基于行块分布函数的通用网页正文抽取

陈 鑫 (Xin Chen)

哈尔滨工业大学社会计算与信息检索研究中心（HIT-SCIR） http://ir.hit.edu.cn/

Google Code 开源网址址：http://code.google.com/p/cx-extractor/

新浪微博：http://weibo.com/cx3180 腾讯微博：http://t.qq.com/cx3180

欢迎意见和建议：xchen@ir.hit.edu.cn

前言

对于 Web 信息检索来说，网页正文抽取是后续处理的关键。虽然使用正则表达式可以准确的抽取某一固定格式的页面，但面对形形色色的 HTML，使用规则处理难免捉襟见肘。

能不能高效、准确的将一个页面的正文抽取出来，并做到在大规模网页范围内通用，这是一个直接关系上层应用的难题。

采用建 DOM 树的方法虽然直观也有效，但建树和搜索是多项式时间，且饱受病态 HTML 的痛苦；采用机器学习或数据挖掘的方法未免有些小题大做。

本算法首次将网页正文抽取问题转化为求页面的行块分布函数，并完全脱离 HTML 标签。通过线性时间建立行块分布函数图，由此图可以直接高效、准确的定位网页正文。同时采用统计与规则相结合的方法来解决系统的通用性问题。

本系统的设计与实现只为践行“简单的事情总应该用最简单的办法来解决”这一亘古不变的道理。整个算法实现不足百行代码。我却相信：量不在多，在法。

一、选题背景

正文抽取在信息检索系统中有重要的作用。大多数网页中除了包含有用信息（正文）外还包含许多噪声信息，例如网站的导航信息、相关链接和广告以及一些脚本语言等。如果一个信息检索系统是基于网页正文内容进行的，那么当用户输入查询关键词后，系统只是查找出正文部分和用户查询匹配的网页返回给用户，这样使得检索出的网页与用户需要更加匹配，从而使用户可以更快地找到自己所需的内容。另外，基于正文的网页去重、分类聚类以及文摘等的结果都会更加准确。

如果把完成一个完整通用的信息检索系统类比为烹制一顿美味佳肴，分词看做是切菜，那么正文提取则是切菜前必需的原料级加工——择菜。因为再出色的厨师也无法将带有黄叶和泥巴的菜做成佳肴，所以正文提取的任务就是把菜择好。

二、系统功能

本系统分在线和离线两种运行方式。

在线状态下，输入是一文本文件，里面包含要进行正文抽取的 URL，每个 URL 单独一行；离线状态下，输入是一文件夹，里面包含了所有要进行正文抽取的 HTML 源文件。

