基于链接和页面分析的主题爬虫算法在个性化新闻推送系统的应用研究

第一章 绪论

1.1 背景

1.2 研究现状

1.3 研究内容

1.4 本章小结

第二章 相关技术综述

2.1 通用爬虫简介

2.2 主题爬虫简介

2.3 消息队列技术综述

2.4 个性化新闻推送系统综述

2.5 本章小结

第三章 爬虫爬行策略研究与分析

3.1 传统主题爬虫爬行策略

3.2 主题爬虫爬行策略

3.3 策略对比

3.4 本章小结

第四章 爬虫爬行策略中主题相关度算法的改进

4.1 Page Rank算法

4.2 Fish Search算法

4.3 改进的Fish Search算法

4.4 Page Rank结合改进的Fish Search算法

4.5 本章小结

第五章 实验结果及分析

5.1 实验目的

5.2 实验环境及数据

5.3 评价指标

5.4 实验设计

5.5 实验结果及分析

5.6 本章小结

第六章 个性化新闻推送系统“及时推”的实现

6.1 需求分析

6.2 系统概述

6.3 系统相关模块实现

6.4 系统相关模块运行效果

6.5 本章小结

第七章 总结

第一章 绪论

1.1 背景

1.2 研究现状

1.3 研究内容

个性化新闻推送系统通过用户自定义关注的网站和关键词，爬取每一天更新的新闻，推送到用户的邮箱和客户端。

第二章 相关技术综述

2.1 主题爬虫技术综述

2.1.1 通用爬虫简介

网络爬虫，是一个根据链接来遍历网络上页面，并按照标准HTTP协议访问其页面内容的程序。网络爬虫是搜索引擎的核心部分，为了每次检索的信息能够准确全面，需要它把网页爬取下来，建好索引存储在数据库中，检索结果直接从数据库中查找并返回，它的性能好坏直接影响搜索引擎的整体处理速度。随着技术的发展和人们需求的提高，网络爬虫越来越多地应用于内容安全检测、用户兴趣分析、微博话题挖掘以及个性化信息推送等多种应用中。

一个通用的网络爬虫的结构如图所示，主要由以下四个模块组成：

1. url分析模块，存放待访问的url队列
2. 页面爬取模块，根据HTTP协议访问url对应的页面
3. 页面分析模块，从爬取到的页面抽取特征信息和超链接url
4. url过滤模块，过滤掉已访问过的，或者已存在待访问url队列中url

网络爬虫的爬取过程从一个或者多个待爬行的url队列开始，不断的增加和移除url，利用HTTP协议下载url对应的网页，分析出网页中所有的url，过滤掉非法链接，然后将未被访问的url加入待访问url队列中。整个过程一直循环，直到满足提前定义的结束条件。网络爬虫实际上是沿着网页超链接，根据深度优先、宽度优先、或者其他策略遍历网络信息的过程。现代网络爬虫除了必须要具有好的健壮性，还需要具有分布式、可扩展、高性能、高效率、高时效等特性。

2.1.2 主题爬虫简介

主题爬虫是通用爬虫的扩展，不仅仅是对页面的简单抓取，是根据主题爬取更多的与主题相关的页面。包括确定主题、分析网页超链接和内容、根据主题相关性确定下一个访问的url，以此爬取到更多与主题相关的页面。因此主题爬虫有以下几个问题：

1. 如何描述主题？采用主题作为完整爬取关键词，可能爬取不到包含相似主题的网页，降低了主题爬取的覆盖率。
2. 如何确定爬取到的URLs的优先级顺序，即如何判定网页与主题的相关性？采用何种相关性方法对主题爬虫爬取数据的准确率有着及其重要的作用。
3. 如何提高爬取的效率，增强其算法的健壮性，不丢失信息？

主题爬虫基于通用爬虫，增加了主题相关性分析的模块，是主题爬虫的核心模块。包括url分析模块、页面爬取模块、页面分析模块、主题相关性分析模块以及url过滤模块，其流程如图所示。

