mp3、TCP/IP、HTTP 等;
6. 游戏词汇,如 RPG、Star Craft 等;
7. 网络新词汇,如网络国际顶级域名,网络用语 (lol、btw 等)。

正因为计算机领域的飞速发展以及该领域新术语的大量涌现,我国面临如何及时发现、引入和翻译这些新术语的问题,让汉语和中国的科技发展能够与世界 IT 产业发展保持同步。但是,目前缺少一个权威的机构及时地检测英语中新产生的术语并给予规范的命名,使得当前汉语中外来计算机类新术语的使用纷繁芜杂,这主要表现在两个方面:

- 蹩脚翻译：由于没有经过专家的反复讨论，大量的外来语翻译非常蹩脚。例如“菜单”是计算机领域的“Menu”的中译名，虽然大多数经常接触计算机的人已经习惯这个词语，但是对于更多的刚刚接触计算机的人来讲，这个翻译显然不如专家推荐的“表单”更加合理。
- 多种翻译并存：由于没有权威部门及时推出计算机领域新术语的推荐翻译，往往出现多种翻译并存的现象，让读者混淆不清。例如“Hash Table”，在国内被翻译成“哈希表”，“杂凑表”，“散列表”等等。

由以上的例子可以明显看出，及时发现计算机领域新术语并给出合理翻译的工作，是非常必要的。但这些新出现的术语散落在海量的网络文本中，靠人工去进行挖掘和检测是不可想象的，因此亟需一个可以检测新闻网站并自动发现新术语的算法。而“基于WEB的计算机领域新术语的自动检测”算法正是为满足这一需求而进行的研究。

## 1.2 前人的工作

对于新词的研究，主要集中于语言学界，对于新词的关注，主要是词典编纂界。在英语世界的计算语言学界，尚未发现对于新词发现这一课题的充分研究。在卷帙浩繁的文本中，如何甄别出新词，仅靠专家是无法做到的，因此可以料想那些著名的词典出版商的背后，都会有一些计算机应用软件，帮助他们统计、甄别、遴选新词。在汉语计算语言学界，对于新词语的发现则受到比较多的关注，因为这与汉语分词有着密切联系，直接关系到汉语语言处理的效率。其主要采用的方法不外乎基于统计和基于规则两种，或者这两种方法的综合，[5,6]两篇论文是这两种方法的典范。在“基于WEB的计算机领域新术语的自动检测”的研究中，将主要使用基于统计的方法。

值得一提的是，英国利物浦大学将其研究发现的新词整理成名为“Neologisms in Journalistic Text”的网页（网址为：<http://www.rdues.liv.ac.uk/welcome.html>）。他们从1989-2000年的报纸 *Independent* 和2003年的报纸 *Guardian* 上收集新词语。在收集过程中，他们使用了一种计算机过滤软件将新词识别出来。这个软件源于 APRIL (Analysis and Prediction of Innovation in the Lexicon)，而 APRIL 项目的目的是，开发对 rare words 的半自动分类系统，并在此基础上研究和预测语言结构的发展趋势。从英国利物浦大学整理的新词表来看，全部是新词，而非词组，大多是新造词，少部分是旧词新意。而我们的



研究,则是自动检测计算机领域新词语。也就是说不仅包括新词,还会包括新词组。这是因为在日新月异的信息科学的发展中,新技术的产生总会催生出新的描述词汇,这些词汇并不仅限于新词,有相当的部分是新词组,例如“Data Mining”,“Information Retrieval”,“Machine Translation”,“Natural Language Processing”等词组,它们多是旧词的新组合。因此,不仅新词,而且新词组,都是要求能够检测发现,也是本算法与利物浦大学研究项目的不同之处。

虽然在计算语言学和自然语言处理学科中,并没有对新词语发现的专门研究,但是新词语发现需要的基本技术,却已经非常成熟,在几个领域得到广泛的应用。例如信息抽取(Information Extraction),关键词语提取(Key-Phrase Extraction)等。下面简单回顾各研究领域的发展状况及对新词语自动检测的意义。

### 1.2.1 信息抽取

信息抽取的主要功能是从文本中抽取出特定的事实信息(Factual Information)。比如,从新闻报道中抽取出恐怖事件的详细情况:时间、地点、作案者、受害者、袭击目标、使用的武器等;从经济新闻中抽取出公司发布新产品的情况:公司名、产品名、发布时间、产品性能等;从病人的医疗记录中抽取出症状、诊断记录、检验结果、处方等等。通常,被抽取出来的信息以结构化的形式描述,可以直接存入数据库中,供用户查询以及进一步分析利用。<sup>[7]</sup>

信息抽取可以说是人们需求和当前技术结合的产物<sup>[8]</sup>。目前的自然语言理解技术还不足以真正理解文本的意义,但是人们急需一种手段能够在海量的文本中发现对自己有用的特定信息。因此,信息抽取,这样一个通过特定信息的特征来提取信息的技术,便应运而生。

信息抽取的蓬勃发展与MUC(Message Understanding Conference)的召开密切相关。MUC是一个竞赛性质的会议,共召开过7次。正是MUC使得信息抽取成为自然语言处理的重要分支,并不断推动这一领域的研究发展。目前信息抽取的研究都集中在一些特定的领域。<sup>[9]</sup>

信息抽取的主要技术包括:命名实体的识别,句法分析,篇章分析与推理及知识获取。其中命名实体的识别与该研究课题较为相似。命名实体是文本中基本的信息元素,是正确理解文本的基础。一般来讲,命名实体是指现实世界中的具体的或抽象的实体,如人、组织、公司、地点等,通常用唯一的标志符(专有

名称)表示,如人名、组织名、公司名、地名等。至于命名实体的确切含义,只能根据具体应用来确定。比如,在具体应用中,可能需要把住址、电子信箱地址、电话号码、舰船编号、会议名称等作为命名实体。<sup>[7]</sup>

对于命名实体的识别,主要有基于规则和基于统计两种方式。一般来讲,基于规则的方法优于基于统计的方法。但是这些规则需要人工编制,而且往往依赖于具体的领域、文本格式等,因此比较耗时,并且容易出现错误。而基于统计的方法,在利用人工标注的语料进行训练后,标注语料的时候不需要人工干预,效率比较高,便于移植到新的领域。

### 1.2.2 关键词语提取

在学术期刊的每篇论文正文前,都会提供一个关键词语列表,用来表示这篇论文的主要内容。而关键词语提取,则试图由计算机自动完成提取关键词语列表的过程。在某些论文,这也称为关键词提取(Key-Word Extraction)。不同之处在于 phrase 更准确,因为列表中往往不仅是单词,还有一些由两个词以上组成的词组。在关键词语提取方面,加拿大教授 Peter D. Turney 做了很多有益的工作。

关键词语提取,是指输入一个文本,系统自动产生一个关键词语列表。其主要的的应用包括: Text Summarization, Human-Readable Index, Interactive Query Refinement, Machine-Readable Index 和 Feature Extraction as Preprocessing for Further Machine Analysis<sup>[10]</sup>。

关键词语提取和信息抽取是不同的。信息抽取是提取特定种类的信息。例如给一些关于恐怖活动的新闻报告,可以通过信息抽取获取指定种类的信息,例如恐怖分子的组织名,恐怖活动发生时间等。而关键词语提取不是局限于某一特定任务,通过关键词语提取可以提取任何给定文本的揭示文章主要内容的词语列表。<sup>[11]</sup>

关键词语提取,其最基本的方法,就是基于统计技术的。这可以从直观想象的到,作为文章的主要内容,这些关键词语在文本中一定出现的频率要高于其他普通的词语。但是这样的方法有根本的缺陷,就是文本中频率较高的词语可能存在大量的与文本话题无关的词语。目前的主流方法,是基于统计的机器学习。

### 1.2.3 中文新词语发现和中文分词

中文不像英文那样,每个单词之间都有空格分隔,因此在中文自然语言处

理中，中文分词是一项重要的工作，词性标注，关键词语提取等一系列的工作都是在此基础上进行的。中文分词（Word Segmentation）又称词语提取（Phrase Extraction）。主要的方法包括<sup>[12]</sup>：

- 词典方法：通过人工手段把已知的词编纂为词典，然后使用该词典将文本中在该词典中出现的词语提取出来。这种方法的优点是简单易于实现。但是这种方法对于识别人名、地名等命名实体会受到极大的限制。
- 语言学方法：采用语法或语义知识，通过规则来识别词语。虽然这个方法在英语领域可以取得很高的准确率，但是由于对复杂而庞大的领域很难用规则描述完备，因此这个方法并不实用。
- 统计方法：这种方法通过在巨大语料上得到的统计信息来识别词语。实验证明，这种方法可以得到不错的效果。这种方法经常会与 N-gram 有关。虽然会有一些的计算量，但是这一方法完全不用人工干预。这一方法的主要缺点在于，有些词语在文本中出现次数较少，统计的方法无法分别出来。另外就是对于 N-gram，由于计算量的问题，N 一般只能取2, 3等作为近似。

### 1.3 毕业设计的主要任务

本毕业设计的主要任务是对前人的工作进行调研，利用已有的自然语言处理技术研究设计“基于 WEB 的计算机领域新词语的自动检测”算法，并完成一个基于 WEB 的计算机领域新词语的自动检测系统，其功能主要包括：

- 可以自动检测每天新闻文本中出现的计算机领域新术语，并提供其出现的句子，及其在一段时间内出现的频度变化。
- 用户可以设置语料库、词表等参数。
- 可以查看一段时间内，所有词语的频度统计排名。
- 算法实现与领域无关。更换相关语料库和词表，该算法完全可以对其他领域，诸如生物学、物理学等领域的新词语进行自动检测。

此外，对该算法需要说明两点：

首先，该算法主要是检测在大众生活中流行的计算机领域新术语。这样做的的原因是，在学术论文中出现的术语，一般为专业人士使用，并会附注对应英文术语，或者直接采用英文表示，没有必要进行自动检测并给予推荐翻译；另一方面这些学术论文的术语由于使用范围较小，使用人数有限，也不适合采用统计的

方法进行自动检测。而在大众生活中流行的新术语，由于绝大多数使用者是非计算机专业人士，因此需要及时检测，并给予合适的翻译，防止出现蹩脚的翻译和多种翻译并存的现象，以免造成误解和歧义。

另外，该算法只是对新术语的“预警”，它只是为用户提供一个可能是新术语的词语列表，至于甄别哪些是真正的新术语，哪些不是，仍然需要人工来确定。由于目前自然语言处理技术的局限，该算法只能做到较大程度地减小用户查找新术语的范围，降低用户的工作量，而不能真正的非常准确地检测和判定新术语。也正是出于这种考虑，算法中检测、判定新术语往往采取比较保守的手段，让所有可能是新术语的词串都能加入到该列表，这样可以保证算法的召回率，让用户放心地在列表中甄别新术语即可，而无需担心新术语不被该列表收入。

以下几章将详细介绍“基于 WEB 的计算机领域新术语的自动检测”的算法设计和实现。

## 第二章 算法思想简介

本章将简单介绍“基于 WEB 的计算机领域新词语的自动检测”的算法思想和整体设计。

术语是某个领域的专用词语，它首先是词语。而新术语，首先应该具备新词语的特点。因此要对新术语进行自动检测，首先要明确新词语的涵义和特点。一般认为，新词语包括新词和新语。对于英语来讲，新词是单词，而新语是词组或短语。那么，怎样的单词或词组才能算作新词语？这在不同的情况下有不同的定义，但至少应当满足以下特点：

1. 新词语是指通过各种途径产生的、具有基本词汇所没有的新形式、新意义或新用法的词语<sup>[13]</sup>；
2. 从时间参照角度来说，新词语是“出现在某一段时间内或自某一时间点以来所首次出现的词汇”<sup>[14]</sup>；
3. 新词语必须受到普遍的认可，被广泛地应用，在语言词汇中站稳了脚跟，而非昙花一现<sup>[15]</sup>。

其中第 1 条是新词语的涵义，而第 2、3 条是新词语的两个基本特点。“基于 WEB 的计算机领域新词语的自动检测”算法就是针对新术语/新词语的这两个基本特征设计。其核心思想是，首先通过语料库的比对，找到“自某一时间点”以来在计算机类语料库中新出现的词语，即候选新术语，它们满足新词语的第 2 条特征；然后通过考查候选新术语在时间上的频度曲线，找到其中“被广泛地应用，而非昙花一现”的词语，确定为新术语。

该算法中，这两个核心思想分别是通过“候选新术语发现”和“时间序列分析”两部分完成的。要完成这两部分，还需要三个语料库作为基础。它们分别是：

- **通用语料库** 是指由不特定于某一领域的文本组成的语料库，其中的词语为人们所常用。我们将采用“英文世界名著1000部”作为通用语料库。
- **计算机领域背景语料库** 进行新术语的检测，需要确定一个时间点，作为考查术语是否为新出现的标准。而计算机领域背景语料库，就是指计算机领域中，该时间点之前出现文本所组成的语料库。通俗地讲，就是“旧”的文本组成的语料库。所谓“背景”，意指新术语是与该语料库进行比对之后，

才被突出出来的。接下来如果不作特殊说明，将简写为“背景语料库”。

- **计算机领域即时新闻语料库** 是指在新术语检测时间点之后的新闻文本组成的语料库。该语料库是以时间为单位存放新闻文本的。该语料库是由定时(如每隔3个小时)从英文 IT 网站获取的新闻文本组成的。需要说明的是，这里所谓的“新闻”，绝大部分是狭义上这些 IT 网站发布的新闻，也包括广义上这些 IT 网站的 Blog、论坛里面网友发表的较受关注的文章。接下来如果不做特殊说明，将简写为“即时新闻语料库”。

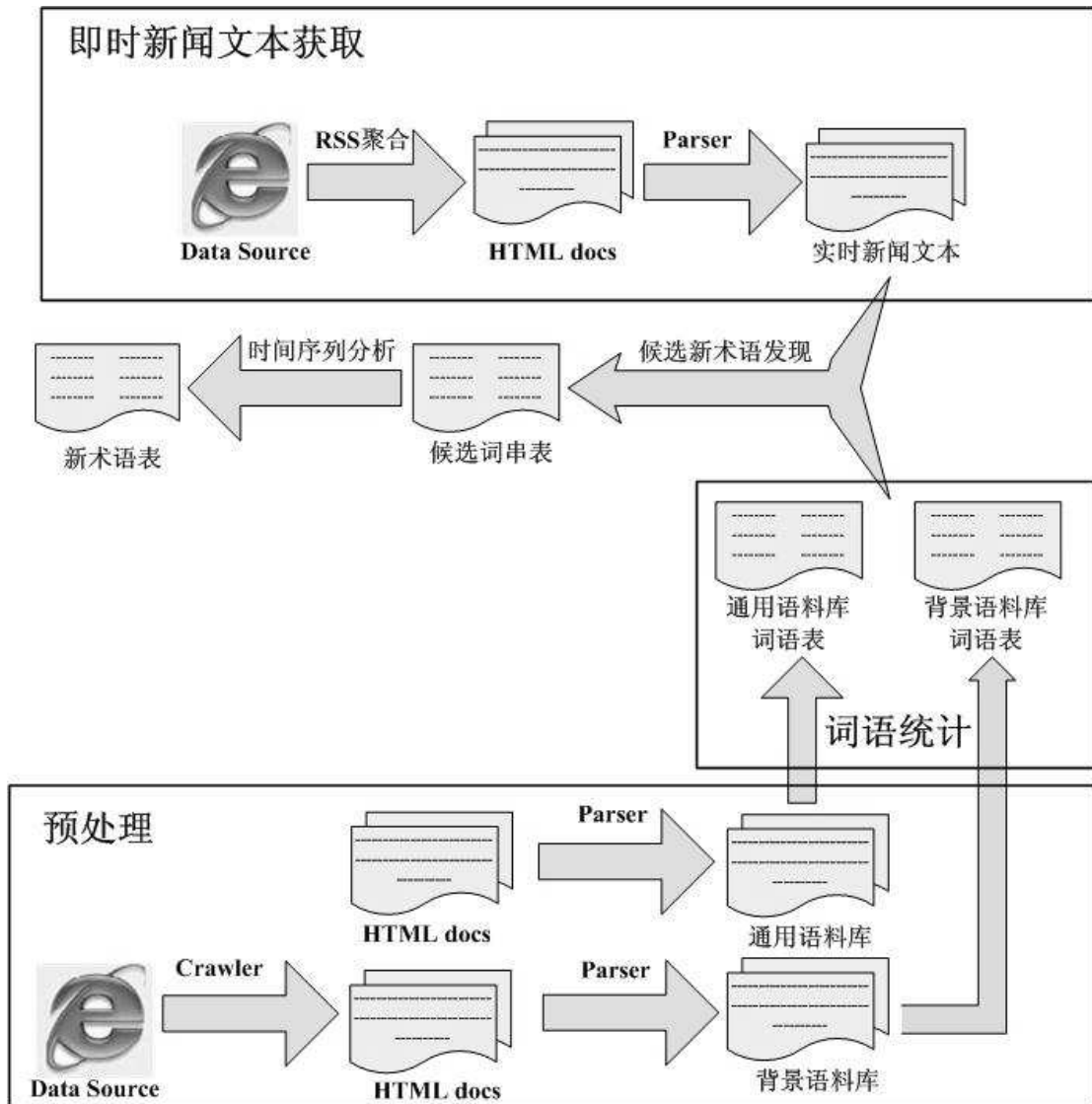


图 2.1 基于 WEB 的计算机领域新术语自动检测算法流程图

在以上三个语料库的基础上，算法流程进行如下。图 2.1 为该算法流程图。

1. 首先, 进行 **N 元词串统计**。统计通用语料库和背景语料库中单词 (一元词串) 和二元词串, 并按照其频度进行排序, 得到单词表和二元词串表。通过统计 N 元词串的方式提取语料库的特征, 这是对语料库的最基本的应用方法之一。
2. 其次, 进行**候选新术语发现**。其主要思路是: 假设通用语料库、背景语料库和即时新闻语料库都是词语的集合, 分别设  $G, B, J$ , 设第一次出现词语的集合为  $N$ , 可以认为:

$$J \subseteq G \cup B \cup N$$

即

$$N = J - (G \cup B)$$

从直观上来讲, 就是即时新闻所使用的词语, 主要来源外乎三个: 通用语料库, 即非领域相关的词语; 背景语料库, 即之前已经出现的计算机领域的术语; 此外就是第一次出现的词语。因此, 这一步骤主要工作就是将即时新闻语料库中新闻文本中, 在通用语料库、背景语料库中已经出现的词语去除, 所得到的剩下的词串即候选新术语。

3. 最后, 进行**时间序列分析**。对于候选新术语, 只考查它们是否具有新词语的一个特点: “在某个时间点之后出现”, 但要被认定为新词语, 还需要满足另外一个特点: “新词语必须受到普遍的认可, 被广泛地应用, 在语言词汇中站稳了脚跟, 而非昙花一现”。自然而然, 就需要“时间序列分析”这一部分来完成考查候选新术语是否具备这一特点, 最终确定哪些候选新术语是新术语。这一部分, 将主要考查候选新术语在一段时间内的频度曲线, 通过评价函数完成对候选新术语的判定。

以下几章详细介绍算法各部分的详细设计和实现。





## 第三章 预处理

预处理的主要工作是：获取通用语料库和背景语料库的原始数据；然后对两部分的原始数据进行解析，获取正文文本，构成通用语料库和背景语料库。

通用语料库的原始数据采用英美小说文本《英文世界名著1000部》的 English Literature 部分。背景语料库的原始数据采用从五大英文 IT 网站抓取的网页。下面主要介绍如何生成背景语料库。

### 3.1 生成背景语料库

背景语料库的原始数据来源于以下五个英文 IT 类网站。其中最后一列“Alexa 排名”显示了该网站在全球网站中访问量排名。“新闻类单项排名”显示这些网站在 IT 类新闻网站或计算机类新闻网站的排名，这是因为 Alexa 对“Information Technology”和“Computers”进行了区分，而在本研究中将不对此作明确区分。Alexa (www.alexa.com) 是一家专门发布网站世界排名的网站，它是当前拥有 URL 数量最庞大，排名信息发布最详尽的网站。Alexa 每天在网上搜集超过1,000GB的信息，然后进行整合发布，现在它搜集的 URL 数量已经超过了 Google。表格中的排名是2006年2月21日的数据。可以看到，这五个网站是世界上影响力最大的五大 IT 类新闻网站。

表 3.1 五大英文 IT 新闻网站列表

网站名	URL	Alexa 排名	(IT/计算机) 新闻类单项排名
CNet News	www.news.com	54	1 (IT 新闻)
ZDNet	news.zdnet.com	658	2 (IT 新闻)
Slashdot	slashdot.org	287	1 (计算机新闻)
PC World	www.pcworld.com	571	2 (计算机新闻)
PC Magazine	www.pcmag.com	571	3 (计算机新闻)