两种运行状态下的输出都是经过正文提取后的文本，具体格式如图 1 所示。

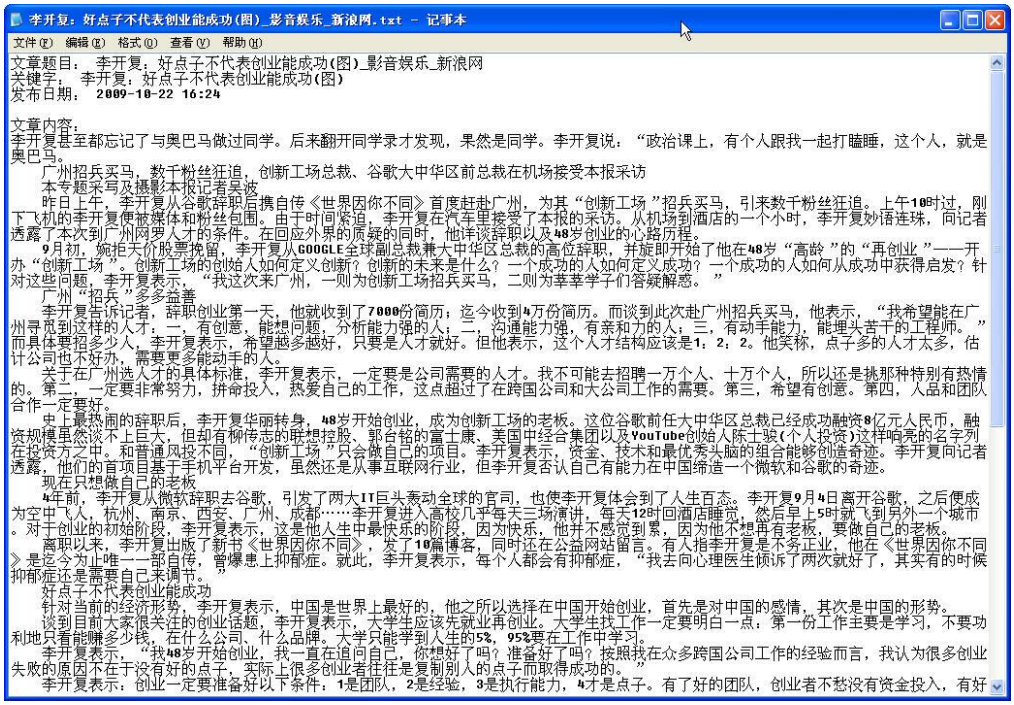

图 1：系统输出文件的格式，包括题目、关键字、发布日期和正文等

三、系统框架

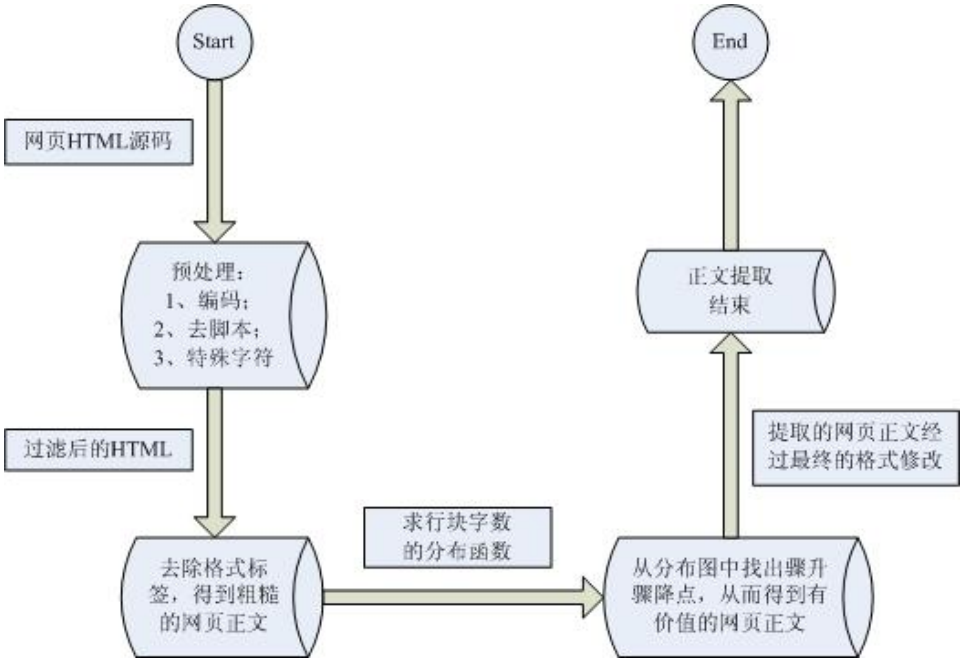

图 2： 系统流程图

四、现有方法的不足

现有第一类方法：基于 Dom 树

1、利用开源包 HTML Tidy 处理 HTML，更正其中错误或不规范的地方；

2、根据较规范的 HTML 建立 Dom 树，然后递归遍历 Dom 树，比较并识别各种非正文信息，包括广告、链接群和非重要节点信息；

- ✓ 广告信息移除：需要建立经常更新的广告服务器列表；
- ✓ 链接群移除：计算网页包含的链接个数同非链接的词个数的比例；

3、非正文信息移除后，Dom 树中剩余的内容就是正文信息，直接从余下的树节点中抽取。

存在问题：Dom 树的建立对 HTML 是否良构要求较高，而且树的建立和遍历时空复杂度高，树遍历方法也因 HTML 标签不同会有差异。

代表性论文：

[1]基于结构树解析的网页正文抽取方法, 刘秉权,王喻红等；

[2]基于统计的网页正文信息抽取方法的研究, 孙承杰,关毅；

现有第二类方法：基于网页分割找正文块

1、网页正确分割后，正文提取工作简化为对正文块的判断；

2、分割是利用 HTML 标签中的分隔线以及一些视觉信息（如文字颜色、字体大小、文字信息等）。

存在问题：不同网站的 HTML 风格迥异，分割没有统一方法，通用性难以保证。

代表性论文：

[1]基于网页分块的正文信息提取方法, 黄玲,陈龙；

现有第三类方法：基于标记窗

1、先取出文章标题；

2、两个标签及其内部包含的文本合在一起称为一个标记窗（比如<h1>text</h1>中 text 就是标记窗内的文本），取出所有标记窗内的文字；

3、对文章标题和每个标记窗内的文本分词；

4、计算标题序列与每个标记窗文本序列的词语距离 L，如果 L 小于一个阈值，则将此标记窗内的文本看做是正文文本。

存在问题：标记窗的提法很好，但每个标记窗文本都要先分词，在计算词序列距离，效率？

代表性论文：

[1]基于标记窗的网页正文信息提取方法, 赵欣欣, 索红光等

现有第四类方法：基于数据挖掘或机器学习

用到了文本分类、聚类、隐马模型、数据挖掘等。

存在问题：简单问题复杂化。

代表性论文：

[1]基于数据挖掘思想的网页正文抽取方法的研究, 蒲宇达,关毅,王强

现有第五类方法：基于逻辑行和最大接纳距离的网页正文抽取