主题爬虫在第一次爬取时，需要一个初始种子url，初始种子url必须是与主题相关度很高的大型门户网站，这样才能获得较多与主题相关的url，提高爬取的准确率。接下来，从原始html页面抽取超链接信息和特征信息，按照事先确定好的判定网页相关性的算法分析网页的相关性，保留与主题相关性高的页面。然后提取上述页面的超链接，过滤掉非法链接，结合网页内容的相关度或者重要性综合得出url的优先级，接着取出优先级最高的url爬取它的内容，循环此过程，直到满足结束条件就退出。

2.2 消息队列技术综述

消息队列技术是应用程序间的通信方法。应用程序间通信不需要建立专用连接，通过读取队列里的消息来通信，使得应用程序接收方和发送方不必同时执行。其成熟的产品主要有有IBM WEBSPHERE MQ、RabbitMQ等。

RabbitMQ是基于AMQP的erlang实现的。其核心是Message Exchange、Message Queue两种实体，一个应用程序产生的消息不是直接发送给另一个应用程序，而是发送给Message Exchange，同时指定一个Rounting Key，如果与Message Queue的Binding Key匹配，Message Exchange就将消息发送给Message Queue，等待另一个应用程序的读取。主要有以下几个特点：

1. 可以在多个平台上运行，支持多种操作系统。
2. 提供消息的持久性支持，即服务器意外崩溃后消息能够恢复。
3. 提供了消息接受者和消息发布者的确认机制，保证消息准确接收。

2.3 个性化新闻推送系统综述

目前，个性化新闻推送系统主要是通过用户访问纪录对用户兴趣进行建模，然后跟据用户兴趣模型对用户推送新闻。但是，用户对新闻的访问只能代表某一时刻的关注，并不能代表用户的兴趣，导致为用户推送的新闻狭隘化。

2.4 本章小结

第三章 爬虫爬行策略

主题爬虫与通用爬虫的最大区别在于其主题相关度计算及相关的爬行策略，所以要实现一个完整的爬虫系统，采用合适的爬虫爬行策略及其重要。本章结合已有的爬行策略和自己的理解，对现有的爬行策略进行了分析、研究与对比。

3.1 爬虫爬行策略

爬虫爬行策略是爬虫的核心问题，指的是如何在海量的信息网站中爬取到人们感兴趣的信息，由此可见爬虫爬行策略的好坏直接影响整个爬虫系统的性能。如果能设计出一个好的爬行策略，那么爬虫就能最大可能达到我们需要的效果。但是由于目标不同，爬虫的爬行策略也会不同，接下来本节先主要讲述已有的爬虫爬行策略。

3.2 通用爬虫爬行策略

通用爬虫的目标是遍历整个互联网，尽可能多的采集网页。其爬行策略主要为深度优先、广度优先和深度优先和广度优先结合三种。深度优先、广度优先以及深度优先和广度优先结合的爬行策略都是将网络看成一个复杂而庞大的有向图，将网络上的所有网页当成节点，网页之间的超链接当成节点之间的连接，根据图论的搜索方法，对网页节点采取深度优先、广度优先或者深度优先和广度优先结合的遍历策略。下面以一个结构图来说明这三种算法的区别。

3.1.1 深度优先

深度优先的爬行策略是从初始网页节点开始，首先访问该节点，然后沿着一条超链接路径出发，访问其邻结点，一直爬行到没有包含任何超链接信息的网页节点为止，再返回访问其他未被访问的节点，循环下去，直到所有的节点都被访问到。那么根据深度优先的爬行策略，图上的网页节点访问路径应该是ABDFCE。深度优先的爬行策略优点是爬取的页面很全面；缺点则是效率很低，当页面目录很深的时候，容易导致爬虫陷入。

3.1.2 广度优先

广度优先的爬行策略是从初始网页节点开始，访问此节点后，就访问其他没访问过的相邻节点，然后再访问剩下节点的邻结点，直到爬行结束。比如上图，从A节点访问开始，继续访问时可以选择A的邻结点B或者C访问，所以广度优先的爬行策略有多种爬行顺序。按照上图总共会出现多种爬行顺序，ABCDFE是其中的一种。广度优先的爬行策略优点是当初始网页有足够多的链接时，会得到很好的爬取结果；但是，广度优先的爬行策略缺点也很明显，当爬行深度越来越高时，则会产生许多无效的网页，导致算法的效率降低。