如算法流程图 2.1 所示，对于背景语料库原始数据的预处理包括两个过程：

- 网络爬虫 (Crawler) 程序抓取网页数据：实现从英文五大 IT 网站获取历史网页，组成背景语料库原始数据集。

- HTML Parser 工具解析网页：实现从原始数据集的网页中提取正文文本，组成背景语料库。

下面分别介绍两个过程的具体实现。

### 3.1.1 抓取网页

网页抓取工作是采用离线浏览软件完成的。在本毕业设计中，共测试三款离线浏览软件：

- WebZip: 界面美观，但是下载速度较慢，供设置的参数较少。
- Teleport: 老牌的离线浏览软件，界面朴素，速度较快，但是供设置的参数较少。
- Offline Explorer: 界面朴素，但是下载速度较快，据称是离线浏览软件中下载速度最快的，而且供设置的参数齐全，可以很好地控制下载网页文件属性。非常适合大量数据的下载。

由于 Offline Explorer 的优异特性，最终决定在毕业设计中使用它来抓取网页。在下载的过程，根据各网站的情况，不断调整 Offline Explorer 的参数，从而屏蔽重复网页和垃圾网页。这涉及到动态网页的问题。

所谓动态网页一般指的是采用 ASP, PHP, JSP, CGI 等程序动态生成的页面，该网页中的大部分内容来自与网站相连的数据库，在网络空间中并不存在这个页面，只有接到用户的访问要求后才生成并传输到用户的浏览器中。在动态页的 URL 中包含了问号 (?) 和百分号 (%)。还有一些符号诸如 &, %, + 和 \$ 等在一个动态页的 URL 中也经常能看到。如 URL `http://www.americanbooks.com/cgi-bin/items.cgi?name=naturaldiet`。这样的参数被称作“环境变量”(query string)。这些参数的组合指明产生什么样的网页或者导向哪个静态网页。

搜索引擎的 Spider 不愿意读取放在 cgi-bin 目录下的网页，或是 URL 中包含了符号“?”的字符。其原因就在于，如果在 CGI 中提供了“无穷”数量的 URL，那么 Spider 往往就会因为对这些“无穷”网页的检索而被牢牢套住，陷入死循环。这就是所谓的蜘蛛陷阱(spider traps)。数据库程序对 Spider 亦有可能创建一个与此类似的情形。为避开可能的陷阱，Spider 对于那些带有符号“?”的 URL 中的“?”之后的字符一概不予读取。所以在下载 IT 网站历史网页过程中，需要注意这样的蜘蛛陷阱问题，但也不能仅仅读取“?”之前的 URL，因为有些网站的不同动态 URL 是对应含有正文的不同动态网页的。

例如通过分析下载网页，会发现大量重复的无用网页（如错误提示网页，登陆网页等），其主要原因就是网站许多网页中，含有动态网页 URL，这些 URL 的环境变量指向同一个静态网页，或者会生成相同的动态网页，如错误提示网页，登陆界面网页等。网络爬虫程序得到这些动态网页 URL 之后，就会将同一个网页重复下载，如果不进行相应的设置，就会导致下载数据中充斥着大量重复而无用的网页。以下是对各网站结构特点的总结，其中具体说明了对各网站的特殊设置以防止类似现象发生。

- **Slashdot:** 该网站的组织形式是最方便进行网页抓取的，所有网页都是 HTML 格式，结构简单，信息量较大。该网站通过 xxx.slashdot.org 来进行不同栏目的区别。
- **CNet:** 类似于 Slashdot 的组织形式，该网站的栏目如 news，是在 news.com.com 下，reviews 是在 reviews.cnet.com 下。但是由于该网站使用了动态网页，给网页抓取带来一定的困难，经常会抓取到多张重复页面，即虽然是不同的 URL，但 URL 的环境变量指向同一张静态网页或生成相同的动态网页。在 news.com.com 下，不同的动态 URL 环境变量会指向相同的登陆界面或者错误提示界面，因此只抓取.html 静态网页文件，忽略动态网页；而在 reviews.cnet.com 下，不同的动态 URL 其环境变量参数会产生不同的页面，需要将这些动态网页全部抓取。
- **ZDNet:** 类似于 Slashdot，CNet 的架构，其不同的版面由 xxx.zdnet.com 表示出来。在这里，主要抓取 news 和 blogs 两个版面的数据。ZDNet 面临和 CNet 相似的动态网页处理问题，但与 CNet 不同的是，这里动态 URL 生成不同的有用动态网页，不是指向静态网页。因此无法像 CNet 的 news.com.com 下那样只抓取.html 静态网页文件，需要将所有的 URL 指向的文件都抓取下来。
- **PC Magazine:** 该网站的组织形式比较特别，对网页按照其功能进行分类，如文章在 www.pcmag.com/article2/xxx.xxx 下，而指向各网页的导航网页，则在 www.pcmag.com/category2/xxx.xxx 下。但是无法实现只抓取/article2/下面的文章页面，因为这些页面都需要通过导航网页进入。所以，对该网站的抓取策略是，设置其抓取数据量的上限。
- **PC World:** 该网站的架构是通过 www.pcworld.com 域名之后的一级子目录划分板块的，在这里，主要抓取/news/，/reviews/和/digitalduo/这三个板块的

内容。

### 3.1.2 提取正文

对网页的正文提取工作采用开源软件 HTMLParser 完成。该软件是从著名开源软件发布网站 Sourceforge (<http://sourceforge.net>) 下载的。该软件的主页是 <http://htmlparser.sourceforge.net>。本实现使用的版本是 Version 1.6。

HTMLParser 是一个用 Java (J2SE) 实现的即时 HTML 语法分析程序, 利用它可以很容易的对网页的内容进行分析、过滤和提取。该软件提供接口, 可以提取 HTML 文件的正文、标题、链接、文本、图片等内容。该软件还提供 StringExtractor 类的公共接口, 不仅实现对 HTML 文件的语法分析, 还提实现对 JSP、ASP 等含有函数体的文件进行内容分析、过滤和提取。

在实验中, 主要调用 HTMLParser 的 StringExtractor 类, 实现对 HTML, ASP, JSP 等文件的文本内容提取。

这里存在一个问题, 每个网页除了正文, 还会有大量的网站标题、导航栏、广告等非正文内容。这在一定程度上会影响下一步 N 元词串统计对语料库的特征提取的正确程度。不过由于在“即时新闻文本获取”部分获取的即时新闻文本中也会有相同的非正文内容, 因此, 在一定程度上抵消这种问题产生的影响。

## 3.2 语料库统计数据

通用语料库的原始数据采用英美小说文本《英文世界名著1000部》的 English Literature 部分。该原始数据文件大小为 288MB, 占用磁盘空间为 296MB, 共有 7837 个文件。其中部分文件是被简单的 HTML 的 Tag 封装过的, 所以采用 HTML Parser 进行了解析, 所得到的正文文本和原始数据大小变化不大。对HTML文件的解析方法与上述第 3.1.2 节相似。

通过网络爬虫 (Crawler) 程序, 在 2006 年 3 月 2 日到 2006 年 3 月 7 日的时间内, 完成对五个 IT 网站的历史网页的抓取。获得背景语料库的原始数据量如表 3.2 所示。

表 3.2 IT 网站抓取历史网页数据量统计表

网站名	文件大小	占用空间	文件数
Slashdot	62.9MB	77.1MB	1384
ZDNet	901MB	1.27GB	39966
CNet News	6.14GB	7.07GB	107898
PC Magazine	4.32GB	4.87GB	63427
PC World	2.14GB	2.48GB	40556
总计	13.5GB	15.7GB	253227

经过 HTML Parser 工具，获取纯文本数据量如表 3.2 所示。

表 3.3 IT 网站历史网页获取纯文本数据量统计表

网站名	文件大小	占用空间	文件数
Slashdot	21.6MB	29.1MB	1384
ZDNet	181MB	362MB	39966
CNet News	612MB	1.02GB	107898
PC Magazine	401MB	675MB	63427
PC World	191MB	344MB	40556
总计	1.37GB	2.40GB	253227



## 第四章 即时新闻文本获取

如算法流程图 2.1 所示，算法的“即时新闻文本获取”部分用来生成“即时新闻语料库”。该部分主要包括以下两个步骤，其中第2步如前述第 3.1.2 节：

1. 定期（如每隔3个小时）从五大 IT 新闻网站发布的新闻 RSS 源获取新闻网页；
2. 对获取的新闻网页，通过 HTML Parser 提取新闻文本。所有新闻文本以天为单位存储。

详细介绍第 1 步之前，首先介绍一下 RSS 的相关知识。

### 4.1 RSS 介绍

#### 4.1.1 背景知识

该小节内容主要摘自参考文献 [16–18]。

RSS 是一种描述同步网站内容的格式，是目前使用最广泛的 XML 应用。RSS 搭建了信息迅速传播的一个技术平台，使得每个人都成为潜在的信息提供者。发布一个 RSS 文件（该 XML 文件又称 RSS feed，即“RSS 订阅源”）后，这个 RSS Feed 中包含的信息就能直接被其他站点或用户调用。如图所示为 IBM alphaWorks 所提供 RSS feeds 的截图：

Name	File Access	Description
All	XML	All new and updated technologies from alphaWorks
Application Development	XML	New and updated App Dev technologies
Autonomic Computing	XML	New and updated Autonomic Computing technologies
Collaboration	XML	New and updated Collaboration technologies
Eclipse	XML	New and updated Eclipse technologies
Grid	XML	New and updated Grid technologies
Java	XML	New and updated Java technologies
Security	XML	New and updated Security technologies
Systems Management	XML	New and updated Sys Mgmt technologies
Web Services	XML	New and updated Web Services technologies
Wireless	XML	New and updated Wireless technologies
XML	XML	New and updated XML technologies

图 4.1 IBM alphaWorks 提供 RSS feeds 的图示。该图引用自参考文献 [16]。

RSS 可以是以下三个解释的其中一个：

- Really Simple Syndication
- RDF (Resource Description Framework) Site Summary
- Rich Site Summary

这三个术语都是指的同一种 Syndication 的技术。

从 RSS 阅读者的角度来看，它是一种方便的信息获取工具。RSS 获取信息的模式与加入邮件列表（如电子杂志和新闻邮件）获取信息有一定的相似之处，即可以不必登录各个提供信息的网站而通过客户端浏览方式（称为“RSS 阅读器”）或者在线 RSS 阅读方式这些内容。如下图所示：



图 4.2 RSS 阅读器图示，左侧栏中为 RSS feed 及 feed 中的新闻标题；右侧栏是点击某标题后得到的新闻正文。RSS 在线阅读的网页布局与此类似。该图引用自参考文献 [16]。

RSS 目前已经广泛用于 blog<sup>①</sup>、wiki<sup>②</sup> 和网上新闻频道，世界多数知名新

- ① Blog 或 Weblog，中文称网志或者博客，是一种网上共享空间，以日记的形式在网络上发表自己的个人内容的一种形式。Weblog 是 Web Log 的缩写，中文意思是“网络日志”，后来缩写为 Blog，而 Blogger（博客）则是写 blog 的人。一个博客（blog）就是一个网页，它通常是由简短且经常更新的 Post 所构成；这些张贴的文章都按照年份和日期排列。博客是继 Email、BBS、ICQ 之后出现的第四种网络交流方式，于1997年由 Jorn Barger 所提出。在2004年以后风靡网络世界，被看作是 Web 2.0 的标志。[19,20]
- ② Wiki（维基）一词来源于夏威夷语的“wee kee”，原本是“快点”的意思。在这里 Wiki 指的是一种网上共同协作的超文本系统，可由多人共同对网站内容进行维护和更新。Wiki 系统创造者的沃德·坎宁安（Ward Cunningham）为 Wiki 下的定义是：一群相互连接并可自由扩展的网页、一套用来储存与修改信息的超文字系统，所有的网页储存在一套数据库中，任何人透过具有表单功能的浏览器用户程序，皆可轻易加以编辑。目前根据这一理念创建和维护的网上百科全书 Wikipedia (<http://wikimedia.org>) 收



闻社、网站都提供 RSS 订阅支持。在过去几年中，RSS 在普及性方面有了惊人的增长。Syndic8.com 维护了一个 RSS 频道索引，它的提要列表在两年中加长了大约1400%。Yahoo 新闻、BBC、Slashdot、LockerGnome、Amazon、CNN、Wired、Rolling Stone 和 Apple Computer 都位于许多最普及的 RSS 提要来源之列。

RSS 之所以有上述三种命名，需要从它的发展历史说起。最初的0.90版本 RSS 是由 Netscape 公司设计的，目的是用来建立一个整合了各主要新闻站点内容的门户，但是0.90版本的 RSS 规范过于复杂，而一个简化的 RSS 0.91 版本也随着 Netscape 公司对该项目的放弃而于2000年暂停。不久，一家专门从事博客写作软件开发的公司 UserLand (<http://www.userland.com/>) 接手了 RSS 0.91 版本的发展，并把它作为其博客写作软件的基础功能之一继续开发，逐步推出了0.92、0.93和0.94版本。随着网络博客的流行，RSS 作为一种基本的功能也被越来越多的网站和博客软件支持。在 UserLand 公司接手并不断开发RSS的同时，很多的专业人士认识到需要通过一个第三方、非商业的组织，把 RSS 发展成为一个通用的规范，并进一步标准化。于是2001年一个联合小组在0.90版本 RSS 的开发原则下，以 W3C 新一代的语义网技术 RDF (Resource Description Framework) 为基础，对 RSS 进行了重新定义，发布 RSS 1.0，并将 RSS 定义为“RDF Site Summary”。但是这项工作没有与 UserLand 公司进行有效的沟通，UserLand 公司也不承认 RSS 1.0的有效性，并坚持按照自己的设想进一步开发出 RSS 的后续版本，到2002年9月发布了最新版本 RSS 2.0，UserLand 公司将 RSS 定义为“Really Simple Syndication”。

目前 RSS 已经分化为 RSS 0.9x/2.0 和 RSS 1.0 两个阵营，由于分歧的存在和 RSS 0.9x/2.0 的广泛应用现状，RSS 1.0 还没有成为标准化组织的真正标准。尽管 RSS 在规范上存在众多分歧，但随着越来越多的站点对 RSS 的支持，RSS 已经成为目前最成功的XML应用。

#### 4.1.2 RSS规范

该小节以 RSS 1.0 与 RSS 2.0 作为范例介绍 RSS 规范，内容主要摘自参考文献[16,22–24]。

---

录词条数仅英文已经超过110万条（2006年5月14日数据），远远超过了老牌的大英百科全书（约8万条）。<sup>[21]</sup>

---

RSS 2.0 文件由一个<channel>元素及其子元素组成。除了频道内容本身之外，<channel>还以项的形式包含表示频道元数据的元素——比如<title>、<link>和<description>。项通常是频道的主要部分，包含经常变化的内容。

频道一般有三个元素，提供关于频道本身的信息：

- <title>：频道或提要的名称。
- <link>：与该频道关联的 Web 站点或者站点区域的 URL。
- <description>：简要介绍该频道是做什么的。

项通常是提要中最重要的部分。每个项都可以关于某个 blog、完整文档、电影评论、分类广告或者任何希望与频道连锁的内容的记录。频道中的其他元素可能不变，但项经常发生变化。

新闻项通常包含三个元素：

- <title>：这是项的名称，在标准应用中被转换成 HTML 中的标题。
- <link>：这是该项的 URL。
- <description>：通常作为 link 中所指向的 URL 的摘要或者补充。

所有的元素都是可选的，但是一个项至少要么包含一个<title>，要么包含一个<description>。

项还有其他一些可选的元素：

- <author>：作者的 e-mail 地址。
- <category>：支持有组织的记录。
- <comments>：关于项的注释页的 URL。
- <enclosure>：支持和该项有关的媒体对象。
- <guid>：唯一与该项联系在一起的永久性链接。
- <pubDate>：该项是什么时候发布的。
- <source>：该项来自哪个 RSS 频道，当把项聚合在一起时非常有用。

以下是一个 RSS 2.0 文件的示例。

Listing 4.1 RSS 2.0 文件示例代码，摘自参考文献[16]

```
1 <rss version="2.0">
2   <channel>
3     <title>The channel's name goes here</title>
4     <link>http://www.urllofthechannel.com/</link>
```

```

5 <description>This channel is an example channel for an article.
6 </description>
7 <language>en-us</language>
8 <image>
9   <title>The image title goes here</title>
10  <url>http://www.urlofthechannel.com/images/logo.gif</url>
11  <link>http://www.urlofthechannel.com/</link>
12 </image>
13 <item>
14   <title>The Future of content</title>
15   <link>http://www.itworld.com/nl/ecom_in_act/11122003/</link>
16   <description> The issue of people distributing and reusing
17     digital media is a problem for many businesses. It may also be
18     a hidden opportunity. Just as open source licensing has opened
19     up new possibilities in the world of technology, it promises to do
20     the same in the area of creative content.
21   </description>
22 </item>
23 <item>
24   <title>Online Music Services - Better than free?</title>
25   <link>http://www.itworld.com/nl/ecom_in_act/08202003/</link>
26   <description>More people than ever are downloading music from
27     the Internet. Many use person-to-person file sharing programs like
28     Kazaa to share and download music in MP3 format, paying nothing.
29     This has made it difficult for companies to setup online music
30     businesses. How can companies compete against free?
31   </description>
32 </item>
33 </channel>
34 </rss>

```

该 RSS 文件有两个项 (Element)，即 13 ~ 22 行和从 23 ~ 32 行。

而 RSS 1.0 规范与 RSS 2.0 的最显著不同是 RSS 1.0 采用的 Dublin Core Module。Dublin Core 是由 The Dublin Core Metadata Initiative (DCMI: <http://dublincore.org/index.shtml>) 编制的常见元数据集。该数据集 (Dublin Core Metadata Element Set, version 1.1) 已经被包括 ISO 在内的组织进行了标准化<sup>①</sup>。

其他不同之处还包括，RSS 1.0 所有内容包含在 <rdf:RDF> ... </rdf:RDF> 之间，等等。以下是一个 RSS 1.0 文件的示例。需要注意其发布日期是用 <dc:date> ... </dc:date> 表示。

<sup>①</sup> ISO Standard 15836-2003 (February 2003): <http://www.niso.org/international/SC4/n515.pdf>

Listing 4.2 RSS 1.0 文件示例代码, 摘自参考文献[24]

```
1 <?xml version="1.0"?>
2 <rdf:RDF
3   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4   xmlns="http://purl.org/rss/1.0/"
5   xmlns:dc="http://purl.org/dc/elements/1.1/"
6 >
7   <channel rdf:about="http://example.com/news.rss">
8     <title>Example Channel</title>
9     <link>http://example.com/</link>
10    <description>My example channel</description>
11    <items>
12      <rdf:Seq>
13        <rdf:li resource="http://example.com/2002/09/01/" />
14        <rdf:li resource="http://example.com/2002/09/02/" />
15      </rdf:Seq>
16    </items>
17  </channel>
18  <item rdf:about="http://example.com/2002/09/01/">
19    <title>News for September the First</title>
20    <link>http://example.com/2002/09/01/</link>
21    <description>other things happened today</description>
22    <dc:date>2002-09-01</dc:date>
23  </item>
24  <item rdf:about="http://example.com/2002/09/02/">
25    <title>News for September the Second</title>
26    <link>http://example.com/2002/09/02/</link>
27    <dc:date>2002-09-02</dc:date>
28  </item>
29 </rdf:RDF>
```

在实验中, 我们将主要用到的项的元素为: `<title>`, `<link>`, RSS 2.0 的`<pubDate>`和 RSS 1.0 的`<dc:date>`。其中`<title>`作为新闻文件名; `<link>`用来获取新闻网页内容; `<pubDate>`或`<dc:date>`用来指明该新闻网页应当存放在哪一天的目录下。

需要说明的是, RSS 2.0 的`<pubDate>`中的时间格式和 RSS 1.0 的`<dc:date>`格式互不相同。RSS 2.0 的时间格式是采用 RFC 822 标准, 而 RSS 1.0 则采用 ISO 8601 标准。

其中 RFC 822 标准关于时间的格式规范为:

Listing 4.3 RFC 822 标准关于时间的格式规范, 摘自参考文献[25]

```

1
2   date-time = [ day "," ] date time ; dd mm yy
3                                     ; hh:mm:ss zzz
4
5   day       = "Mon" / "Tue" / "Wed" / "Thu"
6             / "Fri" / "Sat" / "Sun"
7
8   date      = 1*2DIGIT month 2DIGIT ; day month year
9                                     ; e.g. 20 Jun 82
10
11  month     = "Jan" / "Feb" / "Mar" / "Apr"
12           / "May" / "Jun" / "Jul" / "Aug"
13           / "Sep" / "Oct" / "Nov" / "Dec"
14
15  time      = hour zone ; ANSI and Military
16
17  hour      = 2DIGIT ":" 2DIGIT [":" 2DIGIT]
18           ; 00:00:00 - 23:59:59
19
20  zone      = "UT" / "GMT" ; Universal Time
21           ; North American : UT
22           / "EST" / "EDT" ; Eastern: - 5/ - 4
23           / "CST" / "CDT" ; Central: - 6/ - 5
24           / "MST" / "MDT" ; Mountain: - 7/ - 6
25           / "PST" / "PDT" ; Pacific: - 8/ - 7
26           / 1ALPHA ; Military: Z = UT;
27           ; A:-1; (J not used)
28           ; M:-12; N:+1; Y:+12
29           / ( ("+" / "-") 4DIGIT ) ; Local differential
30           ; hours+min. (HHMM)