1、考虑人们编写网页时的一些启发规则，考虑了正文的物理位置会靠的很近；

2、做出以下推论：

- ✓ HTML 每一行都表示一个完整的语义；
- ✓ 正文代码在物理位置上会靠的很近；
- ✓ 正文代码的一行中大都是文字；
- ✓ 正文代码的一行中非 HTML 标签的文字数量较多；
- ✓ 正文代码的一行中超链接长度所占比率不会很大；

3、凡符合上述推论的代码行被认为是一个正文行；

存在问题：正文在物理位置上的确会很接近，但标题很长的链接群以及较长的文章评论信息同样满足上述条件。仍要靠 html 标签做判断。

代表性论文：

[1] 基于逻辑行和最大接纳距离的网页正文抽取，张霞亮，陈家骏；

以上所有方法存在一个共同问题：正文提取被 HTML 标签所累。正文提取当真必需经过和标签纠缠这一步吗？未必。我从实验室的一篇论文《大规模网页快速去重算法》受到启发：当面对一个实际问题时，最重要的不是你设计的算法有多复杂、多高深，而是你提出的解决办法是否简单、好用。就像去重的核心就是提出了找一个句号左右的特征码，虽简单但好用。

以下是我的 idea，这种方法完全脱离了 HTML 标签。

五、系统实现——求行块分布函数

在 HTML 中，正文和标签总掺杂在一起。不可否认，标签对文字的修饰作用在词权确定和排序结果上有很大作用。但是，也正因为 HTML 标签和正文互相交织的复杂和不规范，使得通用的正文抽取变得难以实现，最终不得不针对不同网站定义不同规则，时空复杂度也大打折扣。

基于此，我想出了一种基于行块分布函数的通用方法，可以在线性时间 O(N)内抽出正文，已用 perl 实现并获得了较好效果。

提出此方法核心依据有两点：1、正文区的密度； 2、行块的长度。

依据 1：一个网页的正文区域肯定是文字信息分布最密集的区域之一，这个区域可能最大但不尽然，比如评论信息较长，或者网页正文新闻较短，而又出现如下大篇紧密导航信息时：

满文军涉毒后复出 “白发”露面只捐款不赚钱
唱响广东清远河源海选来开帷幕 84 岁阿婆参赛
陈楚生纵贯线加盟江苏跨年演唱会
陈思思唱响“魅力汤山” 台湾归来似希腊女神
满文军白发复出 刘信达：他是染的！
江苏卫视跨年演唱会 100 万邀 F4 重聚
“叛将”陈楚生不怵龙丹妮 只要不丢脸不做一哥
满文军头发花白 复出只捐钱不挣钱(图)
……

依据 2：行块的长度信息可以有效解决上述问题。

依据 1 和依据 2 相结合，就能很好的实现正文提取。我将依据 1 和 2 融合在行块分布函数里。具体如下：

首先将网页 HTML 去净标签，只留所有正文，同时留下标签去除后的所有空

白位置信息，留下的正文称为 Ctext.