3.1.3 深度优先和广度优先结合

针对深度优先和广度优先各自的优缺点，又出现了深度和广度结合的优先爬行策略。这种爬行策略会预设爬行深度或广度，当爬虫爬行到预设深度或者预设广度时就停止工作。所有它相比深度优先和广度优先，都提高了效率，但还是会爬取不少的无效网页。

3.1.4 对比分析

从以上几节发现，深度优先、广度优先以及深度优先和广度优先结合都几乎遍历整个网络，能爬取大量的网页，非常适用于通用爬虫的爬行。但是，对于主题爬虫来说，上述三种策略都没有对网页和网页内容作出评估分析，会爬取大量与主题无关的网页，不能满足主题爬虫的需求。

3.2 主题爬虫爬行策略

由于通用爬虫的爬行策略效率低，会产生大量无效网页的问题，不能应用于主题爬虫算法中。目前主题爬虫的爬行策略主要有基于网页链接、基于网页内容等爬行策略，下面讲述这两种爬行策略的原理和优缺点，并进行对比。

3.2.1 基于网页链接的爬行策略

万维网是超文本文档的集合，超文本文档相互之间有一定的链接关系，形成一定的链接结构。基于网页链接的主题爬行策略最重要的就是要对超文本文档的链接结构进行分析，通过对链接的有效评估，决定爬行的顺序。万维网信息分散，但经研究表明，其链接结构具有自组织性：通过超链接，相同或者相关主题的网页结点聚集在一起，形成结点区域，区域内部的结点存在紧密联系，而区域之间的结点很少或者根本没有联系。并且，区域内部的结点也存在一定的规律，其结点主要分为两类，一类指向其他结点，一类被前一类结点所指向。前一类结点称为中心结点，后一类结点称为权威结点。万维网的这种链接自组织性结构为链接的评估提供了依据，如何评估链接主要有以下假设：

1. 从页面A到页面B的一条链接是页面A对页面B的一种推荐。
2. 若页面A与页面B被同一页面链接，则他们可能有相同或者相关的主题。

Page Rank算法和Hits算法是常见的链接分析算法，两者都是通过分析网页的拓扑结构，得出网页的重要程度评价。

3.2.2 基于网页内容的爬行策略

基于网页内容的爬行策略，主要以向量空间模型等信息检索模型为基础，利用网页页面内容（包括网页中文本信息、URL、锚文本和锚文本的上下文等）来评估网页的重要性，以此决定爬行顺序。基于网页内容的爬行策略的核心问题就是对上述纯文本和超文本进行分类和聚类。

1. 纯文本的分类聚类就是对页面纯文本进行检索，准确的说，指通过计算机使用自然语言中的词语对纯文本进行匹配查找。常用的检索技术有精确检索、截词检索、布尔检索和限定范围等。在主题爬虫中，由于文本检索技术忽略了网页页面内部的结构信息，必须限定检索的关键词集合下进行检索，很少单独使用。
2. 超文本的分类聚类不仅包含了对一般的文字，还包含了一些对文本信息处理有用的特征信息的检索。网页文本是一个半结构化的标记语言，其中的HTML标记语言具有结构特征和语义信息，根据特定的HTML标签检索文本，排除了无关的干扰信息，使得文本检索更加精确、快速。从而提高了爬虫爬行的效率。

Best First Search、Fish Search和Shark Search算法是常见的基于页面内容的搜索算法。

3.3 策略对比

基于网页内容的爬行策略需要对纯文本和超文本进行分类聚类，首先使用超文本分类聚类方法将HTML语言中的纯文本抽取出来，再使用纯文本分类聚类方法，检索其中的关键词，最后与主题关键词进行比较判断页面与主题的相关度。