```

ISO 8601 标准关于时间的格式规范为:

Listing 4.4 ISO 8601 标准关于时间的格式规范, 摘自参考文献[26]

```

1   Year:
2     YYYY (eg 1997)
3   Year and month:
4     YYYY-MM (eg 1997-07)
5   Complete date:
6     YYYY-MM-DD (eg 1997-07-16)
7   Complete date plus hours and minutes:
8     YYYY-MM-DDThh:mmTZD (eg 1997-07-16T19:20+01:00)
9   Complete date plus hours, minutes and seconds:
10    YYYY-MM-DDThh:mm:ssTZD (eg 1997-07-16T19:20:30+01:00)

```

```

11 Complete date plus hours, minutes, seconds and a decimal fraction of a
    second
12 YYYY-MM-DDThh:mm:ss.sTZD (eg 1997-07-16T19:20:30.45+01:00)
13
14 where:
15 YYYY = four-digit year
16 MM = two-digit month (01=January, etc.)
17 DD = two-digit day of month (01 through 31)
18 hh = two digits of hour (00 through 23) (am/pm NOT allowed)
19 mm = two digits of minute (00 through 59)
20 ss = two digits of second (00 through 59)
21 s = one or more digits representing a decimal fraction of a second
22 TZD = time zone designator (Z or +hh:mm or -hh:mm)

```

从直观来看，RSS 2.0 与 RSS 1.0 的时间格式区别举例如表 4.1.2 所示。

表 4.1 RSS 2.0 与 RSS 1.0 时间格式比较

时间	RSS 2.0(pubDate)	RSS 1.0(dc:date)
格林威治时间 2003年9月7日0时0分1秒	07 Sep 2002 00:00:01 GMT	2002-09-07T00:00:01Z
美国东部时间 1994年11月5日8时15分30秒	05 Nov 1994 08:15:30 EST	1994-11-05T08:15:30-05:00

## 4.2 从 RSS 获取新闻网页

上面提到，世界的各大网站都已经提供 RSS 新闻发布服务。而本研究监控的五大英文IT网站都提供了丰富的 RSS Feeds，涵盖了网站各栏目。因此在考虑如何监控这些网站的每日发布新闻的时候，想到采用获取其 RSS Feeds 的方式。这样的优点是：RSS Feeds 发布的新闻提供了格式良好的信息，包括标题、链接和发布时间，可以准确、及时、方便地增量式地对每日发布的新闻进行获取和存储。

RSS 服务大行其道，对于 RSS 解析的开源软件也非常多。在参考文献[27]中提供了对三个 RSS Lib (Rome<sup>①</sup>，RSSUtilities<sup>②</sup>，RSSLib4J<sup>③</sup>) 的评测。对其中的 RSSUtilities，RSSLib4J 进行实验，发现 RSSUtilities 在解析 Slashdot 的 RSS

① <https://rome.dev.java.net/>

② [http://java.sun.com/developer/technicalArticles/javaserverpages/rss\\_utilities/](http://java.sun.com/developer/technicalArticles/javaserverpages/rss_utilities/)

③ <http://rsslib4j.sourceforge.net>

文件时候会出现错误，导致程序崩溃，无法输出正常结果。而且该工具没有提供源代码，现有的可执行二进制 class 文件在运行过程中会把所有解析的 tag 标签都输出来，导致程序非常慢。因此决定使用 RSSLib4J。因为本实验只需要对 RSS Feed 进行解析，而 RSSLib4J 恰好提供了这个功能，简单有效，体积小，兼容性好，同时支持 RSS 0.9x、1.0、和 2.0 版本的标准。

RSSLib4J 的解析方法比较简单，代码片断如下所示：

Listing 4.5 RSSLib4J 解析 RSS 代码片断

```

1     RSSHandler hand = new RSSHandler();
2     RSSParser.parseXmlFile(new URL(url), hand, false);
3     RSSChannel ch = hand.getRSSChannel();
4     System.out.println(ch.toString());
5     LinkedList lst = hand.getRSSChannel().getItems();
6     for (int i = 0; i < lst.size(); i++) {
7         RSSItem itm = (RSSItem) lst.get(i);
8         System.out.println(itm.toString());
9     }

```

通过手工将五大新闻网站提供的 RSS Feeds 链接存放在一个本地的文本文件中，每行存储一个链接，定期通过 RSSLib4J 逐条访问该文件存储的链接，获得 XML 文件内容，并对其解析。目前已经收集 RSS Feeds 链接 212 条，该文件可以不断添加新链接，具有很好的可扩展性。

Listing 4.6 RSS Feeds 链接文件部分内容（以#开头的为注释行）

```

1 #cnet
2 #ALPHA: the CNET blog
3 http://reviews.cnet.com/4534-10921_7-0.xml
4 http://reviews.cnet.com/4534-10921_7-0.xml?&node=6033
5 http://reviews.cnet.com/4534-10921_7-0.xml?&node=10863
6 http://reviews.cnet.com/4534-10921_7-0.xml?&node=3504
7 http://reviews.cnet.com/4534-10921_7-0.xml?&node=3118
8 http://reviews.cnet.com/4534-10921_7-0.xml?&node=9020
9 http://reviews.cnet.com/4534-10921_7-0.xml?&node=6450
10 http://reviews.cnet.com/4534-10921_7-0.xml?&node=3513
11 #EDITORS' CHOICE
12 http://reviews.cnet.com/4924-5_7-0.xml?7eChoice=1&orderBy=-7rvDte&maxhits
    =25&dedup=1

```

现在面临的主要问题是，五个网站的 RSS 中的 pubDate 项的格式有很多种，

如何将其正确地解析成为程序可以识别的统一格式，是能否将新闻文本正确存储的重要原因，这是之后进行新词语提取的重要因素。因此在这一方面花费了较多精力。表 4.2 是查看各 IT 网站的 RSS 文件后的统计情况，这是算法实现对发布时间解析的重要标准。

表 4.2 五大英文 IT 网站 RSS 标准和时间格式列表

CNet	Version 2.0	Wed Mar 08 16:23:00 PST 2006
		Tue, 14 Jun 2005 03:00:00 PDT
		February 10, 2006
Slashdot	Version 1.0	2006-03-10T11:40:48+00:00
ZDNet	Version 2.0	Wed, 17 May 2006 18:10:00 PDT
PCWorld	Version 2.0	Wed, 17 May 2006 19:56:04 EST
PCMag	Version 2.0	N/A
pubDate标准	Version 2.0	Sat, 07 Sep 2002 00:00:01 GMT
dc:date标准	Version 1.0	1994-11-05T08:15:30-05:00

因此需要算法实现能够识别各种日期格式，最终将其转换为“YYYY-MM-DD”的格式。由于该目标格式是“Dublin Core”的“dc:date”的前缀，因此对 RSS 1.0 的时间格式，只需要将其前几个字符取出即可以作为目标格式。

需要关注的主要是对 RSS 2.0 的时间格式及“CNet”的两种非规范格式的解析。以下是对三种日期格式进行解析的正规表达式代码片断。其中 Format 1 是 RSS 2.0 标准，而 Format 2 和 Format 3 是 CNet 中出现的另外两种非规范格式。

Listing 4.7 解析 RSS 日期正规表达式

```

1 String week = "(Mon|Tue|Wed|Thu|Fri|Sat|Sun)";
2 String zone = "(UT|GMT|EST|EDT|CST|CDT|MST|MDT|PST|PDT|((\\+|\\-))\\d{4})";
3 String hour = "\\d{2}:\\d{2}(:\\d{2})?";
4 String month = "(Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec)";
5 String monthFull = "(January|February|March|April|May
6     |June|July|August|September|October|November|December)";
7 String space = "(\\ )*";
8
9 //Format 1: Sun, 12 Mar 2006 03:04:47 PST
10 String standard1 = "(" + week + "\\,)" + space + "\\d{1,2}" + space
11     + month + space + "\\d{4}" + space + hour + space + zone;
12 //Format 2: Thu Mar 02 10:51:00 PST 2006
13 String standard2 = week + space + month + space + "\\d{2}" + space
14     + hour + space + zone + space + "\\d{4}";

```



```
15 //Format 3: March 02, 2006
16 String standard3 = monthFull + space + "\\d{2}\\," + space + "\\d{4}";
```

如表 4.2 所示，有的 RSS 文件、或文件的部分新闻项，它们的发布日期是缺失的。对这个问题的解决方案是，配置一个 LOG 文件，记录每天读取的没有发布日期的新闻文本标题。这样，每天读取新的新闻文本，如果没有发布日期，那么将其标题与 LOG 文件中的记录比较，看是否之前已经读取过，如果没有，则将该新闻文本放入到当天目录中。之所以这样做，是因为有些 RSS 文件的更新速度较慢，有些新闻项会长期存在其中，如果不做 LOG 文件记录，那些没有发布日期的新闻文本会被存放在多个日期目录中，最终会降低新术语发现的准确程度。



## 第五章 N 元词串统计

这一部分的主要功能是，通过统计的技术，提取通用语料库和背景语料库的特征，在下一流程“候选新术语发现”部分中，用来进行不同语料库之间的比对，提取候选新术语。

在该部分主要统计单词（Unigram）和二元词串（Bigram）的 TF（Term Frequency）和 DF（Document Frequency）。其统计流程和方法比较类似，下面以二元词串统计过程为例描述算法过程。

### 5.1 流程介绍

该部分的方法比较简单，主要是一个递归遍历某文件夹下所有文件的过程。每当读入一个文件路径，就对该文件内容进行 Bigram/Unigram 的统计，存放在 HashMap（该变量命名为 FileHashMap）数据结构中，当对该文件统计完毕，再将 FileHashMap 的内容加入到另一个全局的 HashMap（命名为 DirHashMap）数据结构中。当对该语料库的所有文件统计结束后，DirHashMap 中就包含了最终的统计结果，将其输出到文件，供之后的流程使用。

为了避免使统计数据过于庞大，在进行 N 元词串统计的过程中，引入了类似于 stop word list 的概念——“frequent word list”。stop word（停词）是搜索引擎技术中的一个概念，许多常用词，例如“a”，“the”等，虽然会经常使用，但是在搜索引擎中，对搜索结果毫无益处。因此在大多数搜索引擎中，一般设有 stop word list，在建立索引的时候，会直接忽略这些词，并且不支持对这些词语作为 Query 的搜索请求。这也在一定程度上减小了搜索引擎的索引空间和检索时间。在这里，我们引入 frequent word list，其中包含从 Brown Corpus 中统计出来的频度最高的 5000 个单词。<sup>①</sup>

<sup>①</sup> 该词表取自：<http://www.edict.com.hk/textanalyser/wordlists.htm>

表 5.1 Frequent Word List 片断 (前 10 个词)

Rank	Word	Instances	Frequency(%)
1	The	69970	6.8872
2	of	36410	3.5839
3	and	28854	2.8401
4	to	26154	2.5744
5	a	23363	2.2996
6	in	21345	2.1010
7	that	10594	1.0428
8	is	10102	0.9943
9	was	9815	0.9661
10	He	9542	0.9392

出于和“stop word list”同样的目的,引入这个 frequent word list,也是为了减小统计过程的时间和统计结果的空间。具体做法是在上述从文件内容统计词语到 FileHashMap 过程中,如果发现 Bigram 的两个词都出现在 frequent word list,或者 Unigram 的单词出现在其中,则不对其进行统计。这样可以大大降低 FileHashMap 和 DirHashMap 的规模,加速统计的运行速度。

其中对每个文件的统计,具体流程如下:

1. 首先读入文件内容,全部转换为小写形式,划分句子;
2. 对每个句子通过空格划分成单词序列;
3. 遍历单词序列,统计词语到 FileHashMap,如果 Bigram 的两个词都出现在 frequent word list,或者 Unigram 的单词出现在其中,则不对其进行统计;
4. 将 FileHashMap 词条加入到 DirHashMap 中。

## 5.2 流程实现

该流程在逻辑上比较简单,但是在具体实现中,会有一些难点。以下对两个难点进行一下介绍。

### 5.2.1 划分句子

算法的一个关键步骤,是对文本进行句子划分。这个问题看似很小,如果把握不好,会使统计结果大相径庭,影响算法最后的正确度。在算法实现中,我们

使用正规表达式来表示出句子的边界，并以此来划分句子。

首先在 Java 中书写正规表达式，需要注意以下问题：

1. 正规表达式中,对于作为表达式中的一个元素,如“,”，“.”等和表达式的保留字符(如“Xn,+”中的“,”)相区别的时候,需要通过转义符号“\”对其进行标注。
2. 正规表达式中,“\”在写成Java的String字符串的时候,需要在其前再加“\”转义。
3. 对于空格,可以在表达式中直接键入空格表示,但是这样只能匹配键入数量的空格。因此需要注意,在书写正规表达式时,表达式对键入的空格是敏感的。

其次在划分句子,有几点特殊处理和需要特别注意的细节:

1. 虽然称为划分句子,实际上,是划分语义连续的单词序列。因此,并非只有句点、逗号等才成为句子边界,包括“”,”,”“(,”)”等都被看作是句子的边界。
2. 在语料库,尤其是通用语料库中,存在大量的“I'd”的现象,在上述的 frequent word list 中,是将这看作一个单词的,因此在划分句子的时候,需要将“I'd”中的“'”与文中表示讲话的引号“'”区分开来。
3. 需要注意作为连词符号的“-”与该符号用在其他情形下的区别,例如在一些文本中用几个“-”作为破折号,或者用作注释。前者不能作为分句边界,而后者需要作为句子边界。
4. 需要注意处理多个标点符号在一起的情况,如文中某人讲话结束,会有“.”或者“:”等,否则如果单独处理,则会在划分的句子中出现大量的空串的现象,因为会误认为“”和“.”之间存在一个句子。

### 5.2.2 统计数据的存储结构选取

在算法设计中,可以使用两种数据结构进行词串统计,首先是 HashMap,其次是树结构。它们的特点分别如下:

1. HashMap 通过哈希桶的方式存储词串和它频度的 Map 结构,查找速度和插入速度较快,但是空间复杂度较高;
2. 树结构,每个节点记录一个字母或其他字符,从根节点到叶节点记录一个词串序列,并在各词串结束的节点记录该词串的频度统计值,其查找速度

和插入速度比 HashMap 要快，但是由于每个非叶节点，都需要保存几十个（取决于其下一个字符的取值范围）指向子节点的指针，因此空间复杂度会比较高。而且由于数据结构不杂，调试比较困难。

在实验中，将这两种方式都进行了实现。发现 HashMap 从时间和空间复杂度都优于后者。因此采用 HashMap 的数据结构来存储统计数据。

在实现中设置了两个变量，FileHashMap 和 DirHashMap。之所以要这样划分，有以下两点原因：

- 为统计 DF 的方便。将每个文件的词语统计结果存放在 FileHashMap 中，在由 FileHashMap 向 DirHashMap 添加的时候，可以比较方便地对这些词语的 TF 进行设置：如果在 DirHashMap 中没有出现过该词语，TF 就为 1；如果已经出现过，则在原来基础上加 1。
- 降低算法时间复杂度。如果统计每个文件的词语直接加入到 DirHashMap，每次都要在 DirHashMap 这个庞大数据结构上进行查找操作，会让算法运行时间大大加长。

### 5.3 统计结果

对用语料库和背景语料库词语的统计结果如下：

表 5.2 通用语料库和背景语料库统计表

文件名	数目
通用语料库单词表	290259
背景语料库单词表	531369
通用语料库二元词串表	3845050
背景语料库二元词串	2912505

需要说明的是，词表的顺序是按如下规则进行的：首先按照 TF 降序排序，如果 TF 相同再按 DF 降序排序。以下是几个词语表的片断。

表 5.3 通用语料库、背景语料库二元词串表片断（前10个二元词串）

Rank	通用语料库二元词串	TF	DF	背景语料库二元词串	TF	DF
1	in vain	2630	1407	digital cameras	718010	145299
2	to marry	2605	1147	mp3 players	559628	144342
3	fond of	2569	1415	cell phones	380275	133113
4	the princess	2557	310	ziff davis	296656	54550
5	the duke	2489	413	laser printers	261697	68414
6	the midst	2445	1464	inkjet printers	247240	54874
7	the king's	2349	689	cnet networks	228420	111699
8	midst of	2337	1407	pc magazine	215460	57303
9	thou art	2174	512	digital camcorders	206004	57361
10	the castle	2032	458	pc world	193396	37032

表 5.4 通用语料库、背景语料库单词表片断（前10个二元词串）

Rank	通用语料库单词	TF	DF	背景语料库单词	TF	DF
1	o	22711	2404	cnet	1532813	114112
2	thou	21820	1260	digital	1462935	189309
3	thy	16267	1172	com	1425222	207619
4	ye	15008	1397	software	1125663	203359
5	thee	14304	1206	pc	1045624	157961
6	de	9297	1250	cameras	940549	179059
7	exclaimed	6783	2205	desktops	860330	145926
8	honour	6655	1990	tech	842498	170211
9	afterwards	6080	2411	mp3	820968	168299
10	hath	5973	717	printers	789361	140170

可以看到，排名前十的词语中已经没有常见词语如“a”、“the”等。在单词表中，这些 frequent words 可能只占 5000 个位置，但在二元词串表中，这些高频词的组合多达  $5000 * 5000 = 25,000,000$ ，剔出一些不可能的搭配，这些高频词的可用组合仍然相当可观。需要注意的是，在后面使用到这些词语表的时候，需要与 frequent word list 配合使用，才能真正体现通用语料库和背景语料库的特征。





## 第六章 候选新术语发现

“候选新术语发现”部分的主要功能是通过语料库之间的比对，从即时新闻语料库中提取候选新术语，供下一流程“时间序列分析”对其进行分析，最终确定新术语。该流程主要包括两个部分：一部分是孤岛词串发现算法，在新闻文本中发现候选新术语；另一部分是对发现的候选新术语进行统计和存储。

### 6.1 孤岛词串发现算法设计

该算法主要功能是：发现孤岛词串，对孤岛词串进行修整，并最终确认输出候选新术语。算法流程如下图所示：

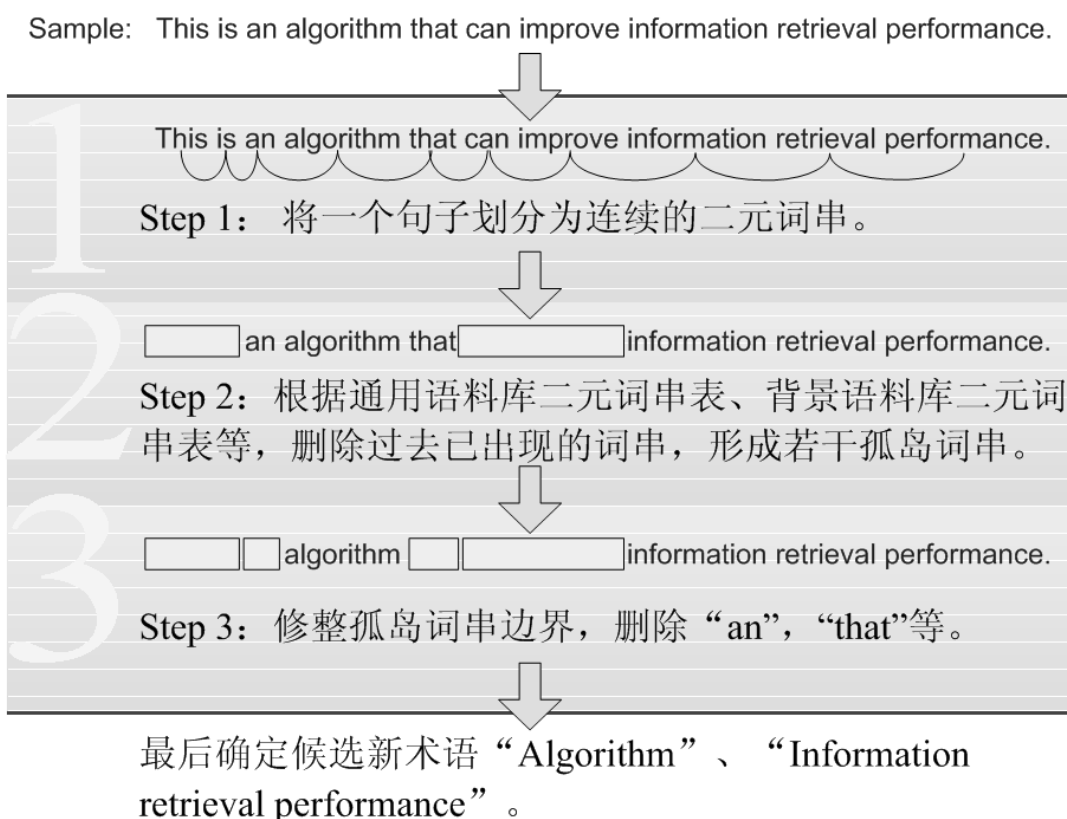


图 6.1 孤岛词串发现算法示意图

此算法由三个步骤组成，在详细叙述之前，需要事先引入几个词语表和定义几个集合。首先在该部分的“孤岛词串发现算法”中需要使用大量的词语表，它

们分别是：

- 通用语料库二元词串表、背景语料库二元词串表：它们用在“孤岛词串发现算法”的第2步，主要作用是删除以前出现过的词语，发现孤岛词串。另外它们也用在“孤岛词串发现算法”的第3步，用来删除孤岛词串是二元词串且在以前出现过的情况。
- Frequent Word List（高频词表）：也是用在“孤岛词串发现算法”的第2步，主要作用是和通用语料库二元词串表、背景语料库二元词串表一起删除以前出现过的词语，发现孤岛词串。另外也用在“孤岛词串发现算法”的第3步，用来删除孤岛词串是单词且在以前出现过的情况。
- Stop Word List（停词表）：用在“孤岛词串发现算法”的第2步，主要作用也是删除以前出现的词语，发现孤岛词串。在这里我们采用 Google 提供的 Stop Word List<sup>①</sup>，其中只含有少量的人称代词，助词，冠词等。