基于网页链接的爬行策略需要计算页面节点的入度和出度值，只有在爬虫对网络拓扑结构有相对完整的统计情况下，才能计算出准确的网页的重要性程度值。另外，该爬行策略没有考虑到网页内容的重要程度，也许页面在当前网页范围里很重要，但并不是所需要的。所以在主题爬虫系统中，基于网页链接的爬行策略不能单独使用，需结合能判断网页主题的基于网页内容的爬行策略。

3.4 本章小结

爬行策略是主题爬虫的核心，本章介绍了现有的多种爬行策略并进行了对比。首先介绍了通用爬虫的三种爬行策略，并对比分析其优缺点；接着重点介绍了主题爬虫的主要爬行策略：基于网页链接和基于网页内容的爬行策略，并对这两类爬行策略进行了详细分析和对比。

第四章 爬虫爬行策略中主题相关度算法的改进

主题爬虫的目的是爬取到更多与主题相关的页面，减少爬取到与主题不相关的页面的可能性。爬行过程中，需要对已有网页内容分析处理以确定其URL的优先级，与主题相关度高的URL优先级高，将优先处理。本章对爬虫爬行策略中PageRank和Fish Search主题相关度算法，进行了分析和研究，并针对Fish Search算法存在的问题，提出了一种改进了的Fish Search算法，最后，结合Page Rank和改进了的Fish Search算法设计出基于链接和页面内容的主题爬虫策略并实现其算法。

4.1 Page Rank算法

PageRank算法是基于网页链接的的主题爬行策略常用的算法，本节会对该算法进行研究和分析。

4.1.1 Page Rank算法思想

PageRank算法由S.Brin和L.Page发明，S.Brin和L.Page利用网络的超链接结构给所有的页面定义了一个PageRank值，用Page Rank值的高低代表页面的重要性，以此确定爬行的页面顺序。PageRank算法的原理是将学术论文的引用分析方法应用到网页链接中。一篇论文的重要性是可以根据其被引用的次数来判断，同样一个网页的重要性也可以根据其被链接的次数来判断。具体来讲，如果网页A链接到网页B,相当于网页B的被链接次数（PageRank值）增加，同时网页A的PageRank值减小，最后由每个网页的PageRank值来评估链接。

4.1.2 Page Rank算法描述

Page Rank算法最开始是运用于谷歌搜索结果的排序，最近较多应用于网页链接重要性评估中。Page Rank计算公式如下：

其中，代表网页u的重要度，代表指向网页u的页面的重要度，代表指向页面u的超链接数，为指向u的网页的集合，c为常数，一般c取0.85。经观察公式，可以看出影响页面u的重要度的因素有以下几点：

1. 页面u的入度，如果链接到u的网页越多，越大。
2. 链接到页面u的网页重要度，链接到u的网页重要度越高，越大。
3. 链接到u的网页出链数量，链接到u的页面出度越大，平均传递给u的重要度值就越小。

4.1.3 PageRank算法优缺点

PageRank算法通过网页链接的PageRank值的高低代表网页的重要性程度，并成功运用与google搜索引擎，证明了其有效性。但这种算法只考虑了网页链接，没有考虑到网页内容，如果运用到主题爬虫中，易造成“主题漂移”现象，导致爬取到许多与主题不相关的页面。所以在本文的爬虫系统中，考虑采用结合一种基于网页内容判断主题相关度的算法，在4.2节研究了Fish Search基于网页内容判断主题相关度的算法。

4.2 Fish Search算法

Fish Search 算法是一种基于网页内容的爬行策略的经典算法，本节会对该算法进行研究和分析。

4.2.1 Fish Search算法思想

标准的Fish Search算法是由De Bra在1994年提出的。该算法将爬虫的爬行过程比喻成鱼的捕食过程，算法中的网页就是鱼的食物，沿某个方向的遍历就是鱼的捕食路径。当爬行过一个网页后，提取的新的URL就是鱼的Children，一直循环下去，直到鱼找不到食物饿死，即爬虫找不到任何链接的时候就结束爬行了。