表 6.1 Google Stop Word List

a	about	an	are	as	at	be	by
com	for	from	how	in	is	it	of
on	or	that	the	this	to	was	what
when	where	who	will	with	the	www	

- Empty Word List（虚词表）。该词表来源是 DVL/Verity Stop Word List<sup>②</sup>，并将其中的动词删除。该虚词表用在“孤岛词串发现算法”的第2步，主要作用是修整孤岛词串边界。这里之所以不用 Stop Word List，是其中含词量太小；不用 Frequent Word List，是其中含有大量有实际意义的动词，如果将这些词从孤岛词串的边界去掉，极有可能造成错误。
- 通用语料库单词表、背景语料库单词表：它们用在“孤岛词串发现算法”的第3步，主要作用是删除孤岛词串是单词且在以前出现过的情况。
- Current IT Term List：该词表包括目前收集到的计算机类新术语。用在“孤岛词串发现算法”之后，查看修整后的孤岛词串是否在之前已经作为新术语被检测到。它有两个来源：一个来源是 IT 类网站 PC Magazine 的 IT

① 该列表取自 <http://www.ranks.nl/tools/stopwords.html>。

② 取自 [http://dvl.dtic.mil/stop\\_list.pdf](http://dvl.dtic.mil/stop_list.pdf)，虽然命名为 Stop Word List，除了常用助词、代词外，里面还包括高频动词、副词和形容词等。

Encyclopedia<sup>①</sup>。其中收集了超过 2000 个 IT 类术语。另一个来源是该算法发现新术语会自动存放到该术语表中。这个词表主要用来对修整后的孤岛词串，查看其是否已经是之前发现过的（新）术语。

以下是根据几个词语表的内容定义的几个集合：

- 设旧二元词串表中频度最高的前 10000 个二元词串集合为 $\mathbb{O}$ ，通用二元词串表中频度最高的前 10000 个二元词串集合为 $\mathbb{G}$ ，Frequent Word List 中单词组成的二元词串集合为 $\mathbb{F}$ ，那么可以对这些集合取并集得：

$$\mathbb{A} = \mathbb{O} \cup \mathbb{G} \cup \mathbb{F}.$$

- 设 Stop Word List 中单词组成的二元词串集合为 $\mathbb{S}$ 。
- 设 Empty Word List 单词组成的集合为 $\mathbb{E}$ 。
- 设通用语料库单词表中频度最高的前 5000 个单词集合为 $\mathbb{N}$ ，背景语料库单词表中频度最高的前 5000 个单词组成的单词集合为 $\mathbb{L}$ ，Frequent Word List 组成的单词结合为 $\mathbb{R}$ ，那么可以对这些集合取并集得：

$$\mathbb{W} = \mathbb{N} \cup \mathbb{L} \cup \mathbb{R}.$$

- 设 Current IT Term List 单词组成的集合为 $\mathbb{I}$ 。

下面详细叙述算法主体三个步骤的具体实现。

### 第一步：划分二元词串

划分二元词串的过程比较简单。首先将新闻文本转换为小写形式。对其进行划分句子。划分句子在第 5.2.1 小节已经详细叙述。然后将每一句子划分为二元词串。对一个句子的单词序列，表示如下：

$$w_1 \quad w_2 \quad w_3 \quad \dots \quad w_{n-1} \quad w_n.$$

可以得到二元词串：

$$B_1, B_2, B_3, \dots, B_{n-1}.$$

很明显，除去句首和句尾的两个单词，其他单词都分别隶属于相邻的两个二元词串。换句话说，任何相邻的两个二元词串都在一个单词上重叠。用数学公式

<sup>①</sup> 其网页为<http://www.pcmag.com/encyclopedia/>。

表示就是：

$$w_i, w_{i+1} \in B_i \quad \forall i \in [1, n-1]. \quad (\text{公式 1})$$

### 第二步：初步提取孤岛词串

第二步完成初步提取孤岛词串。这需要用到第 5 章“N 元词串统计”部分得到的通用语料库词串表和背景语料库词串表，这一步还要用到 Stop Word List。

直观来讲，我们对单词是否在之前出现过，有以下判断：

1. 如果一个二元词串的两个单词都是 stop word，那么这种组合是新词语或新词语一部分的可能性非常小。因此这两个单词应该被标记为 Appeared。在很小的概率下，即使该二元词串是新词语的一部分，这样标记导致的后果只是把新词语分割为两部分，并不影响正确找到该新词语。
2. 单词序列中如果一个单词所在的两个二元词串都已经在语料库中以很高的频率出现过，那么这个单词一般不太可能是新词语或其中的一部分，而是在过去曾经经常出现的旧单词，因此应当被标记为 Appeared。而如果单词所在的某个二元词串未曾在语料库中以很高频率出现过，那么该单词可能作为该二元词串的一部分，成为新词语，该单词应当被标记为 Unappeared。
3. 比较特殊的情况，是如果  $B_1$  或  $B_{n-1}$  已经在语料库中以很高的频率出现过，由于再没有前后另一个词串的影响，我们简单地将  $w_1$  或  $w_n$  标记为 Appeared。

详细算法描述为，首先对其中任何一个单词  $w_i$  用以下函数判别：

$$f(w_i) = \begin{cases} \text{Appeared} & \text{if } B_{i-1} \in \mathbb{S} \text{ or } B_i \in \mathbb{S} \\ \text{Appeared} & \text{if } i = 1 \text{ and } B_1 \in \mathbb{A} \\ \text{Appeared} & \text{if } i = n \text{ and } B_{n-1} \in \mathbb{A} \\ \text{Appeared} & \text{if } i \in (1, n) \text{ and } B_{i-1}, B_i \in \mathbb{A} \\ \text{Unappeared} & \text{others} \end{cases} \quad (\text{公式 2})$$

通过该函数可以对该句的每个单词打上标签 Appeared/UnAppeared，然后将单词序列中标签为 Unappeared 的连续的单词提取出来，这就是孤岛词串。

### 第三步：修整孤岛词串边界

孤岛词串提取出来之后，需要对其边界进行修整，如图 6.1 中，Step 2 之后，提取出的“an algorithm that”中，从直观来讲，“an”、“that”都不应当是新词语的

一部分，需要将它们去掉，得到真正的新词语“algorithm”。这一步骤的详细算法描述如下：

1. 对一个孤岛词串，如果它是一个单词，查看其或者其 Stemming 之后的形式是否属于集合 $\mathbb{W}$ ，即是否在之前很高频度出现过，如果是，则丢弃，否则进入步骤 3；如果它是一个二元词串，查看其是否属于集合 $\mathbb{A}$ ，即是否在通用语料库二元词串表和背景语料库二元词串表的高频部分或 Frequent Word List 的二元词串组合中出现，如果是，则丢弃，否则进入步骤 2；如果它是一个含有超过三个单词的词串，进入步骤 2。
2. 查看词串的首尾单词是否属于集合 $\mathbb{E}$ ，即是否出现在 Empty Word List 中，如果是则将该单词从这个词串中删除，进入步骤 3；
3. 如果该词串较 1 时有修改，则递归重新进入步骤 1，对修整后的新词串进行新一轮修整；否则进入步骤 4。
4. 查看词串是否属于集合 $\mathbb{I}$ ，即是否出现在 Current IT Term List 中，如果是，则丢弃；否则，输出为新术语候选词串。

由于孤岛词串发现算法使用了 Stemming，所以进行一下简单介绍。这部分主要参考文献[10,11,28-32]。

在信息检索中，把 foxes 映射为 fox 的类似问题称为词干还原 (stemming)。除了复数变化之外，词干还原还可以应用于处理词缀。例如，可以处理以-ing 结尾的英语动词。有时高精确率的信息检索系统往往很难做到，其中一个重要原因就是对于一个单词，其往往有多种形态，例如“give”，“given”，“giving”虽然在形态上并不相同，但在意义上却可以被认为是一个单词。因此在信息检索中，为了减少因为单词的不同形态造成的匹配失误，会采取词干还原的技术对 Query 等进行处理，将其还原为基本的单词形态。词干还原，不仅可以提高信息检索系统的精确率，也可以减小 Dictionary 的大小，降低查找的时间复杂度和空间复杂度。

对于信息检索系统，一般不会在乎经过词干还原后的单词是否是真正意义上的单词，因此，“computation”会被还原成“comput”而不是“compute”；“policies”被还原成“polic”而非“police”。

Porter (1980) 和 Lovins (1968) stemming 算法是对英语的词干还原算法中最流行的两种。这两种都采用启发式规则进行英语后缀的去除。除此之外，另一种途径是利用字典的方式建立一个精确的列表，存放每个单词对应的词干。但

是相对来讲，启发式规则的方法更被接受，因为建立一个准确的字典列表需要花费相当的人力。

Lovins 词干还原算法比 Porter 更为激进或冒险。也就是说，Lovins 更有可能把两个单词还原为同一种形式，但同时也极有可能出错。例如，Lovins 能够将“psychology”和“psychologist”正确还原为“psychologi”，Porter 算法会错误地将他们分别还原为“psychologi”和“psychologist”。但是，另一方面，Porter 可以正确将“police”和“policy”映射到“polic” and “polici”，但 Lovins 会错误把它们还原为同一种形式“polic”。

在程序中我们采用 Porter 本人开发的 Snowball Stemmer 开源工具<sup>①</sup>。这是 Porter 为了将 Porter 词干还原算法推广到除英语之外的其他语言而设计的。该程序具有良好的可扩展性，允许使用者自定义规则进行词干还原。在这里，我们只是简单使用其中的英语词干还原接口。其接口的使用示例如下：

Listing 6.1 Snowball Stemmer 对英语单词进行词干还原代码示例

```
1 String word = "police";
2 englishStemmer stemmer = new englishStemmer();
3 stemmer.setCurrent(word);
4 stemmer.stem();
5 String stemmed = stemmer.getCurrent();
```

Snowball 接口比较简单，如示例代码，第 1 行是要进行词干还原的单词，在代码第 3 行将其送入 Stemmer，第 4 行进行词干还原，第 5 行将还原后的词干输出。

## 6.2 存储形式与数据结构

当孤岛词串在被确认为候选新术语之后，应当如何统计和存储呢？经过多方面的考虑，在算法实现中准备采用 XML 文件的形式存放。采取这种格式的主要原因是 XML 文件格式具有很好的可读性，无论是对人还是对计算机程序。

<sup>①</sup> 该开源项目主页为<http://snowball.tartarus.org/>。

---

## 6.2.1 XML 规范及读写简介

XML 即为可扩展的标记语言 (eXtensible Markup Language)。XML 是一套定义语义标记的规则, 这些标记将文档分成许多部件并对这些部件加以标识。理解 XML, 首先要理解标记。在 HTML 里每个标志都是有确切含义的。例如, 在 HTML 中, 标签<B>的含义是要求 HTML 浏览器将一段文本加粗表示。而 XML 并非象 HTML 那样, 提供了一组事先已经定义好了的标签, 而是提供了一个标准, 利用这个标准, 可以根据实际需要定义自己的新的置标语言, 并为你的这个置标语言规定它特有的一套标签。准确的说, XML 是一种源置标语言, 它允许你根据它所提供的规则, 制定各种各样的置标语言。<sup>[33]</sup>

开发 XML 应用程序时常用到几种模型, 我们可以使用根据这些模型创建的 API 来分析和操纵 XML 结构, 这些模型可以是基于对象的, 如文档对象模型 (Document Object Model, DOM) 和 JDOM; 也可以是基于事件的, 如 Simple API for XML (SAX)。另外, Java API for XML Parsing (JAXP) 提供了使用 DOM、SAX 和 XSLT 处理 XML 文档的通用接口。

在本算法实现中, 是使用开源软件 dom4j 解析器完成对 XML 的读写。dom4j 是一种解析 XML 文档的基于 Java 语言的开放源代码。该解析器提供了对 DOM、SAX 和 JAXP 的全面支持。源码包括 dom4j 类、XPath 引擎以及 SAX 和 DOM 接口<sup>[34]</sup>。本算法实现使用的是 2005 年 04 月 15 日发布的 1.6 版本。

以下简单介绍如何使用 dom4j 接口进行 XML 文件的读写。首先解析 XML 文件的方法非常简单, 给出 XML 文件的 URL, 既可以通过以下代码解析得到 Document 对象:

Listing 6.2 dom4j 解析 XML 文件代码示例, 摘自参考文献[35]

```
1 import java.net.URL;
2 import org.dom4j.Document;
3 import org.dom4j.DocumentException;
4 import org.dom4j.io.SAXReader;
5
6 public class Foo {
7     public Document parse(URL url) throws DocumentException {
8         SAXReader reader = new SAXReader();
9         Document document = reader.read(url);
10        return document;
11    }
```

```
12 }
```

那么如何通过 Document 对象得到自己想要的内容, 这需要通过 Iterator 迭代器来完成, 代码如下:

Listing 6.3 dom4j 通过 Iterator 迭代器获取 XML 项或属性代码示例, 摘自参考文献[35]

```
1 public void bar(Document document) throws DocumentException {
2
3     Element root = document.getRootElement();
4
5     // iterate through child elements of root
6     for ( Iterator i = root.elementIterator(); i.hasNext(); ) {
7         Element element = (Element) i.next();
8         // do something
9     }
10
11    // iterate through child elements of root with element name "foo"
12    for ( Iterator i = root.elementIterator( "foo" ); i.hasNext(); ) {
13        Element foo = (Element) i.next();
14        // do something
15    }
16
17    // iterate through attributes of root
18    for ( Iterator i = root.attributeIterator(); i.hasNext(); ) {
19        Attribute attribute = (Attribute) i.next();
20        // do something
21    }
22 }
```

以上是如何解析 XML 文件和获取内容。下面代码演示如何通过程序生成 Document, 该示例生成的 Document 中的“root”节点下有两个“author”项。

Listing 6.4 dom4j 生成 Document 文档代码示例, 摘自参考文献[35]

```
1 import org.dom4j.Document;
2 import org.dom4j.DocumentHelper;
3 import org.dom4j.Element;
4
5 public class Foo {
6     public Document createDocument() {
7         Document document = DocumentHelper.createDocument();
8         Element root = document.addElement( "root" );
9     }
10 }
```



```

10     Element author1 = root.addElement( "author" )
11         .addAttribute( "name", "James" )
12         .addAttribute( "location", "UK" )
13         .addText( "James Strachan" );
14
15     Element author2 = root.addElement( "author" )
16         .addAttribute( "name", "Bob" )
17         .addAttribute( "location", "US" )
18         .addText( "Bob McWhirter" );
19
20     return document;
21 }
22 }

```

如果要将 Document 输出到文件，只需要通过 FileWriter 类就可以了。

### 6.2.2 通过 XML 存储候选新术语信息

对于在 XML 中具体的存储形式，总结下来有两种可行的方案：

- I 以词串作为记录单位，每个 XML 文件记录一个候选新术语在一段时间内出现的情况。
- II 以时间为记录单位，每个 XML 文件记录某天出现的所有候选新术语的情况。

这两种方案，通过分析最终选择方案II，其主要原因如下：

1. 在“时间序列分析”部分，我们主要对一段时间内候选新术语的出现频度进行考查，从而得出“某天出现哪些新术语”。这样，我们首先要找到在该天出现的候选新术语，然后考查其后一段时间内的频度。而如果采用方案I的话，每次都要对所有的候选新术语进行遍历，确定哪些候选新术语是在该天出现的。这会非常耗费时间，在逻辑关系上也比较繁复。
2. 而方案II以时间为记录单位，记录每天出现的候选新术语的情况，则可以很好解决以上的问题。当要考查4月1日出现的候选新术语时，只需要将该天的候选新词语读入，并在之后的一段时间内的统计记录中查找这些候选新术语的出现情况。

在方案II中 XML 文件每个候选新术语设置了以下几个属性或项，以便“时间序列分析”部分进行统计、分析和及时找到该候选新术语出现的位置：

- 候选新术语词串；

- 出现日期：由于方案II是按时间进行候选新术语存放的，因此此项用处并不是很大，只是为了防止以后可能用到；
- 该日期出现 TermCount：统计在该日期的新闻文本中，该候选新术语出现的 Term Frequency；
- 该日期出现 DocCount：统计在该日期的新闻文本中，该候选新术语出现的 Document Frequency；
- 在文本中位置：包括文本文件名及行数。

以下是候选新术语 XML 文件格式示例：

Listing 6.5 5月21日的候选新术语统计文件中对“kanweg”的记录

```
1 <Candidate Name="kanweg">
2   <Date>2006-05-21</Date>
3   <TermCount>2</TermCount>
4   <DocCount>2</DocCount>
5   <OccurPositions>
6     <Position>
7       <FileName>.xxx registry sues US government</FileName>
8       <Line>172</Line>
9     </Position>
10    <Position>
11      <FileName>Recipe for Making Symetrical Holes in Water</FileName>
12      <Line>104</Line>
13    </Position>
14  </OccurPositions>
15 </Candidate>
```

对某天新闻文本的候选新术语的 TF、DF 以及在文本中出现的位置等数据的统计分为三级完成：

1. 句子级：统计句子中出现的候选新术语，用 HashSet 存放。
2. 文件级：将所有句子的候选新术语存放 to HashMap<String, ArrayList<Integer>> 中，其中 String 存放候选新术语词串，ArrayList 记录其在文件出现的行数。
3. 目录级：将所有文件的候选新术语放到 HashMap<String, CandidateInfo> 中，其中 String 记录候选新术语词串，CandidateInfo 记录候选新术语的统计数据（TF、DF和出现文件名和行数）。

最后将 HashMap<String, CandidateInfo> 中的数据转换成 XML 文件输出。

在这一部分曾经发现生成的 XML 文件中，许多位置的 TermCount 与下面 OccurPosition 里给出的位置数目不对应，经过调试发现问题在于，当孤岛词串发现后，程序会对其有修整边界和查看是否在 Current IT Term List 出现的过程，而在刚开始是把这个过程放在上述的目录级一层完成。这样做的问题是：从文件级的 HashMap 读出孤岛词串，如果两个孤岛词串如：algorithm 和 an algorithm that，进行修整边界之后，后者被去掉了“an”和“that”，和前者同为一个候选新术语了。而在目录级，向 HashMap 加入候选新术语的统计出现位置(文件名+行数的Map)数据的时候，是采用 put 的方式（可以认为同一个文件中统计出来孤岛词串必定互相不同，那么某一孤岛词串的位置信息不会两次put同样一个文件名），这样在放入第二个词串的时候，会把第一个词串的位置信息给冲掉。修改方式是将修整边界等过程提前到文件级做，这样就在文件层杜绝了同一候选新术语在一个文件中被统计两遍以上的情况，也避免统计信息被错误覆盖。



## 第七章 时间序列分析

该部分对候选词串列表中的词语进行分析，考查候选新术语在一段时间内的频度的变化趋势，最终确定新术语表。

### 7.1 时间序列分析的必要性

根据对“新词语”的定义第三条，“新词语必须受到普遍的认可，被广泛地应用，在语言词汇中站稳了脚跟，而非昙花一现”，从而有了“时间序列分析”部分，即查看上一流程“候选新术语发现”得到的候选新术语在一段时间内能否以一定的频度规律出现，而不仅仅是“昙花一现”，如果是后者，则被摒除在新术语之外。正对“新词语”的这一要求，说明进行时间序列分析的必要性。