4.2.2 Fish Search算法描述

Fish Search算法的关键之处在于动态维护了三个URL队列，分别是前端队列B、中间队列M和末尾队列E，其算法描述如下：

1. 首先选择一个初始URL放入待爬取URLs队列中，并对最大爬取深度depth赋值。
2. 提取初始网页的子URL，如果初始网页是主题相关的，则其子URL的相关度值为1，depth值不变，否则相关度值为0，depth值减一。并按照（3）（4）（5）将新的URLs插入队列。
3. 若该URL对应的网页与主题相关，就选取m\*width（m，width为预先确定的参数）个URL插入到前端队列B中。
4. 若该URL对应的网页与主题无关且depth不为0，就选取width（width为预先确定的参数）个URL插入到中间队列M中。
5. 剩余的URLs插入到末尾队列E中。有充足的时间才会爬取这个队列的URLs。
6. 队列为空时停止爬取。

4.2.3 Fish Search算法优缺点

Fish算法对待爬行的URLs进行了优先级判断，且其原理简单，容易理解。但是该算法存在两个问题，第一个问题是只是考虑简单匹配关键词来计算相关度，导致结果准确度不够；第二个问题是使用离散点的表示主题是否相关，无法完全表示页面与主题的相关性。

4.3 改进的Fish Search算法

4.3.1 算法分析

由上一节我们了解到，Fish Search算法存在一些问题。首先，Fish Search算法采用布尔模型信息检索方式计算页面主题相关度。首先将网页中的文本提取出来分词，得到一个关键词的集合，然后与主题中的关键词集合对比，统计同时出现的关键词个数，最后用同时出现的关键词个数除以主题中关键词的个数，得到的比值就是网页的主题相关度。该比值是离散点，导致该比值无法完全表示页面与主题的相关性。

然后，Fish Search算法对所有页面上的关键词没有区分，忽视了关键词位置的重要性。一个网页上有多种文本，不同标签里的文本意义不同，对网页主题的重要性影响也不同，不考虑关键在网页中的位置，导致相关度结果不准确。因此本节对Fish Search算法的主题相关度方法进行了改进。

4.3.2 算法改进

准确获得一个网页的主题是计算网页与主题相关度的前提，网页的标签是语义化的，能表示网页主题的文本主要存在于页面标题title和关键词keywords、页面里链接a的锚文本以及页面里的其他标签。页面标题和关键词最能代表网页的主题，其权值最高；页面里链接的锚文本一般是其子网页的主题，对页面的主题判断也有一定的作用，其权值居中；其他正文中的文本一般只是提到而已，对页面的主题判断作用不大，其权值最小。所有针对关键词在网页中出现的不同的位置赋值公式如下：

设主题关键词出现在网页标题中的权值为 ，并根据公式2带入参数，得到的计算公式入公式3所示。

设主题关键词出现在网页关键词中的权值为 ，并根据公式2带入参数，得到的计算公式入公式4所示。

设主题关键词出现在网页锚文本中的权值为 ，并根据公式2带入参数，得到的计算公式入公式5所示。

公式（3）（4）（5）中的T、K和L分别代表主题关键词在网页标题、网页关键词和网页锚文本中出现的次数，表示关键词i分别在网页标题、网页关键词和网页锚文本中出现的次数，w为公式2中的权重。

此时可得出网页的权值的公式如下。

将公式（3）（4）（5）的带入公式6得：

主题关键词的权重包括两部分，第一部分关键词权重指的是出现在页面中的关键词的权重，即；另一部分关键词权重则指的是为未出现在页面中的关键词权重，设为，其计算公式如下：