对于某些新术语，它们的频度趋势是经过一段时间的上升期之后，会变成人们的日常用语，而在之后每天保持一定的出现频度，围绕某一数值上下振动。如著名的“Google”在近几年的频度，其在2005年中对其关注程度有了一个飞跃之后，便趋于平缓，围绕一定的频度上下波动：

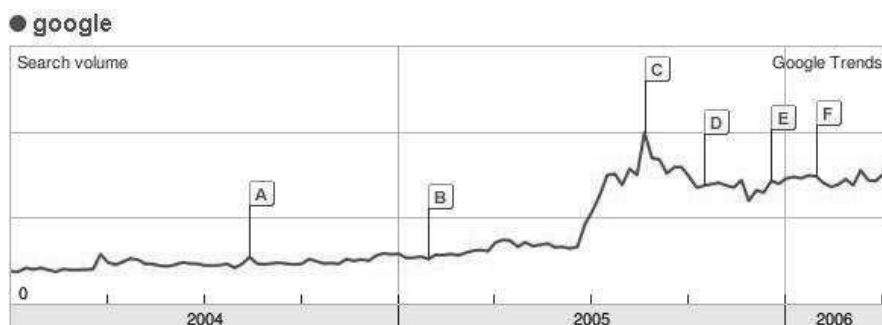


图 7.1 “Google”在搜索引擎Google中被作为关键词查找的频度示意图，取自 Google Trends (<http://www.google.com/trends>)。

再如“Web 2.0”，这是由一批计算机科学专家提出来的新理念，其核心思想是“以 Web 作为平台为人们服务”，从下图可以看到，这一对网络服务的新观念，在 2005 年之后开始受到越来越多的关注，至今仍然没有趋缓的迹象。

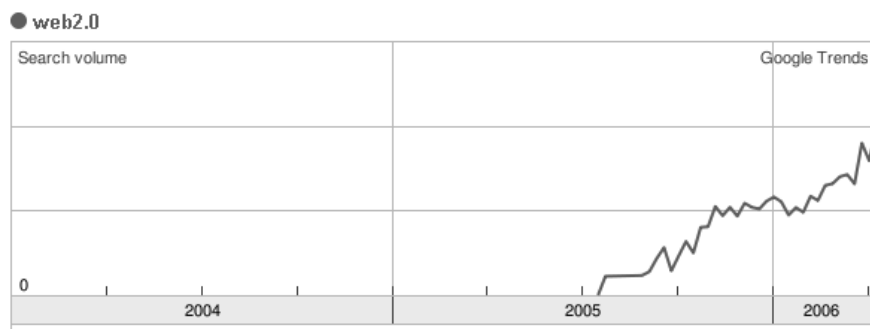


图 7.2 “Web 2.0”在搜索引擎Google中被作为关键词查找的频度示意图，取自 Google Trends。

上述的特点只适合少数能够长期吸引人们关注的事物。对大部分新术语来讲，它们能够做到的，是在刚出现的一段时间内被大众关注、讨论和争议，之后就会从人们关注的视野中消失，只以较低的频度偶尔出现。只有在有更新的相关事件发生的时候才会再次被聚焦。下面的例子将比较直观说明这一特点。

“Google Finance”<sup>①</sup> 和 “Google Notebook”<sup>②</sup> 是著名搜索引擎公司 Google 今年分别于 3 月 22 日和 5 月 16 日推出的两款倍受关注的产品。从下图可以看出，当产品推出前后，会有较多新闻关注，因此在这段时间，产品名称会保持一定的出现频度和关注程度。而这一频度会在一段时间之后衰减，但不会仅仅在某一天出现之后，马上消失匿迹。

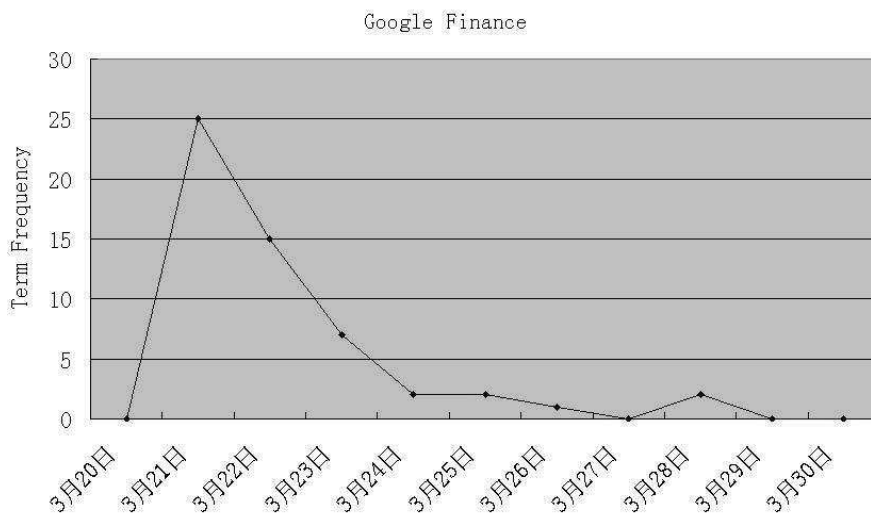


图 7.3 “Google Finance”在发布前后的频度示意图

① 产品链接: <http://www.google.com/notebook>。

② 产品链接: <http://finance.google.com/finance>。

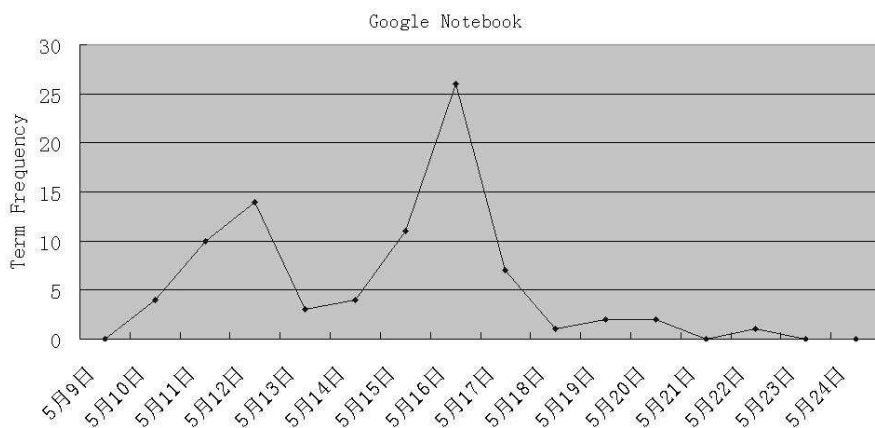


图 7.4 “Google Notebook”在发布前后的频度示意图

时间序列分析的主要功能是对每一个候选新术语，根据其在一段时间内的频度变化，通过一个评价函数，给该候选新术语打分，当分数超过某一个阈值的时候，就最终确定该词语为新术语，提交给用户。

该部分需要提供几个可供调整的参数：

- 开始日期 (start date)：指分析日期数据的开始点。
- 粒度 (granularity)：指分析的基本单位，例如可以以 1 天为单位分析数据，也可以以 7 天为单位分析数据；
- 分析点数目 (analysis number)：指分析的基本单位个数，如以 7 天为单位进行分析，那么需要确定分析几个 7 天的数据。

之所以要求可以设置粒度，从下图可以比较直观得出：

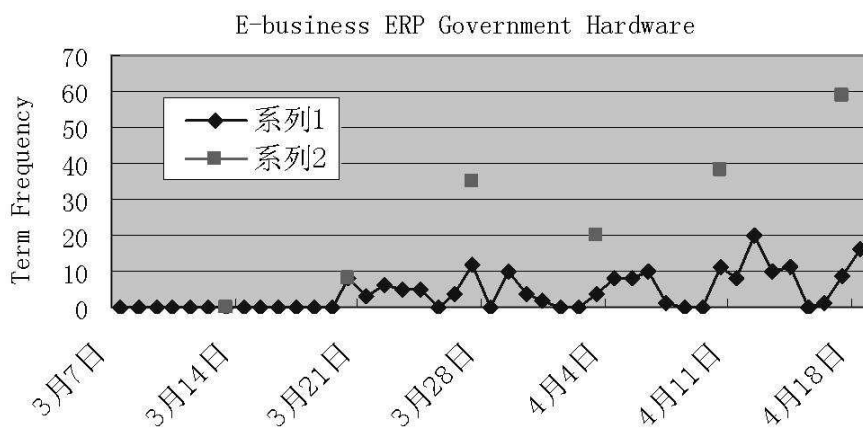


图 7.5 “E-business ERP Government Hardware”的频度示意图

其中系列 1 图示以 1 天为粒度的频度, 系列 2 图示以 7 天为粒度的频度。可以看到, 在系列 1 中的频度上下波动较大, 无法判断其是否为新术语, 而在系列 2 计算每 7 天的频度和, 则能够发现比较明显的上升趋势。

## 7.2 算法设计

首先设定时间序列分析的开始日期为  $s$ , 粒度为  $g$ , 分析点数目为  $n$ 。需要进行以下两个步骤实现该部分功能:

### 7.2.1 聚合数据

“候选新术语发现”部分产生的数据最终存储为 XML 文件格式。在该部分, 首先需要将数据聚合。方法如下:

1. 读出日期  $s$  的 XML 文件中的候选新术语, 构成候选新术语集合  $\mathbb{C}$ ;
2. 对词语集合  $\mathbb{C}$  中每个词语  $t$ , 查看其在从  $s$  开始,  $g \times n$  天内的频度, 得到  $g \times n$  组频度数据  $(w_1, w_2, \dots, w_{g \times n})$ ;
3. 对该频度数据, 每  $g$  为一组, 进行聚合, 这里的操作方式是进行简单的求平均, 即:

$$a_i = \frac{1}{g} \sum_{k=i \times (g-1) + 1}^{i \times g} w_k, \quad i \in [1, n] \quad (\text{公式 3})$$

这样数据  $a_1, a_2, \dots, a_n$  即为聚合结果。

### 7.2.2 评价数据

对聚合数据要进行评价, 通过所得的分数与阈值比较, 判定该候选新术语是否达到新术语的标准。之所以采用评价函数而不是建模的方式, 是因为通过对大量词语频度图的观察, 可以看到它们的频度曲线并不满足某种已知的曲线, 不能通过一般意义的数学建模和数值算法来对其进行拟和。

设候选新术语  $t$  对应的频度数据为  $a_1, a_2, \dots, a_n$ , 经过一些分析, 确定评价函数  $f(a_{i+1}, a_i)$  如下:

$$f(a_{i+1}, a_i) = \begin{cases} 1 & \text{if } a_{i+1} > a_i \\ -0.5 & \text{if } a_{i+1} = a_i \\ -1 & \text{if } a_{i+1} < a_i \end{cases} \quad (\text{公式 4})$$



利用上述评价函数，设阈值为  $\delta$ ，评价某候选新术语的频度数据方法如下：

1. 对频度数据  $a_1, a_2, \dots, a_n$  的任意  $i \in [1, n)$ ，计算  $f(a_{i+1}, a_i)$ ；
2. 通过以下求和公式得到评价函数分数：

$$S = \sum_{i=1}^N f(a_{i+1}, a_i) \quad (\text{公式 5})$$

在  $N = 1 \rightarrow n - 1$  过程中，比较  $S$  和阈值  $\delta$ ，一旦  $S > \delta$  则判定该词语为新术语。

直观来讲，这个评价函数的基本想法是考查候选新术语所有分析点组成的频度曲线上，呈上升趋势的阶段所占比例。

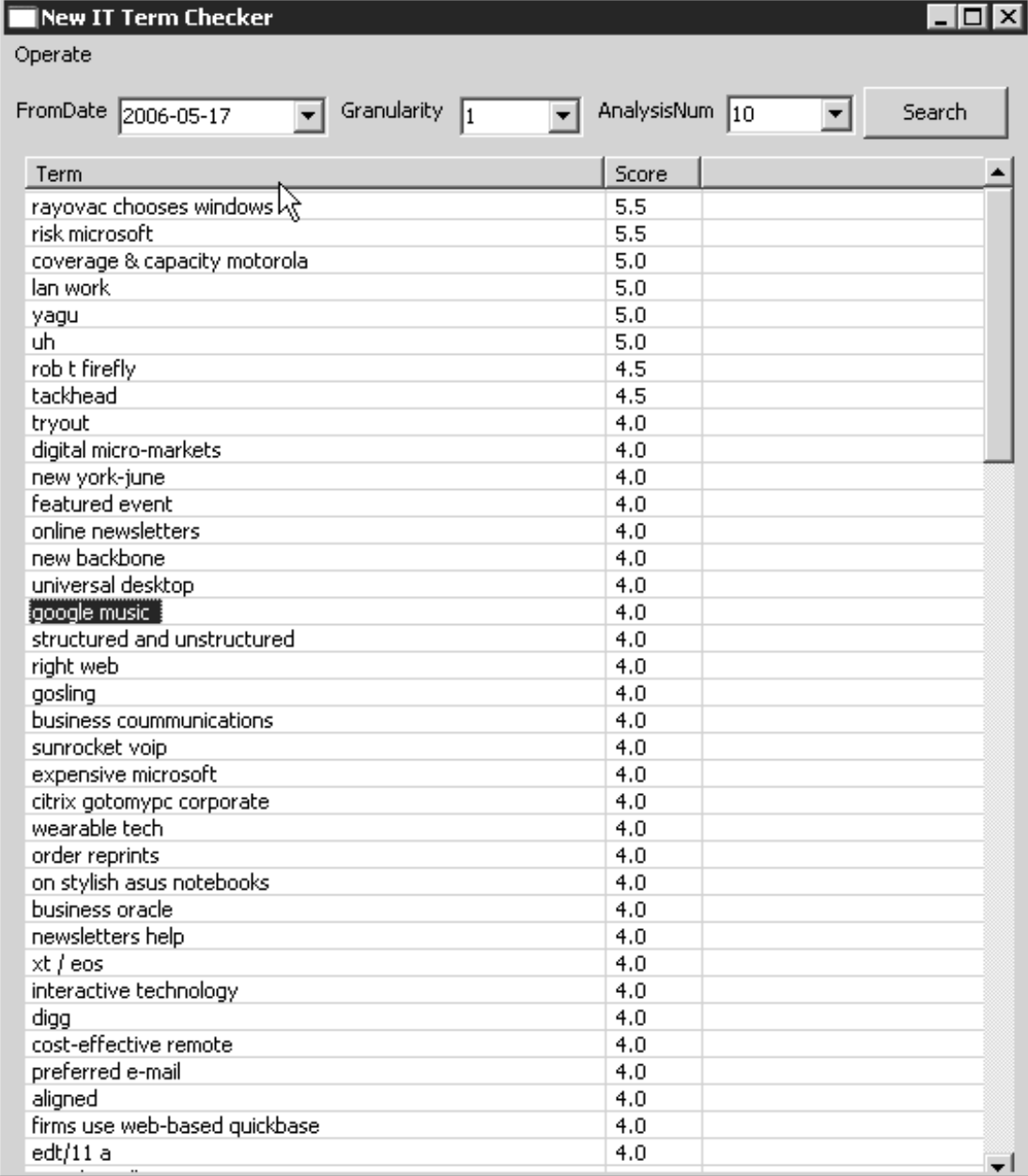
### 7.3 结果查看

为了查看“时间序列分析”之后所得到的新术语的结果，我们设计了一个比较简单的界面。主界面上侧分别是三个下拉列表，可以设置开始日期（start date），粒度（granularity）和分析点数目（analysis number）。设置好以上三个参数之后，点击“Search”按钮，界面会调用“时间序列分析”的相关接口函数，得到新术语及其评价函数分数，在下侧的表格控件中显示，第 1 列为新术语，第 2 列为该术语得分。如图 7.6 所示，是在 2006 年 5 月 17 日，粒度为 1，分析点数目为 10 的参数设置下的主界面示意图。

对于得到的新术语仅仅知道其得分还是不够的，要考查该算法实现的正确性，还需要查看所列新术语的频度波动情况。选中某个新术语，然后选择菜单栏“Operate”下的“Coordinate”选项，就会显示在新术语在选定的分析点上的频度（Term Frequency）波动曲线坐标图。其横坐标是日期，如果粒度大于 1，该日期则是每个分析点的开始日期；其纵坐标是频度值，目前只考虑和显示 Term Frequency。如图 7.7 所示，为 2006 年 5 月 17 日新术语“Google Music”的频度波动曲线坐标图。

此外，还需要查看这些新术语出现的位置，以便人工查看这些新术语是否符合标准要求。点击菜单栏“Operate”下的“Position”选项，就会显示新术语在各日期出现的文件名及行数。该界面的主要视图是一个 TableTree（表树）结构，每个父节点显示日期及在该日期下的 TF 和 DF，而叶节点有三列内容，包括文件名、出现行号以及查看该文件的按钮“File...”；点击“File...”，则会调用本地

的文本编辑器（如 UltraEdit，可以显示行号）打开该文件，从而可以根据第 2 列提供的行号查看新术语出现的位置。如图 7.8 所示，为 2006 年 5 月 17 日新术语“Google Music”的出现位置表树示意图。



Term	Score
rayovac chooses windows	5.5
risk microsoft	5.5
coverage & capacity motorola	5.0
lan work	5.0
yagu	5.0
uh	5.0
rob t firefly	4.5
tackhead	4.5
tryout	4.0
digital micro-markets	4.0
new york-june	4.0
featured event	4.0
online newsletters	4.0
new backbone	4.0
universal desktop	4.0
google music	4.0
structured and unstructured	4.0
right web	4.0
gosling	4.0
business coummunications	4.0
sunrocket voip	4.0
expensive microsoft	4.0
citrix gotomypc corporate	4.0
wearable tech	4.0
order reprints	4.0
on stylish asus notebooks	4.0
business oracle	4.0
newsletters help	4.0
xt / eos	4.0
interactive technology	4.0
digg	4.0
cost-effective remote	4.0
preferred e-mail	4.0
aligned	4.0
firms use web-based quickbase	4.0
edt/11 a	4.0

图 7.6 5 月 17 日新术语列表

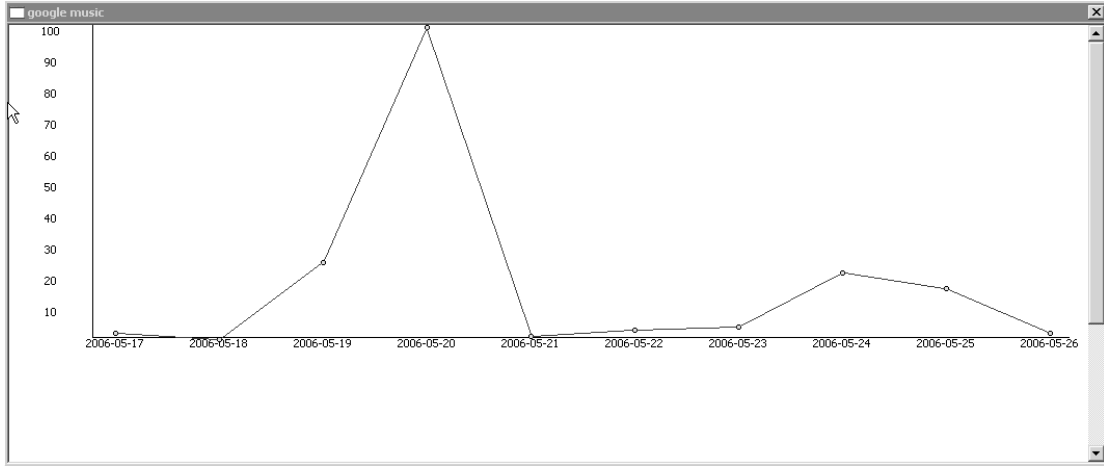


图 7.7 “Google Music” 频度曲线坐标图

Date/Name	TF/Line	DF/Details
2006-05-17	2	2
2006-05-19	24	18
Microsoft# Open source 'not reliable or dependable'	66	File...
Net neutrality field in Congress gets crowded	54	File...
Some apes, birds can think ahead, studies show	52 62	File...
Meet Apple's version of Deadheads	48	File...
Christie's to hold 'Star Trek' garage sale	47 53	File...
Backer of .xxx adult domain tries again	38 48	File...
Bill aims to school judges in patent law	38	File...
Zero-day Word flaw used in attack	55	File...
Microsoft brings back the towels	46	File...
Skype bug may expose user data	30	File...
CBS Radio signs up for Portable People Meter	43	File...
Web inventor says brainchild ready for big leap	58	File...
Dell switch dents Intel shares	39	File...
VC Tom Perkins resigns from HP's board	30 40	File...
Source# State Dept. to limit Chinese computer use	57	File...
Fire fest in Frisco--When it's OK to play with fire	50 60	File...
Apple countersues Creative in patent dispute	46 52	File...
Live from San Francisco, Scott McNealy	42	File...
2006-05-20	97	90
2006-05-21	1	1
2006-05-22	3	3
2006-05-23	4	3
2006-05-24	21	20
2006-05-25	16	16
2006-05-26	2	2

图 7.8 “Google Music” 出现位置示意图



## 第八章 实验结果及分析

### 8.1 准确率分析

设粒度为 1，分析点数目为 10，通过“基于 WEB 的计算机领域新术语的自动检测”算法得到 2006 年 5 月 17 日至 5 月 20 日出现的新术语列表见附录 B 中表 B.1, B.2, B.3 和 B.4。在算法返回的这四天新术语中，通过人工选取，确定其中的真正新术语列表见附录 B 中表 B.7。需要说明的是，受到主观的影响，该人工选定的新术语数目只是对实际新术语数目的估计。接下来就以上述数据进行分析。

表 8.1 算法返回新术语的准确率

日期	人工确定新术语数目	算法返回新术语数目	准确率 (%)
2006.06.17	17	126	13.5
2006.06.18	12	71	16.9
2006.06.19	6	35	17.1
2006.06.20	21	177	11.9

表 8.1 显示 2006 年 5 月 17 日到 20 日的算法返回新术语数目，人工确定新术语数目，以及由此计算的准确率。由此可见，该算法的准确率并不高。但是，正如第 1 章所述，这个算法只起到“预警”的作用，它的最终目的，不是代替用户决定某些新出现的词语是否为新术语，而是把新术语出现的范围大大缩小，减小了用户发现新术语的工作量。

下面通过手动方式对这四天的算法返回新术语和人工确定新术语的来源进行统计。现在把算法返回新术语的来源分为五个部分：广告、用户 ID、新闻正文（包括标题）、用户评论和其它（指导航栏、链接的锚文本等）。需要说明的是，由于网页布置具有很大的随意性，而且人工统计也会受到主观的影响，因此以下的统计数据只是对实际情况的粗略估计。如表 8.2 是算法返回新术语来源分类统计，表 8.3 是人工确定新术语来源分类统计，表 8.4 显示了算法返回各来源新术语的准确率。

表 8.2 算法返回新术语来源分类，第一个数据为个数，第二数据为百分比（%）

日期	广告	ID	新闻正文	评论	其它	总计
2006.06.17	58/46.0	8/6.3	26/20.6	7/5.6	27/21.4	126
2006.06.18	51/66.2	4/5.2	11/14.3	4/5.2	7/9.1	77
2006.06.19	14/40.0	2/5.7	10/28.6	1/2.9	7/20.0	35
2006.06.20	89/50.3	2/1.1	40/22.6	0/0.0	47/26.6	177

表 8.3 人工确定新术语来源分类，第一个数据为个数，第二数据为百分比（%）

日期	广告	ID	新闻正文	评论	其它	总计
2006.06.17	6/35.3	0/0.0	8/47.1	0/0.0	3/17.6	17
2006.06.18	9/75.0	0/0.0	1/8.3	1/8.3	1/8.3	12
2006.06.19	1/16.7	0/0.0	4/66.7	0/0.0	1/16.7	6
2006.06.20	5/23.8	0/0.0	6/28.6	0/0.0	9/42.9	21

表 8.4 各来源新术语的准确率

日期	广告（%）	ID（%）	新闻正文（%）	评论（%）	其它（%）
2006.06.17	10.3	0.0	30.8	0.0	11.1
2006.06.18	17.6	0.0	9.1	25.0	14.3
2006.06.19	7.1	0.0	40.0	0.0	14.3
2006.06.20	5.6	0.0	15.0	0.0	19.1

通过表 8.2 可以看出，算法返回的新术语中，来源于广告的占了近半数。主要原因在于，同一支广告，在发布期间，会在多个新闻网页中出现。这样这些广告虽然可能在每个新闻网页中只出现一次，但会在该天的多个新闻网页中都有出现，这会让该广告中的词串具有较高的 Term Frequency，因此广告中的候选新术语也在“时间序列分析”中较有可能被判定为新术语。一方面，一支广告的集中投放，从间接可以说明其宣传的产品或理念会被大众接受，有成为新术语的可能；但另一方面，由于是同一支广告，是同一句话或一段话，在一段日期内的反复出现，这会使算法在“候选新术语发现”流程从这些广告中误选的候选新术语很容易被确定为新术语。通过表 8.4 可以看出以广告为来源的新术语准确率较低。

其次，目前新闻网站都会在新闻网页上提供网友评论新闻的功能，因此在每个新闻文本中，还会有大量的用户评论。这就会导致一些比较活跃的网友 ID 被误认为新术语，因为他们在发帖的时候，ID 会在每个帖子中出现。例如表 B.4

中的“linvir”就是一个活跃 ID。它们在频度曲线上的表现与新术语极其相似。

另外，算法返回的网友评论中的新术语较少，主要原因是一般网友评论只是对新闻主题的评论，其主要作用是提高新闻中出现新术语的频度，一般不会成为新术语的来源。

至于导航栏、链接的锚文本中的新术语，这主要由这些网站起引导作用。当某一个新事物将要或者已经成为讨论或关注的热点，网站就会开设一个子栏目，并在导航栏提供链接，如 2006 年 6 月 17 日的新术语 universal desktop。但是由于导航栏、链接的锚文本，一般都是比较短小的简写文字，它们一般不被认为是新术语，所以以这些作为来源的新术语的准确率较低。

Date/Name	TF/Line	DF/Details
2006-05-19	1	1
Read your Reader from anywhere	71	File...
2006-05-20	3	3
Google Music gets closer	73	File...
Google (service name here) isn't that pop...	74	File...
Turning the map Green	73	File...
2006-05-23	1	1
Measure Map getting closer#	71	File...
2006-05-24	1	1
Google video ads good# I think so	77	File...
2006-05-25	3	3
Calling International Rescue!	72	File...
Google Calendar RSS problems#	73	File...
Improving Software Usability#	82	File...
2006-05-26	19	5
Dell Installs Google Software at Factory	149	File...
Web 2.0 trademark tailspin	54	File...
Picasa for Linux	0 18 21 22 28 44 75	File...
Google Releases Picasa for Linux	65 68 136 207 208 248 259 269 319	File...
Dude, you got Google	74	File...
2006-05-27	7	4
Gmail to get prices	49 59 79	File...
Google's Picasa for Linux Arrives	8 10	File...
Dell wields its leverage and disrupts Wintel	59	File...
PowerPage wins on appeal	56	File...
2006-05-28	3	1
Google Video mix up	42 53 71	File...

图 8.1 “Picasa” 出现位置示意图

最后，对于以新闻正文作为来源的新术语，具有较高的准确率。主要原因是新闻正文不像广告一样，会在一段时间内都是同一段话的重复，如果能够在其中发现新术语，只有可能是这个新术语引起人们的关注，因而有较多的新闻对其进行报道，或使用这个新术语。如“Picasa”，这是 2006 年 6 月 19 日发现的新术语，它是 Google 开发的图片管理软件。在 2006 年 6 月 26 日发布了 Linux 版

本。其受关注的程度可以从以下两幅图示中看出。其中图 8.1 是“Picasa”出现位置示意图，图 8.2 是 Picasa 的频度曲线坐标图。从这两图可以明显看到，Picasa 的受关注程度在 6 月 26 日和 27 日达到最大，也即 Picasa 的 Linux 版本发布的当天和第二天。并且这两天的 Term Frequency 远大于 Document Frequency，从新闻标题也可以看出，是有不同的新闻一起报道 Picasa。

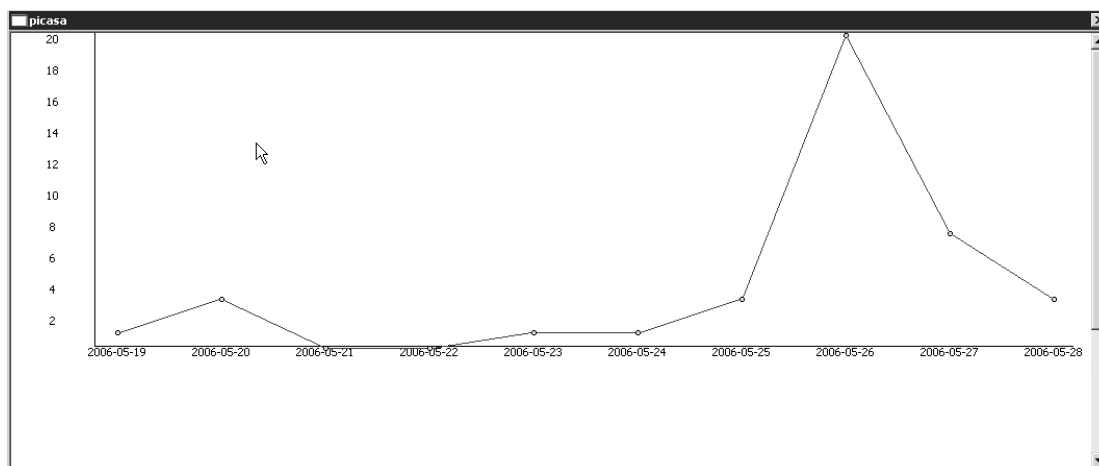


图 8.2 “Picasa”频度曲线坐标图

综上所述，可以看到，算法的准确率较低主要的原因有：第一，大量的广告词在一段时间内反复出现在新闻网页中，会对数据的可靠性造成一定的影响；第二，活跃用户的 ID 等也是造成算法准确率较低的重要原因。在下一章对存在问题的讨论中，会有对这些问题解决方案的探讨。

## 8.2 粒度和分析点数目对结果的影响

下面考查和分析一下参数“粒度”和“分析点数目”对算法返回结果的影响。

表 8.5 显示在 2006 年 5 月 17 日，粒度为 1，在不同分析点数目下算法返回新术语数目。可以看出，随着分析点数目的增长，算法返回新术语数目呈明显下降趋势。这并不意味着分析点数目越大，算法返回新术语准确率越高，例如当分析点数目为 19 的时候，唯一返回的新术语“yagu”是一个用户 ID。如前所述，真正的新术语，一般只会在发布的前后几天内保持较高的频度。表 B.6 是在 2006 年 5 月 17 日，粒度为 1，分析点数目为 14，算法返回的新术语列表，可供参考。



表 8.5 2006.05.17 不同分析点数目下算法返回新术语数目 (粒度=1)

分析点数目	返回新术语数目
5	17954
6	888
7	899
8	916
9	125
10	126
11	126
12	66
13	66
14	66
15	6
16	6
17	8
18	1
19	1

表 8.6 示在 2006 年 5 月 17 日, 分析点数目为 5, 在不同粒度下算法返回新术语数目。表 8.7 示在 2006 年 5 月 17 日, 分析点数目为 10, 在不同粒度下算法返回新术语数目。同样可以看出, 随着粒度的增长, 算法返回新术语数目明显下降。这与上面对不同分析点数目下的返回新术语数目趋势的解释相似。由于新术语的受关注时间只是很短的一段, 不适合使用较大的粒度进行检测。表 B.5 是在 2006 年 5 月 17 日, 粒度为 2, 分析点数目为 10, 算法返回的新术语列表, 可供参考。

表 8.6 2006.05.17 不同粒度下算法返回新术语数目 (分析点数目=5)

粒度	返回新术语数目
1	17954
2	7126
3	5326
4	4450
5	0
6	0

表 8.7 2006.05.17 不同粒度下算法返回新术语数目 (分析点数目= 10)

粒度	返回新术语数目
1	126
2	15
3	23

## 第九章 结论和展望

### 9.1 结论

总体来讲，目前为止，已经基本完成了任务书中所提的日程和任务。通过实验表明，该算法已经能够自动检测到相当的新术语。需要再一次说明的是，该算法实现只是“预警系统”，它目前能做到的只是为用户提供一个可能新术语的列表，甄别哪些是真正的新术语，哪些不是，仍然需要人工来确定。该算法只能做到极大程度地减小用户查找新术语的范围，降低用户的工作量。也正是出于这种考虑，该算法检测新术语的时候，只采取比较保守的手段，让所有可能的词串加入到该列表，这样可以保证算法的召回率，让用户放心地只需要在列表中甄别新术语即可，无需担心出现的新术语不被该列表收入。未来的工作也是要在保证较高召回率的前提下，不断的提高准确率，即进一步降低用户的查找新术语的工作。

### 9.2 存在问题

从实验的结果来看，存在许多问题需要进一步解决，他们主要是：

#### 9.2.1 广告问题

在第 8 章中已经对广告问题对算法产生的影响作了描述。如果能够更准确地处理广告中出现的新词串，可以极大提高算法的性能。

目前可以从两个方面分析如何解决这个问题：

1. 第一种方法，可以考虑在进行新术语发现之前，对新闻文本进行广告去除的工作。这样可以比较好的摒除广告中词语对算法的干扰。但这样做的主要问题是：
  - 首先，广告词也是新词语的一个重要的源泉，如果仅仅将广告部分删除，无疑会舍弃掉这一个重要来源，缩小算法挖掘和检测新术语的范围。

- 另外，对网页去除广告、导航栏，提取正文也是当前研究的热点，由于 HTML 的特点，尚无法做到非常准确地提取正文。
2. 第二种方法，是在评价函数方面对广告中出现的新词语进行一定的特殊处理。由于广告一般在一个网页中只会出现一次，因此在某日出现在广告中的词串的统计特点是其 Term Frequency 和 Document Frequency 相等，或者相差不大。目前的评价函数只是对 Term Frequency 的简单操作，未来可以对新词语的频度波动曲线进行更深入研究，进一步优化评价函数，从而消除广告对算法性能的不良影响。

### 9.2.2 新闻评论问题

如第 8 章中已经提到的，目前新闻网站都会提供用户发表对新闻评论的功能。这样就需要解决以下的问题：

1. 在网站中，用户的标识是他们的 ID，每当用户发表评论的时候，其 ID 总会出现在评论的标题中。而这些 ID 一般不是常用的单词，形式千奇百怪，因此那些活跃的用户，其 ID 由于频繁出现在新闻网页中，因此会被算法误认为新术语。该问题的可能解决办法如下为：
  - 由于这些活跃 ID 是非常有限的几个，因此当这些 ID 被算法误认为新术语的时候，会被加入到已发现术语列表记录下来，以后再次发现它的时候，也不会报告为新术语。也就是说算法具有一定的自适应能力，对于这些活跃 ID 只会错判一次。
  - 另外，由于评论的格式具有一定的规范，如回复的评论会有“Re:”的标识，在评论人的 ID 之前会有“by”，等等。因此可以通过正规表达式等来识别，如果发现格式如“Re: xxx by yyy”的时候，则将 yyy 认定为用户 ID 忽略掉。
2. 在用户的评论中，一方面由于反复提及新闻中的关键词，可以提高新术语的频度；但另一方面，网络交流中的非正式性也同时带来大量的新的缩写、简化等词汇的用法，当然这些里面也有一部分可能会成为新兴的网络用语而广泛运用，所以在未来需要对这部分词语进行更细致和长久的分析和考查。

### 9.3 未来工作

算法实现最终要交由用户使用，因此在实验完成之后，需要为用户提供一个高效、方便、可靠的图形化界面，降低操作难度。目前计划提供的图形化界面功能包括：

1. 提供对一些程序使用列表的文件路径、阈值等参数的设置；
2. 提供对某天、某粒度、某分析点数目下的新词语的发现，返回一个新术语及其评分的表格，并提供对某新术语的频度曲线图和查看所在位置的图形界面；
3. 允许用户选择算法返回新术语表格中的某些词串加入到已发现新术语列表中；
4. 允许用户查看已发现术语列表，并允许用户加入、修改和删除该列表表项；
5. 提供查看某一时间段内候选新术语的频度统计排名，作为最简单的统计数据呈现给用户；
6. 为了让用户可以每天自动接收到发现的新术语，提供定期自动检测新术语，并将结果每日通过 E-mail 方式发送给指定邮箱的功能。

另外目前该算法只是使用了 NLP 最基本的技术，也就是统计以及 Stemming 方法。这些方法虽然已经可以起到一定的效果，但是再加提高，则需要更多的 NLP 技术。未来工作中，将考虑使用词性标注、语法分析等技术来不断提高算法的效果。

一般而言，一个可以被认定为新词语的词串，如果它是单词，那么它可以是动词、名词以及缩写等之前并没有出现过的字母组合；如果它是词组，则只有可能是动词组或名词组，即它们的主干是一个动词或名词，而其他则为该单词的修饰成分。例如，“Data Mining”中“Data”修饰“Mining”，“E-business ERP Government Hardware”中“Hardware”是主干，而“cobble together”则以动词“cobble”为主干。换句话说，如果一个词串它的组合是“动词+形容词”或者“名词+代词”等组合，则该词串从语法角度上来讲，就不可能是一个词组，也就不可能是新术语。因此从更高层次来审查“候选新术语发现”得到的候选新术语，从语法角度删除或修整一些完全不符合语法规则的词串，可以很大程度上提高该算法的效果，能够将真正的新术语在返回结果中所占的比例提高。

当然，这样面临的问题是，“候选新术语发现”基本是通过统计的方式来发现候选新术语，这些词串有可能出现的情况是一个真正的新术语被一个累赘的单词粘在一起，如果加入“语法审查”部分，对这些词串仅仅删除了事，可能会漏掉相当数量的新术语，这是未来使用语法分析技术需要注意克服和解决的。

## 插图索引

图 2.1	基于 WEB 的计算机领域新术语自动检测算法流程图 . . . . .	8
图 4.1	IBM alphaWorks 提供RSS feeds的图示 . . . . .	17
图 4.2	RSS 阅读器图示 . . . . .	18
图 6.1	孤岛词串发现算法示意图 . . . . .	35
图 7.1	“Google” 在搜索引擎Google中被作为关键词查找的频度示意图 . .	47
图 7.2	“Web 2.0” 在搜索引擎Google中被作为关键词查找的频度示意图 . .	48
图 7.3	“Google Finance” 在发布前后的频度示意图 . . . . .	48
图 7.4	“Google Notebook” 在发布前后的频度示意图 . . . . .	49
图 7.5	“E-business ERP Government Hardware” 的频度示意图 . . . . .	49
图 7.6	5 月 17 日新术语列表 . . . . .	52
图 7.7	“Google Music” 频度曲线坐标图 . . . . .	53
图 7.8	“Google Music” 出现位置示意图 . . . . .	53
图 8.1	“Picasa” 出现位置示意图 . . . . .	57
图 8.2	“Picasa” 频度曲线坐标图 . . . . .	58





## 表格索引

表 3.1	五大英文 IT 新闻网站列表	11
表 3.2	IT 网站抓取历史网页数据量统计表	15
表 3.3	IT 网站历史网页获取纯文本数据量统计表	15
表 4.1	RSS 2.0 与 RSS 1.0 时间格式比较	24
4.2	IT RSS	26
表 5.1	Frequent Word List 片断 (前 10 个词)	30
表 5.2	通用语料库和背景语料库统计表	32
表 5.3	通用语料库、背景语料库二元词串表片断 (前10个二元词串)	33
表 5.4	通用语料库、背景语料库单词表片断 (前10个二元词串)	33
表 6.1	Google Stop Word List	36
表 8.1	算法返回新术语的准确率	55
表 8.2	算法返回新术语来源分类	56
表 8.3	人工确定新术语来源分类	56
表 8.4	各来源新术语的准确率	56
表 8.5	2006.05.17 不同分析点数目下算法返回新术语数目 (粒度=1)	59
表 8.6	2006.05.17 不同粒度下算法返回新术语数目 (分析点数目=5)	59
表 8.7	2006.05.17 不同粒度下算法返回新术语数目 (分析点数目=10)	60
表 B.1	2006.05.17 算法返回新术语列表 (粒度=1 分析点数目=10)	XXV
表 B.2	2006.05.18 算法返回新术语列表 (粒度=1 分析点数目=10)	XXVIII
表 B.3	2006.05.19 算法返回新术语列表 (粒度=1 分析点数目=10)	XXX
表 B.4	2006.05.20 算法返回新术语列表 (粒度=1 分析点数目=10)	XXXI
表 B.5	2006.05.17 算法返回新术语列表 (粒度=2 分析点数目=10)	XXXVII
表 B.6	2006.05.17 算法返回新术语列表 (粒度=1 分析点数目=14)	XXXVII
表 B.7	2006.05.17-2006.05.20 人工确定新术语列表	XXXIX