其中，

由此得到网页与主题相关度的计算公式如下，用sim表示相关度的值。

将公式（6）和公式（7）代入上式可得：

由公式观察得sim的值在0到1之间，是连续的，改进了Fish Search相关度值离散的问题。可以通过其大小评价网页与主题的相关程度和优先级排序。

4.3.3 算法具体实现

伪代码

4.4 Page Rank结合改进的Fish Search算法

4.4.1 算法分析

改进后的Fish Search算法度量页面主题相关度时，考虑了主题关键词在网页中的位置，并最后得到一个连续的主题相关度度量公式，对比原始的Fish Search显然要好些。但是，无论是原始的还是改进后的Fish Search算法，都没有考虑网络拓扑结构的影响，不能发现高重要度的网页。并且，爬虫初期由于爬取的网页数量少，无法确定比较完整的网络拓扑结构，导致算不出准确的Page Rank值。所以，针对Fish Search算法和PageRank的不足，本节将PageRank算法与改进后的Fish Search算法结合，设计出基于Page Rank和改进后的Fish Search算法的主题相关度计算方法。

4.4.2 算法改进

该算法由Page Rank算法与改进后的Fish Search算法组成，其主题相关性便是这两种算法的相关度综合得到。设相关度为，其综合相关性的计算公式为：

其中，权值为给定的参数，用于平衡R和sim的比例。R值是公式2计算出来的页面的Page Rank值，sim是公式8计算出来的页面内容的主题相关度值。

4.4.3 算法步骤

（1）初始化URL列表和主题关键词。

（2）提取初始URL的子链接。

（3）根据相关度计算方法计算URLs的主题相关度，并跟预先设置的阈值进行比较，并放入队列指定位置。

（4）一直循环下去直到队列没有URL。

算法的流程如图所示：

4.4.4 算法的具体实现

伪代码

4.5 本章小结

本章对爬虫爬行策略的核心问题，如何计算页面的重要程度以确定待爬行的URL的优先级进行了研究。首先研究了基于网页链接的爬行策略的常用算法PageRank，以及基于网页内容的爬行策略的常用算法Fish Search，并对比其优缺点；然后针对Fish Search算法的不足，提出了一种改进的Fish Search算法；最后结合Page Rank算法，设计出基于Page Rank算法和改进后的Fish Search算法的页面主题相关度计算方法。

第五章 实验结果及分析

本章设计了一个通用爬虫框架，分别对Page Rank、Fish Search、改进后的Fish Search以及结合PageRank和改进后的Fish Search的算法进行实验，分析实验结果并得出结论。

5.1 实验目的

基于第四章提出的主题相关度计算算法，分别实现基于Page Rank、Fish Search、改进后的Fish Search以及结合PageRank和改进后的Fish Search的算法的主题爬虫。目标是：改进后的算法能在较短的时间内，爬取大量与主题相关的页面，并且系统能稳定运行。

5.2 实验环境

实验基于MyEclipse开发环境，使用Java编程语言和Mysql数据库开发完成。硬件环境为多台台式机，配置为：Pentium 4双核，主频1.4GHz，内存4GB。

5.3 评价指标

为了很好的评价改进后算法的效果，引入了查准率、发现率和平均每个单位页面的爬取时间性能指标。查准率指的是爬取到的主题相关的页面数与实际主题相关的页面数之比。发现率是爬取到的主题相关的页面数与爬取到的页面总数之比。平均每个单位页面的爬取时间指的是爬取页面的总时间与爬取到的页面总数之比。

5.4 实验设计及数据

实验1采用新浪网体育类目主页和关键词足球，每爬取100个页面，统计与主题相关的页面个数，计算其与爬取页面总数的比值，即发现率。

实验2采用上述新浪网体育类目的数据作为数据源，其中足球类页面有2000个页面，同样的，每爬取100个页面，计算查准率、发现率和平均每个单位页面的爬取时间。

实验实现了一个通用的爬虫框架，下面结合框架界面介绍实验的主要步骤：

1. 输入初始URL和主题关键词。如图1所示。
2. 点击开始爬取。 系统下载初始URL对应的页面，抓取页面上的链接，并选择优先级高的链接爬行。如图2所示。
3. 爬取下来的页面保存在Mysql数据库。如图3所示。