## 公式索引

公式 1	.....	38
公式 2	.....	38
公式 3	.....	50
公式 4	.....	50
公式 5	.....	51



## 参考文献

- [1] The Barnhart Dictionary Companion's Website Homepage. Technical report, 2006. <http://lexikhouse.com/>
- [2] 汪榕培. 英语词汇学教程. 上海: 上海外语教育出版社, March, 1997
- [3] 张德鑫. “水至清则无鱼”——我的新生词语规范观. 北京大学学报(哲学社会科学版), 2000, 37:106-119
- [4] 罗剑平. IT 业英语新词汇的分类及其构成. 吉首大学学报(社会科学版), 2001, 21:100-103
- [5] 邹纲. 面向 Internet 的中文新词语检测. 中文信息学报, 2004, 18:1-9
- [6] 郑家恒. 基于构词法的网络新词自动识别初探. 山西大学学报(自然科学版), 2002, 25:115-119
- [7] 李保利. 信息抽取研究综述. 计算机工程与应用, 2003, 10:1-5
- [8] Grishman R. Information Extraction: Techniques and Challenges. Proceedings of SCIE '97: International Summer School on Information Extraction, London, UK: Springer-Verlag, 1997. 10-27
- [9] Wikipedia. Information extraction — Wikipedia, 2006. [https://secure.wikimedia.org/wikipedia/en/wiki/Information\\_extraction](https://secure.wikimedia.org/wikipedia/en/wiki/Information_extraction). [Online; accessed 5-May-2006]
- [10] Turney P D. Extraction of Keyphrases from Text: Evaluation of Four Algorithms. Technical report, October 23, 1997. [http://www.apperceptual.com/ml\\_text\\_keys.html](http://www.apperceptual.com/ml_text_keys.html)
- [11] Turney P D. Learning to Extract Keyphrases from Text. Technical report, February 17, 1999. [http://www.apperceptual.com/ml\\_text\\_keys.html](http://www.apperceptual.com/ml_text_keys.html)
- [12] Ong T H, Chen H. Updateable PAT-Tree Approach to Chinese Key Phrase Extraction using Mutual Information: A Linguistic Foundation for Knowledge Management. Proceedings of the Second Asian Digital Library Conference, 1999
- [13] 亢世勇. 《新词语大词典》前言. 2003.
- [14] 高永伟. 近20年英语国家对新词的研究. 外语与外语教学(大连外国语学院学报), 1998, 114:8-10
- [15] 刘叔新. 汉语描写词汇. 北京: 商务印书馆, 1990
- [16] Lewin J. RSS 2.0内容提要. Technical report, January, 2004. <http://www-128.ibm.com/developerworks/cn/xml/x-rss20/>
- [17] DoNews. RSS — DoNews IT 百科全书, 2006. <http://wiki.csdn.net/index.php/RSS>. [Online; accessed 14-May-2006]
- [18] Pilgrim M. What Is RSS. Technical report, December, 2002. <http://www.xml.com/pub/a/2002/12/18/dive-into-xml.html>

- [19] Wikipedia. 网志— Wikipedia., 2006. <https://secure.wikimedia.org/wikipedia/zh/wiki/Blog>. [Online; accessed 13-May-2006]
- [20] DoNews. Blog — DoNews IT 百科全书, 2006. <http://wiki.csdn.net/index.php/Blog>. [Online; accessed 14-May-2006]
- [21] Wikipedia. Wiki — Wikipedia, 2006. <https://secure.wikimedia.org/wikipedia/zh/wiki/Wiki>. [Online; accessed 13-May-2006]
- [22] Winer D. RSS 2.0 Specification. Technical report, January, 2005. <http://blogs.law.harvard.edu/tech/rss>
- [23] Begeed-Dov G, LLC J S, Brickley D, et al. RSS 1.0 Specification. Technical report, May, 2001. <http://web.resource.org/rss/1.0/spec>
- [24] RSS Tutorial for Content Publishers and Webmasters. Technical report, September, 2005. <http://www.mnot.net/rss/tutorial/>
- [25] Crocker D H. Standard for ARPA Internet Text Messages. Technical report, August, 1982. <http://www.ietf.org/rfc/rfc0822.txt>
- [26] NOTE-datetime. Technical report, August, 1998. <http://www.w3.org/TR/NOTE-datetime>
- [27] java rss lib 评测. Technical report, February, 2005. <http://blog.csdn.net/zhaozixin/archive/2005/02/06/282333.aspx>
- [28] Daniel Jurafsky, James H. Martin, 冯志伟、孙乐译. 自然语言处理综论 (Speech and Language Processing — An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition) . 2005
- [29] Lovins J B. Development of a Stemming Algorithm. Mechanical translation and computational linguistics, 1968, 11:22–31
- [30] Kantrowitz M, Mohit B, Mittal V. Stemming and its effects on TFIDF ranking (poster session). Proceedings of SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA: ACM Press, 2000. 357-359
- [31] Porter M. Snowball: A language for stemming algorithms. Technical report, October, 2001. <http://www.snowball.tartarus.org/texts/introduction.html>
- [32] Porter M. The Porter Stemming Algorithm. Technical report, January, 2006. <http://www.tartarus.org/martin/PorterStemmer/index.html>
- [33] XML 认证教程, 第 1 部分: XML 简介. Technical report, March, 2003. <http://www-128.ibm.com/developerworks/cn/xml/x-cert/part1/>
- [34] Vohra D. 使用 dom4j 解析 XML. Technical report, April, 2004. <http://www-128.ibm.com/developerworks/cn/xml/x-dom4j.html>
- [35] Quick start of dom4j. Technical report, May, 2005. <http://www.dom4j.org/guide.html>
- [36] Porter M. An Algorithm for Suffix Stripping. Program, 1980, 14:130–137

- [37] Wikipedia. N-gram — Wikipedia, 2006. <https://secure.wikimedia.org/wikipedia/en/wiki/N-gram>. [Online; accessed 5-May-2006]
- [38] Wikipedia. Bigram — Wikipedia, 2006. <https://secure.wikimedia.org/wikipedia/en/wiki/Bigram>. [Online; accessed 5-May-2006]
- [39] Zhang Y, ZinciHeywood N, Milios E. Narrative Text Classification for Automatic Key Phrase Extraction in Web Document Corpora. Proceedings of the 7th annual ACM international workshop on Web information and data management, Bremen, Germany, 2005. 51-58





## 致 谢

首先感谢导师孙茂松教授对我的悉心指导，孙老师谨严的治学态度，循循善诱的教导，对学生严格的要求，让我大受裨益。

另外也感谢研究组的苑春法、陈群秀和周强老师，在开题报告和中期答辩中都给我以宝贵的意见和建议，帮助我改进和完善毕业设计的思路。

同时感谢研究组的学长，是他们营造了一个非常融洽的学术讨论气氛，在进行课题钻研的时候，给我以耐心的指点和启发，使我在研究过程中能够顺利解决遇到的每一个问题。

最后感谢父母对我的爱护和教育，感谢同学对我的支持和帮助。



## 声 明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签 名：刘知远 日 期：2006.6.20



## 附录 A 外文资料调研阅读报告

### A.1 Information Extraction

The following are summarized from articles [8,9].

#### A.1.1 Definition

Information extraction(IE) is a type of information retrieval whose goal is to automatically extract structured or semi-structured information from unstructured machine-readable documents. It is a sub-discipline of language engineering, a branch of computer science.

IE's narrower definition is: the identification of instances of a particular class of events or relationships in a natural language text, and the extraction of the relevant arguments of the event or relationship. IE therefore involves the creation of a structured representation(such as a data base) of selected information drawn from the text.

The significance of IE is determined by the growing amount of information available in unstructured form, for instance on the Internet. This knowledge can be made more accessible by means of transformation into relational form.

Typical subtasks of IE are:

- Named Entity Recognition: recognition of entity names (for people and organizations), place names, temporal expressions, and certain types of numerical expressions.
- Co-reference: identification chains of noun phrases that refer to the same object. For example, anaphora is a type of co-reference.

#### A.1.2 The Overall Flow

The process of information extraction has two major parts. First, the system extracts individual “facts” from the text of a document through local text analysis. Second, it integrates these facts, producing larger facts or new facts(through inference).

As a final step after the facts are integrated, the pertinent facts are translated into the required output format.

The individual facts are extracted by creating a set of patterns to match the possible linguistic realizations of the facts. Since the complexity of natural language, in most natural language processing systems, we begin by formatting the input, identifying various levels of constituents and relations, and then state our patterns in terms of these constituents and relations. The process begins with lexical analysis(assigning part-of-speech, etc.) and name recognition(identifying names and other special lexical structures such as dates, currency expressions, etc.). Then a full syntactic analysis(parse) is followed. And in most current systems, it is replaced by some form of partial parsing to identify noun groups, verb groups, and possibly head-complement structures. After all this is done, we can use task-specific patterns to identify the facts of interest.

### A.1.3 Pattern Matching and Structure Building

In most current extraction system, most of the text analysis is performed by matching the text against a set of regular expressions. If the expression matches a segment of text, the text segment is assigned a label, and possibly associated features. Associated with some of the labels in system are semantic structures called *entities* and *events*. These structures will be used to construct the templates.

### A.1.4 Lexical Analysis

The text is first divided into sentences and tokens. Each token is looked up in the dictionary to determine its possible part-of-speech and features. For example, 'Sam' and 'Harry' would be tagged as first names; Inc. will be tagged as a company suffix.

### A.1.5 Name Recognition

The next phase of processing identifies various types of proper names and other special forms, such as dates and currency amounts. Because names appear frequently in many kinds of texts, identifying and classifying them will simplify further processing. Names are important as argument value for many extraction tasks.

Names are usually identified by a set of regular expressions which are stated in terms of part-of-speech, syntactic features, and orthographic features.

Personal names might be identified by a preceding title, by a common first name, by a suffix or by a middle initial.

Company names can usually be identified by their final tokens. However, some major company names may be mentioned without any such overt clues, so it is important to also have a dictionary of major companies.

Name identification typically also includes the processing to identify aliases of a name — name co-reference. For example, a system would identify “the Hewlett-Packard Corp.” with “HP”.

#### A.1.6 Syntactic Structure

Because the arguments to be extracted often correspond to noun phrases in the text, and the relationships to be extracted often correspond to grammatical functional relations, identifying some aspects of syntactic structure will simplify the subsequent phase of fact extraction.

For the identification of the complete syntactic structure of a sentence is a difficult task, most systems build a series of parse fragments. In general, they only build structure about which they are quite certain, either from syntactic or semantic evidence. For each noun group the system will create a “semantic” *entity*.

#### A.1.7 Scenario Pattern Matching

All the processing until now has been ready for the scenario pattern matching. The role of these patterns is to extract the vents or relationships relevant to the scenario. For example, the followings are two patterns,

*person* retires as *position*

and

*person* is succeeded by *position*

where *person* and *position* are pattern elements which match noun groups. “retires” and “is succeeded” are pattern elements which match active and passive verb groups.

#### A.1.8 Co-reference Analysis

Co-reference analysis’s task is to resolve anaphoric references by pronouns and definite noun phrases. For example, in a text there is a “he”, co-reference analysis will look for the most recent previously mentioned entity of type *person*, and will find entity, e.g.  $e_1$ . It will then change references to  $e_1$ .

#### A.1.9 Inferencing and Event Merging

In many situations, partial information about an event may be spread by spread over several sentences. The information needs to be combined. In other cases, some of the information may be only implicit, and need to be made explicit through an inference process. These can be done by production system rules.

### A.2 Phrase Extraction

The following are summarized from articles [12].

Phrase extraction, commonly called word segmentation, for the Chinese language means finding the longest phrase in a word string with precise meaning. This is considered the major barrier to text retrieval, especially for Asian languages.

Key phrase extraction, commonly known as indexing, finding the phrase that are representative of a document. Indexing is fundamental to the success of many recent digital library applications and semantic information retrieval techniques.

Prior research in phrase extraction can be categorized into three categories.

#### A.2.1 Dictionary approach

This approach uses a human-generated dictionary to break out known phrase in the dictionary that have more than one character. The major advantage is quick and simple implementation. A lot of system are still using this approach, especially when



words are not a concern or the dictionary covers the target documents. Because there usually is a partial matching dilemma, several strategies have been devised to improve the matching process, including maximum-matching, minimum-matching, forward-matching, backward matching, bidirectional, and other heuristics. However, the effectiveness of this approach is largely limited by the comprehensiveness of the dictionary, which cannot effectively deal with proper nouns such as names and places and new terms constantly being created in special domains.

### A.2.2 Linguistic approach

An alternative approach uses syntactic and/or semantic knowledge bases, heuristics, or rules to extract phrases. From the natural language processing research in English, it has demonstrated to be able to achieve high precision, depending on the system design. However, although a number of articles have suggested the possibility, there has been no extensive computational linguistic research for the Chinese language, and implementing this approach remains a difficult task. The major drawback is that building complete syntactic and semantic knowledge bases for large and complex domains has proved daunting.

### A.2.3 Statistical approach

This approach generally learns valuable statistical information from a usually large corpus to extract possible phrase. This approach was shown to generate good performance in extracting key phrases. It has a traditional tie to the n-gram approach with a small number of n, such as 2, 3 or 4. Despite its extensive computation, the technique does not require any labor-intensive creation of dictionary or knowledge base and is capable of capturing emerging terminology in the corpus. One major drawback is that it is not able to extract valid phrases that does not occur sufficiently frequently.

Although perhaps not as precise as the linguistic approach, the statistical approach is still a good choice for extracting key phrases. Since it is based on the probable occurrences in the collection, low-frequency phrases, which are more likely not to be key phrase, will be removed automatically. Therefore, a statistical approach is very well suited for extracting key phrases.

There are some challenges for the statistical approach. Kenneth Church has pointed out that some Bible literature has repeated patterns with up to 400 words and constitutes a challenge for the n-gram technique because, without removal of the pattern every sub-pattern in the 400-word sentence could be extracted. Removal of a pattern from the corpus affects the frequency distribution of the corpus, especially repetitive removals of many sub-patterns. An attempt to figure out the new frequency without updating was tried but failed. Different researchers have tried different heuristics to extract better phrases and avoid partial phrases.

### A.3 Stemming

The following are summarized from articles [10,11,29–32].

High precision IR is often hard for a variety of reasons; one of these is the large number of morphological variants for any given term. To address some of the issues arising from a mismatch between different word forms used in the queries and the relevant documents, researchers have long proposed the use of various stemming algorithms to reduce terms to a common base form.

“A stemming algorithm is a computational procedure which reduces all words with the same root (or, if prefixes are left untouched, the same stem) to a common form” [29] In essence, a stemmer operates by matching the ending of a word against a suffix dictionary, removing any suffix that is identified, and checking whether any context-sensitive rules apply.

Stemming is a little different from lemmatization. Lemmatization takes all inflectionally related forms of a word and groups them together under a single lemma. For example, be, been, was, were and is can be achieved by lemmatization.

In poster [30], their experiments with data sets of varying document and query sizes found that improvements in stemming accuracy yielded corresponding improvements in retrieval precision with short queries. At its best performance level, the dictionary based stemmer yielded significant improvements in precision on short queries. The almost linear improvement in performance for  $t_i$  with increasing coverage suggests that further improvements in TF\*IDF ranking for IR may be possible by including specialized, domain specific lexicons in the stem-classes.

The Porter(1980) and Lovins(1968) stemming algorithms are the two popular algorithms for stemming English words. Both algorithms use heuristic rules to remove or transform English suffixes. Another approach to stemming is to use a dictionary that explicitly lists the stem for every word that might be encountered in the given text. Heuristics are usually preferred to a dictionary, due to the labor involved in constructing the dictionary and the computer resources(storage space and execution time) required to use the dictionary.

The Lovins stemmer is more aggressive than the Porter stemmer. That is, the Lovins stemmer is more likely to map two words to the same stem, but it is also more likely to make mistakes. For example, the Lovins stemmer correctly maps “psychology” and “psychologist” to the same stem, “psycholog”, but the Porter stemmer incorrectly maps them to different stems, “psychologi” and “psychologist”. On the other hand, the Porter stemmer correctly maps “police” and “policy” to different stems, “polic” and “polici”, but the Lovins stemmer incorrectly maps them to the same stem, “polic”.

The original stemming algorithm paper was written in 1979 in the Computer Laboratory, Cambridge(England), as part of a larger IR project. The Porter stemmer should be regarded as “frozen”, that is, strictly defined, and not amenable to further modification. As a stemmer, it is slightly inferior to the Snowball English or Porter2 stemmer, which derives from it, and which is subjected to occasional improvements.

Snowball is a small string processing language designed for creating stemming algorithms for use in Information Retrieval. It is “Snowball” named as a tribute to SNOBOL, the excellent string handling language.

There are two main reasons for creating Snowball. One is the lack of readily available stemming algorithms for languages other than English. The other is the consciousness of a certain failure on Porter’s part in promoting exact implementation of the stemming algorithm described in [36], which has come to be called the Porter stemming algorithm.

#### A.4 N-gram and Bi-gram

The following are summarized from articles [37,38].

An N-gram is a sub-sequence of  $n$  items from a given sequence. This idea can be traced to an experiment by Claude Shannon's work in information theory. His idea was that given a sequence of letters, (for example, the sequence "for ex") what is the likelihood of the next letter? From training data, you would derive a probability distribution for the next letter given a history of size  $n$ :  $a = 0.4, b = 0.00001, c = 0 \dots$ ; where the probabilities of all possible "next-letters" sums to 1.0.

More concisely, an  $n$ -gram is a model of predicting  $x_i$  based on  $x_{i-1}, x_{i-2}, \dots, x_{i-n}$ . In application to language modeling, because of computational limitations and the open nature of language (there are infinite possible words), independence assumptions are made so that each word depends only on the last  $n$  words, making it a good Markov model.

An  $n$ -gram of size 1 is a "unigram"; size 2 is a "bi-gram" (or, more appropriately but less commonly, a digram); size 3 is a "trigram"; and size 4 or more is simply called an "n-gram" or "n-1 order Markov Model".

N-grams are a popular technique in statistical natural language processing. In speech recognition, phonemes are modeled using a  $n$ -gram distribution. For parsing, words are modeled such that each  $n$ -gram is composed of  $n$  words. For a sequence of words, (for example "the dog smelled like a skunk"), the trigrams would be: "the dog smelled", "dog smelled like", "smelled like a", and "like a skunk". For sequences of characters, the trigrams that can be generated from "good morning" are "goo", "ood", "od", "d m", "mo", and so forth. Some practitioners preprocess strings to remove spaces, others do not. In almost all cases, punctuation is removed by preprocessing. N-grams can also be used for sequences of words or, in fact, for almost any type of data. They have been used for example for extracting features for clustering large sets of satellite earth images and for determining what part of the Earth a particular image came from.

Bi-gram are groups of two letters, two syllables, or two words, and are very commonly used as the basis for simple statistical analysis of text; one of the most successful language models for speech recognition. They are a type of N-gram.

Bi-grams help provide the conditional probability of a word given the preceding

word, when Bayes' theorem is applied:

$$P(W_n|W_{n-1}) = \frac{P(W_{n-1}, W_n)}{P(W_{n-1})}$$

That is, the probability  $P()$  of a word  $W_n$  given the preceding word  $W_{n-1}$  is equal to the probability of their bi-gram, or the co-occurrence of the two words  $P(W_{n-1}, W_n)$ , divided by the probability of the preceding word.

## A.5 TF\*IDF Method

The following are summarized from articles [39].

TF\*IDF is a standard keyword identification method in information retrieval tasks. It gives preference to words that have high frequency of occurrence in a single document but rarely appear in the whole document collection. This involves in the following steps:

1. For each document, convert it to lower case.
2. Extract all tokens in the document, i.e., identify single words by removing punctuation marks and numbers. A standard set of stop words(*a, about, above, ...*) is discarded at this stage.
3. Apply stemming to obtain word stems and update the number of documents in which each word stem appears.
4. Once all documents are processed using the above three steps, calculate the TF\*IDF value  $w_{i,j}$  of word stem  $i$  in document  $j$  using the following equation:

$$w_{i,j} = \frac{n_{i,j}}{|d_j|} \cdot \log_2 \frac{N}{n_i}$$

where  $n_{i,j}$  is the frequency of word stem  $i$  in document  $j$ ,  $|d_j|$  is the number of word stems in document  $j$ ,  $n_i$  is the number of pages that contain word stem  $i$ , and  $N$  is the total number of documents in consideration.

5. For each document  $j$ , TF\*IDF values of all word stems in this page are normalized to unit length 1.0 as follows:

$$W_{i,j} = \frac{w_{i,j}}{\sqrt{\sum_i w_{i,j}^2}}$$

6. Finally, choose the top five word stems ranked by normalized TF\*IDF values for each document. The number 5 is chosen based on the fact that often 3 to 5 key phrases are included in a technical article.