5.5 实验结果及分析

实验1

实验2

5.6 本章小结

第六章 及时推新闻推送系统的实现

6.1 需求分析

随着时代的进步和计算机技术的迅速发展，以互联网为特征的第四媒体给传统的新闻媒体带来冲击和挑战，伴随着互联网信息时代的带来，信息数量高速增长，据不完全统计，全球互联网文本数据100亿条，其中招聘信息大于10亿条，新闻大约8亿条，招商信息大约3亿条，通知信息大约2亿条。并且互联网信息更新速度飞快，在浩如烟海的网络资源里，想要及时的获取想要的信息变得越来越难。针对上述问题，有以下几种解决办法：

1. 使用搜索引擎，目前的搜索引擎都是面向网络上所有的信息的，随着信息的增长和全面化，针对受限领域和特定主题的搜索的准确率和查全率都是很低的。
2. 使用新闻网站以及客户端，用户可以设置自己感兴趣的主题，阅读自己感兴趣的内容。但是，新闻网站以及客户端上的新闻大部分是用户不关心的，由于不能定制关注的网页，不能推送用户真正需要的信息。

综上，我们知道，用户真正需要的是一个完全定制的信息推送系统。在这个系统里，通过用户定制的关注的网站和关键词，将每天推送该网站更新的包含该关键词的页面给用户。该系统的核心就是爬虫子系统，即如何准确全面地爬取用户定制的感兴趣的网页，因此将本文设计并实现的主题爬虫算法应用到该系统中，为系统提供用户定制的感兴趣的网页。

6.2 系统概述

及时推新闻推送系统是四川省成果转换项目，分别实现了PC端、Android客户端和IOS客户端，并于2014年5月20日全部成功上线。该系统主要由3个子系统组成：

1. 爬虫子系统：根据用户定制的感兴趣的网页和关键词，爬取该网页上每天更新的包含关键词的所有URL，并存入数据库。
2. 信息推送子系统：根据用户特征、用户浏览数据等，每天为用户推送信息到PC网页客户端、Android客户端、IOS客户端以及手机或邮箱。
3. 信息展示子系统：通过PC网页客户端、Android客户端和IOS客户端显示系统给用户推送的信息，方便用户随时随地查看。

以上三个子系统中我负责的是爬虫子系统以及各个客户端的展示子系统。整个过程中，首先，爬虫子系统为信息推送子系统提供爬取到的用户定制的网页信息，接着，信息推送子系统根据特定规则将信息推送给用户，最后显示在信息展示子系统上。因此，爬虫子系统是整个系统的基础，也是关键的一个步骤，如果爬取到的网页信息的准确率和覆盖率不够，那么后面的推荐和展示就没有办法实现。

图2就是该系统的软件架构图。

图2中的几个关键模块主要是主题相关网页信息爬取、网页信息推送和网页信息展示。主题相关网页信息爬取模块是最基础关键的模块，主要包括网页采集模块、网页主题分析模块、链接优先级计算模块和网页信息持久模块。主题相关网页信息爬取模块将爬取的网页信息持久化后存入数据库中，网页信息推送模块从数据可中获取所有的网页信息进行分析后推送个用户，网页信息展示模块就直观地显示推送给用户的信息。

图3就是该系统的硬件结构图

图3中可以看出，系统总共使用了三台服务器，一台用于主题爬虫，一台用于web服务，一台用于数据库。为了充分使用服务器的资源进行主题爬虫，将用于主题爬虫的服务器虚拟化，运行了两个虚拟机，分别装linux操作系统运行主题爬虫的爬行子系统。Web服务器采用开源的Tomcat服务器，用于给用户推送网页信息，并展示在各个客户端上。数据库服务器则是用于存储爬虫子系统爬取的网页信息数据。

6.3 系统主要模块实现效果

由于爬虫子系统和信息推送子系统都是在后台运行的，本节主要分别以不同客户端展现信息展示子系统的实现效果。

6.3.1 PC客户端

6.3.2 Android客户端

6.3.3 IOS客户端

6.4 本章小结

本章首先对及时推新闻推送系统进行了需求分析，接着介绍了系统的构成和软硬件结构，最后分别以PC客户端、Android客户端和IOS客户端的运行情况展示了系统相关模块的实现效果。

第七章 总结