## 附录 B 实验数据列表

表 B.1 2006.05.17 算法返回新术语列表（粒度= 1 分析点数目= 10）

排名	新术语	分数
1	rayovac chooses windows	5.5
2	risk microsoft	5.5
3	coverage & capacity motorola	5.0
4	lan work	5.0
5	yagu	5.0
6	uh	5.0
7	rob t firefly	4.5
8	technicon	4.5
9	tackhead	4.5
10	tryout	4.0
11	digital micro-markets	4.0
12	new york-june	4.0
13	featured event	4.0
14	online newsletters	4.0
15	new backbone	4.0
16	universal desktop	4.0
17	google music	4.0
18	structured and unstructured	4.0
19	right web	4.0
20	gosling	4.0
21	business coummunications	4.0
22	sunrocket voip	4.0
23	expensive microsoft	4.0
24	citrix gotomypc corporate	4.0
25	wearable tech	4.0
26	order reprints	4.0
27	on stylish asus notebooks	4.0
28	business oracle	4.0
29	newsletters help	4.0
30	xt / eos	4.0

续下页

续表 B.1 2006.05.17 算法返回新术语列表 (粒度=1 分析点数目=10)

31	interactive technology	4.0
32	digg	4.0
33	cost-effective remote	4.0
34	preferred e-mail	4.0
35	aligned	4.0
36	duh	4.0
37	firms use web-based quickbase	4.0
38	managing through workplace disruption	4.0
39	edt/11 a	4.0
40	cingular cell	4.0
41	click below	4.0
42	optimizing data	4.0
43	proliant dl385	4.0
44	winhec	4.0
45	enterprise anti-matter	4.0
46	eweek editorial	4.0
47	reworking	4.0
48	real-time access	4.0
49	adaptec	4.0
50	ab bc mb nb nf ns nt on pe qc sk yt	4.0
51	free phones	4.0
52	business priorities	4.0
53	web buyers guide	4.0
54	chicago-june	4.0
55	calendar features	4.0
56	need robust data	4.0
57	unexpected workplace disruption	4.0
58	davis eseminar update	4.0
59	magazine's free	4.0
60	san francisco-june	4.0
61	pcmagcasts	4.0
62	own website	4.0
63	trialware	4.0
64	subscriber help	4.0
65	free pcmagcasts	4.0
66	ae ap	4.0

续下页



续表 B.1 2006.05.17 算法返回新术语列表 (粒度=1 分析点数目=10)

67	oracle one	3.5
68	fuse your	3.5
69	businesses oracle	3.5
70	security microsoft	3.5
71	macworld podcasts weblog	3.5
72	solutions catalogue	3.5
73	patch mgmt	3.5
74	etech	3.0
75	xxx domain	3.0
76	geoffspear	3.0
77	critical strategies	3.0
78	leveraging existing	3.0
79	up collaboration	3.0
80	find video	3.0
81	supercomputing	3.0
82	mgt	3.0
83	onhollywood	3.0
84	dragonwriter	3.0
85	a sme business oracle	3.0
86	baseline briefing	3.0
87	non-profit hospital	3.0
88	iprism provides cost-effective internet	3.0
89	eweek wireless update	3.0
90	infrastructure modernization	3.0
91	hp on	3.0
92	steven j	3.0
93	overkill	3.0
94	editor larry seltzer's weblog	3.0
95	storage rss	3.0
96	nat breaks voip	3.0
97	that??s	3.0
98	next>>	3.0
99	vux984	3.0
100	com storage	3.0
101	erp survival guide zdnet	3.0
102	storage initiatives	3.0

续下页

续表 B.1 2006.05.17 算法返回新术语列表 (粒度= 1 分析点数目= 10)

103	online event	3.0
104	flamebait	3.0
105	oracleas portal oracle	3.0
106	report casts doubt on vista's security	3.0
107	product overview	3.0
108	cio alert	3.0
109	blogspot	3.0
110	simplify storage	3.0
111	net/	3.0
112	minus caffeine	3.0
113	mumbai	3.0
114	goals w/ ca	3.0
115	sybase is enabling customer	3.0
116	cnn	3.0
117	downloadable white	3.0
118	yayagu@gmailcola	3.0
119	data archiving	3.0
120	saleens281	3.0
121	vaughan-nichols	3.0
122	webroot enterprise spy audit webroot	3.0
123	com's security	3.0
124	sme issues	3.0
125	network admins	3.0
126	baseline tools	3.0

表 B.2 2006.05.18 算法返回新术语列表 (粒度= 1 分析点数目= 10)

排名	新术语	分数
1	that??s	4.0
2	download trial	3.5
3	znet webcasts	3.5
4	whiney mac fanboy	3.5
5	new bonus packs	3.5
6	video resources	3.5
7	chrysler mercedes benz na cio drives consolidation	3.5
8	troll	3.5

续下页

续表 B.2 2006.05.18 算法返回新术语列表 (粒度=1 分析点数目=10)

9	googlepages	3.5
10	issue archive	3.0
11	cbs radio	3.0
12	com rss feeds	3.0
13	year-ahead planner	3.0
14	ws/	3.0
15	users wanting	3.0
16	chair ducker	3.0
17	products>	3.0
18	year ahead planner	3.0
19	watch premium	3.0
20	transfer rates-speed	3.0
21	boycott	3.0
22	vista-capable	3.0
23	state pastor	3.0
24	gee	3.0
25	former prez clinton to become microsoft	3.0
26	once ballmer retires	3.0
27	across domains	3.0
28	networks motorola	3.0
29	still coding at 4 am	3.0
30	money pit	3.0
31	prototyping in	3.0
32	mobiletatsu-njg	3.0
33	content login	3.0
34	add microsoft	3.0
35	links work	3.0
36	serial attached scsi	3.0
37	microsoft storage	3.0
38	who's afraid	3.0
39	devsource interviews usability pundit dr	3.0
40	business resilience	3.0
41	microsoft's actions	3.0
42	nielsen on	3.0
43	linux challenger to sharepoint champion	3.0
44	foley	3.0

续下页

续表 B.2 2006.05.18 算法返回新术语列表（粒度= 1 分析点数目= 10）

45	there's mactel	3.0
46	bad hutch	3.0
47	khasim	3.0
48	its perceived pro-gay-rights stance	3.0
49	ms blogger links	3.0
50	michael_t	3.0
51	redmond software	3.0
52	user-access-protection feature	3.0
53	vista too protective	3.0
54	eln	3.0
55	pdf need	3.0
56	lucovsky ponders	3.0
57	ongoing cultural	3.0
58	name tracker	3.0
59	dell needs	3.0
60	seamless mobility	3.0
61	proactive measures for forward-looking enterprises ibm	3.0
62	designing apps	3.0
63	duet	3.0
64	fremont	3.0
65	wii	3.0
66	microsoft reversing the brain-drain tide	3.0
67	microsoft's planning	3.0
68	mini-microsoft blog	3.0
69	internet stories	3.0
70	prolific webbloggers	3.0
71	mobile messaging	3.0

表 B.3 2006.05.19 算法返回新术语列表（粒度= 1 分析点数目= 10）

排名	新术语	分数
1	gps car	4.0
2	zoom ex-z850	4.0
3	car video	4.0
4	eidewt	4.0

续下页

续表 B.3 2006.05.19 算法返回新术语列表 (粒度=1 分析点数目=10)

5	desktops or notebooks	4.0
6	proliant dl385 server	3.5
7	drives offers dual	3.5
8	free 14 day	3.5
9	exploiting google	3.0
10	picasa	3.0
11	replacing tape livevault	3.0
12	cyber crime	3.0
13	requests halt	3.0
14	eweek storage	3.0
15	get broadbandconnect from cingular	3.0
16	performance monitoring	3.0
17	that??s	3.0
18	eweek news	3.0
19	protecting backup media with aes encryption emc	3.0
20	samkass	3.0
21	security reviews	3.0
22	security rss	3.0
23	neutrality field	3.0
24	reasons why online	3.0
25	sobig virus	3.0
26	latitude d620 and d820	3.0
27	big google talk fallacy in hitwise's top 20 google domains meme	3.0
28	pixma mp830 office-all-in-one printer	3.0
29	whineymacfanboy@gmail	3.0
30	vista development	3.0
31	new job>>	3.0
32	integrating compliance	3.0
33	piracy & counterfeiting	3.0
34	manipulate large	3.0
35	get broadbandconnect	3.0

表 B.4 2006.05.20 算法返回新术语列表 (粒度=1 分析点数目=10)

排名	新术语	分数
----	-----	----

续下页

续表 B.4 2006.05.20 算法返回新术语列表 (粒度=1 分析点数目=10)

1	notebooks opinions	3.5
2	notebook products	3.5
3	open-source desktops	3.5
4	find desktop	3.5
5	hp's image	3.5
6	notebooks coverage	3.5
7	ibm's evolution	3.5
8	notebooks news	3.5
9	notebooks reviews	3.5
10	add eweek desktops	3.5
11	notebooks analysis	3.5
12	lenovo j105 desktop	3.0
13	computer expense	3.0
14	nw200 new	3.0
15	cavalier homes	3.0
16	windows embedded	3.0
17	resellers expect allfusion r7	3.0
18	michael cherry	3.0
19	insider information	3.0
20	fbi agent recounts outsourcing horror	3.0
21	recovery livevault	3.0
22	handy reference	3.0
23	covad voip save	3.0
24	tech analysis	3.0
25	video>>	3.0
26	innovative application	3.0
27	try microsoft	3.0
28	networkworld inc	3.0
29	microsoft reseller	3.0
30	vernier networks steps	3.0
31	quizzes	3.0
32	ipswitch ws_ftp server	3.0
33	emerson business	3.0
34	introducing intel	3.0
35	accelerating your	3.0
36	partner recruitment	3.0

续下页

续表 B.4 2006.05.20 算法返回新术语列表 (粒度=1 分析点数目=10)

37	free on-line courses	3.0
38	servers & data	3.0
39	malware becoming	3.0
40	customized software versus build-from-scratch	3.0
41	around topics	3.0
42	linvir	3.0
43	enterprise online event	3.0
44	nw subscription	3.0
45	bonus packs	3.0
46	ibm puts up	3.0
47	software distributor	3.0
48	want & get	3.0
49	get certified	3.0
50	renew today	3.0
51	e-mail article	3.0
52	criminals	3.0
53	vulnerability management	3.0
54	turnkey integration	3.0
55	work newsletter today	3.0
56	ibm partner consolidates music retailer view	3.0
57	gilroy takes	3.0
58	two-way web	3.0
59	online guide	3.0
60	csc	3.0
61	map control	3.0
62	updated monthly	3.0
63	nominations	3.0
64	google grabs	3.0
65	drive exchange-lotus migrations view	3.0
66	tests/ buyer's guides	3.0
67	modern-day mba	3.0
68	one gui	3.0
69	reporting piracy works	3.0
70	ant p	3.0
71	opsource	3.0
72	try digital eweek	3.0

续下页

续表 B.4 2006.05.20 算法返回新术语列表 (粒度=1 分析点数目=10)

73	free podcast downloads	3.0
74	mti	3.0
75	upcoming conference	3.0
76	channel business	3.0
77	unified threat	3.0
78	vendor solutions	3.0
79	quantify security	3.0
80	pa pr ri	3.0
81	infrastructure requirements download	3.0
82	downloadable roi calculators & tools from baseline	3.0
83	upcoming ms	3.0
84	verisign ssl certificates	3.0
85	seven podcasts	3.0
86	order configuration	3.0
87	audit defensible it compliance controls	3.0
88	ip migration	3.0
89	big flops	3.0
90	introduce technology	3.0
91	links buy	3.0
92	year-ahead calendar	3.0
93	let bonus packs help	3.0
94	break-even point	3.0
95	in sarbanes-oxley	3.0
96	eds win outsourcing extensions	3.0
97	ibm's john guido	3.0
98	security expertise drives i	3.0
99	expertise for web environments	3.0
100	pricing guidelines	3.0
101	sales objection handling	3.0
102	developer resource	3.0
103	microsoft allows	3.0
104	cobble together	3.0
105	guesswork	3.0
106	microsoft empower for isvs rewards your	3.0
107	upcoming eseminars	3.0
108	signature series	3.0

续下页



续表 B.4 2006.05.20 算法返回新术语列表 (粒度=1 分析点数目=10)

109	adopt soa	3.0
110	reliable ip telephony systems	3.0
111	patch & spyware management	3.0
112	scalable hosting solutions	3.0
113	attention microsoft solution	3.0
114	proactive customer support deliver secure	3.0
115	adaptec adaptec sata ii raid outperforms lsi	3.0
116	insider buyer's	3.0
117	hardware tools including provisioning	3.0
118	hosting solutions	3.0
119	calculating an outsourcing ratio	3.0
120	voip & convergence	3.0
121	sbs 2003 sales	3.0
122	adoption strategies	3.0
123	security forums	3.0
124	featured calculators	3.0
125	automated discovery	3.0
126	enterprise all-star	3.0
127	everio gz-mg37	3.0
128	exponential rate	3.0
129	built-in and routinely updated control mappings	3.0
130	rackspace provides unrivaled levels	3.0
131	frank derlfer	3.0
132	linux ibm	3.0
133	onforce	3.0
134	idc white	3.0
135	networkworld	3.0
136	channel resellers	3.0
137	catalog publishing	3.0
138	fill-in form	3.0
139	mopping up	3.0
140	cdw	3.0
141	hidden dangers	3.0
142	ct de dc fl ga gu hi id	3.0
143	microsoft resource	3.0
144	measuring roi	3.0

续下页

续表 B.4 2006.05.20 算法返回新术语列表（粒度= 1 分析点数目= 10）

145	join amd	3.0
146	video library	3.0
147	oracle soa leadership roundtable podcast	3.0
148	free hands-on training lab	3.0
149	blue's avnet relationship	3.0
150	calculate cost and roi	3.0
151	ingram micro	3.0
152	blended threats	3.0
153	join elliot markowitz	3.0
154	eweek career	3.0
155	microsoft empower program	3.0
156	procurement outsourcing gains momentum	3.0
157	around soa	3.0
158	protection gap	3.0
159	licensing requirements	3.0
160	today's 10 most-read stories	3.0
161	efficiencies	3.0
162	on-demand eseminars	3.0
163	unitrends	3.0
164	on consolidating data	3.0
165	ut vt vi va wa wv wi wy	3.0
166	primer	3.0
167	resellers	3.0
168	software publishers	3.0
169	business buyer's	3.0
170	reprint	3.0
171	unique cisco-powered zero-downtime network	3.0
172	icio	3.0
173	ibm provides	3.0
174	level platforms	3.0
175	business jumps ahead with xenos super adapters	3.0
176	right disk-based	3.0
177	world newsletter	3.0

表 B.5 2006.05.17 算法返回新术语列表 (粒度= 2 分析点数目= 10)

排名	新术语	分数
1	xxx	4.0
2	rayovac chooses windows	4.0
3	your ilm strategy	4.0
4	risk microsoft	4.0
5	oracle one	3.0
6	kinte	3.0
7	businesses oracle	3.0
8	utilize multithreading processors	3.0
9	aberdeen group apani networks	3.0
10	ajax resources	3.0
11	insider threat benchmark report & strategies	3.0
12	java everywhere	3.0
13	jpr75_z	3.0
14	sun developer	3.0
15	solutions catalogue	3.0

表 B.6 2006.05.17 算法返回新术语列表 (粒度= 1 分析点数目= 14)

1	yagu	6.0
2	rayovac chooses windows	5.5
3	risk microsoft	5.5
4	coverage & capacity motorola	5.0
5	lan work	5.0
6	uh	5.0
7	rob t firefly	4.5
8	technicon	4.5
9	tackhead	4.5
10	tryout	4.0
11	digital micro-markets	4.0
12	new york-june	4.0
13	featured event	4.0
14	online newsletters	4.0
15	new backbone	4.0
16	universal desktop	4.0
17	google music	4.0

续下页

续表 B.6 2006.05.17 算法返回新术语列表 (粒度=1 分析点数目=14)

18	structured and unstructured	4.0
19	right web	4.0
20	gosling	4.0
21	business coummunications	4.0
22	sunrocket voip	4.0
23	expensive microsoft	4.0
24	citrix gotomypc corporate	4.0
25	wearable tech	4.0
26	order reprints	4.0
27	on stylish asus notebooks	4.0
28	business oracle	4.0
29	newsletters help	4.0
30	xt / eos	4.0
31	interactive technology	4.0
32	digg	4.0
33	cost-effective remote	4.0
34	preferred e-mail	4.0
35	aligned	4.0
36	duh	4.0
37	firms use web-based quickbase	4.0
38	managing through workplace disruption	4.0
39	edt/11 a	4.0
40	cingular cell	4.0
41	click below	4.0
42	optimizing data	4.0
43	proliant dl385	4.0
44	winhec	4.0
45	enterprise anti-matter	4.0
46	eweek editorial	4.0
47	reworking	4.0
48	real-time access	4.0
49	adaptec	4.0
50	ab bc mb nb nf ns nt on pe qc sk yt	4.0
51	free phones	4.0
52	business priorities	4.0
53	web buyers guide	4.0

续下页

续表 B.6 2006.05.17 算法返回新术语列表 (粒度=1 分析点数目=14)

54	chicago-june	4.0
55	calendaring features	4.0
56	need robust data	4.0
57	unexpected workplace disruption	4.0
58	davis eseminar update	4.0
59	magazine's free	4.0
60	san francisco-june	4.0
61	pcmagcasts	4.0
62	own website	4.0
63	trialware	4.0
64	subscriber help	4.0
65	free pcmagcasts	4.0
66	ae ap	4.0

表 B.7 2006.05.17-2006.05.20 人工确定新术语列表

2006年05月17日	
新术语	例句或说明
Adaptec	Protect Your Storage Investment with Adaptec.
Business Communication	Free White Paper: The Next Leap In Business Communications.
Critical Strategy	CIO Alert: Critical Strategies for Managing Information in the 21st Century with Allan Alter & Elliot Markowitz.
Data Archiving	Headline
Enterprise Anti-matter	Headline
ETech	Headline
Google Music	Google Music gets closer Garrett Rogers.
online newsletter	Headline
overkill	Beware of broadband speed overkill
Oracle ONE	Oracle ONE: Technology and Solutions Catalogue for Small, Medium and Growing Businesses Oracle
real-time access	This process must support both structured and unstructured data with real-time access that is secure, compliant and automates the information lifecycle management process.
storage initiative	Federal government executives must implement a new generation of storage initiatives that simplify storage infrastructure management...

续下页

续表 B.7 2006.05.17-2006.05.20 新术语列表

SunRocket	Company name
supercomputing	Headline
Trialware	Diskeeper 10 Trialware for Networks, 30-Day Trialware
Wearable Tech	Headline
WinHEC	The move comes less than a week before the company will host its Windows Hardware Engineering Conference (WinHEC) in Seattle.
Workplace Disruption	Business Continuity: Managing Through Workplace Disruption.
xxx domain	Supporters said a .xxx domain would have made it easier to confine sex sites or filter them out...
2006 年 05 月 18 日	
新术语	例句或说明
bonus pack	Add up to \$1,200 of value with the new BONUS PACKS.
Business Resilience	Business Resilience: Proactive Measures for Forward-Looking Enterprises IBM
Code Name Tracker	Headline
googlepages	Product name
Mobile Messaging	Headline
Seamless Mobility	Seamless Mobility: A Continuity of Experiences Across Domains, Devices and Networks Motorola.
Serial Attached SCSI	...processor and Serial Attached SCSI hard drives offers dual core ...
transfer rates-speed	The HP ProLiant DL385 server with AMD Opteron ? processor and Serial Attached SCSI hard drives offers dual core processing power and faster data transfer rates-speed and performance in one sleek package.
user-access-protection feature	The user-access-protection feature in Vista will add an extra layer of security to Windows. But could too much security be a bad thing?
Vista-capable	The "Vista-capable" program allows machines that meet a minimum set of requirements to tout themselves as able to run the new Windows.
Watch Premium	Get a 14 Day Free Trial of Microsoft Watch Premium!
Year-Ahead Planner	Microsoft Watch Code Name Tracker 4.4 and Year-Ahead Planner 4.4
2006 年 05 月 19 日	

续下页

续表 B.7 2006.05.17-2006.05.20 新术语列表

新术语	例句或说明
Google Talk	The big Google Talk fallacy in Hitwise's Top 20 Google Domains meme
integrating compliance	Integrating Compliance Into Your Disaster Recovery Plan ZDNet
office-all-in-one	Headline
Performance Monitoring	Effective Change Management Through Application Performance Monitoring
picasa	Product name
SoBig Virus	Headline
2006年05月20日	
新术语	例句或说明
Automated Discovery	Automated Discovery: The First Step Toward Business Service Management Success with Frank Derlfer.
Channel Business	Headline
E-Mail article	Headline
Enterprise Online Event	Join AMD in the Enterprise Online Event
handy reference	This handy reference card contains features at a glance, sales objection handling, pricing guidelines & more.
Insider Information	Microsoft News and Insider Information
IP Migration	The Return on Voice over IP Migration
Licensing Requirements	Headline
mop up	Mopping Up Dirty Data
On-Demand eSeminars	Headline
Open-Source Desktops	Headline
Partner Recruitment	Vernier Networks Steps Up Partner Recruitment
Protection Gap	Close the Security Protection Gap with Integrated Threat Management with Elliot Markowitz.
Software Distributor	Headline
Turnkey Integration	Headline
two-way Web	...he is only now realizing his early vision of a two-way Web where people can easily work together on the same page...
Unified Threat Management	Unified Threat Management: The Best Defense Against Blended Threats
Vendor Solutions	Headline
Vernier Networks	Vernier Networks Steps Up Partner Recruitment

续下页

续表 B.7 2006.05.17-2006.05.20 新术语列表

Vulnerability Management	Using Vulnerability Management to Improve IT Security with Frank Derfler.
Year-Ahead Calendar	Product name