定义 1. 行块：
以 Ctext 中的行号为轴，取其周围 K 行（上下文均可,K<5,这里取 K=3,方向向下, K 称为行块厚度），合起来称为一个行块 Cblock，行块 i 是以 Ctext 中行号 i 为轴的行块；

定义 2. 行块长度：
一个 Cblock，去掉其中的所有空白符（\n,\r,\t 等）后的字符总数称为该行块的长度；

定义 3. 行块分布函数：
以 Ctext 每行为轴，共有 LinesNum(Ctext)-K 个 Cblock，做出以[1, LinesNum(Ctext)-K]为横轴，以其各自的行块长度为纵轴的分布函数；

行块分布函数可以在 O(N)时间求得，在行块分布函数图上可以直观的看出正文所在区域。我从国内各大主流媒体中随机各选择一篇网页，求出行块分布函数如下图所示：

（X 轴为行号，y 轴为以该行号为轴的行块长度）

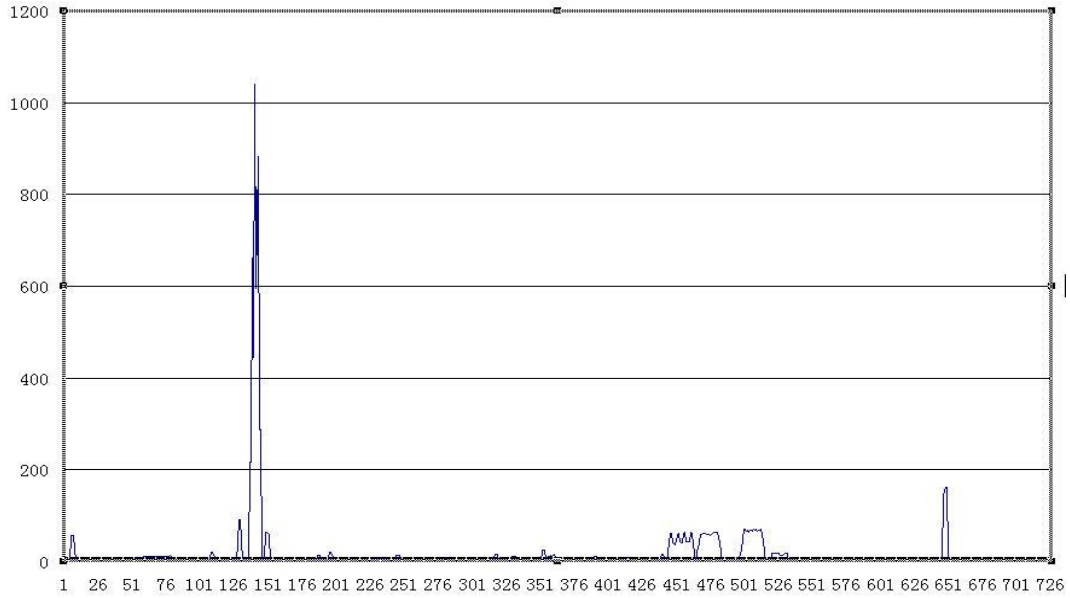

图 3. 新浪新闻，正确文本区域行号为 136——145

http://ent.sina.com.cn/y/2009-11-09/13572762965.shtml

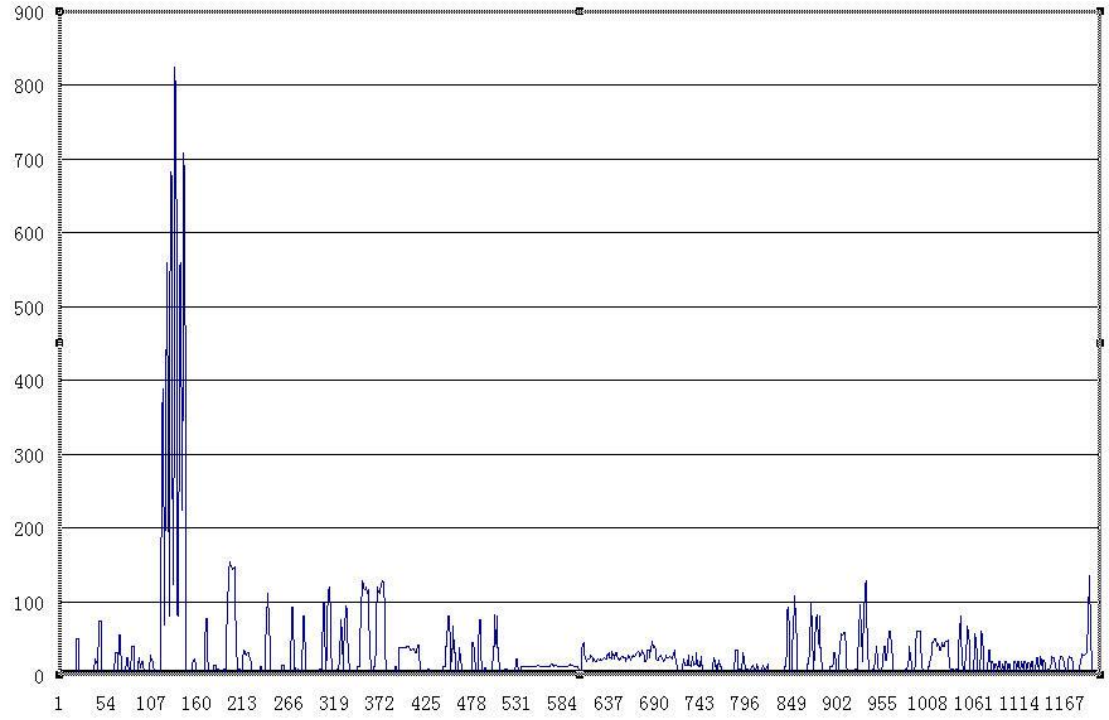

图 4. 搜狐新闻，正确文本区域行号为 118——146

http://music.yule.sohu.com/20091109/n268066205.shtml

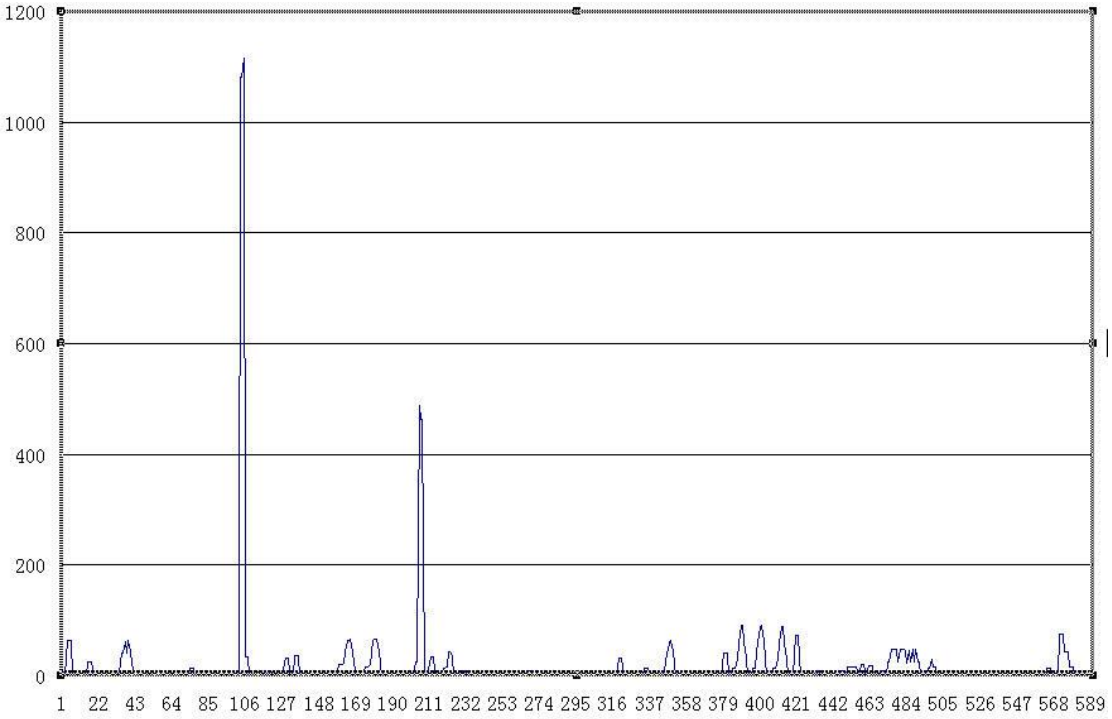

图 5. 腾讯新闻，正确文本区域行号为 102——123

http://ent.qq.com/a/20091109/000253.htm

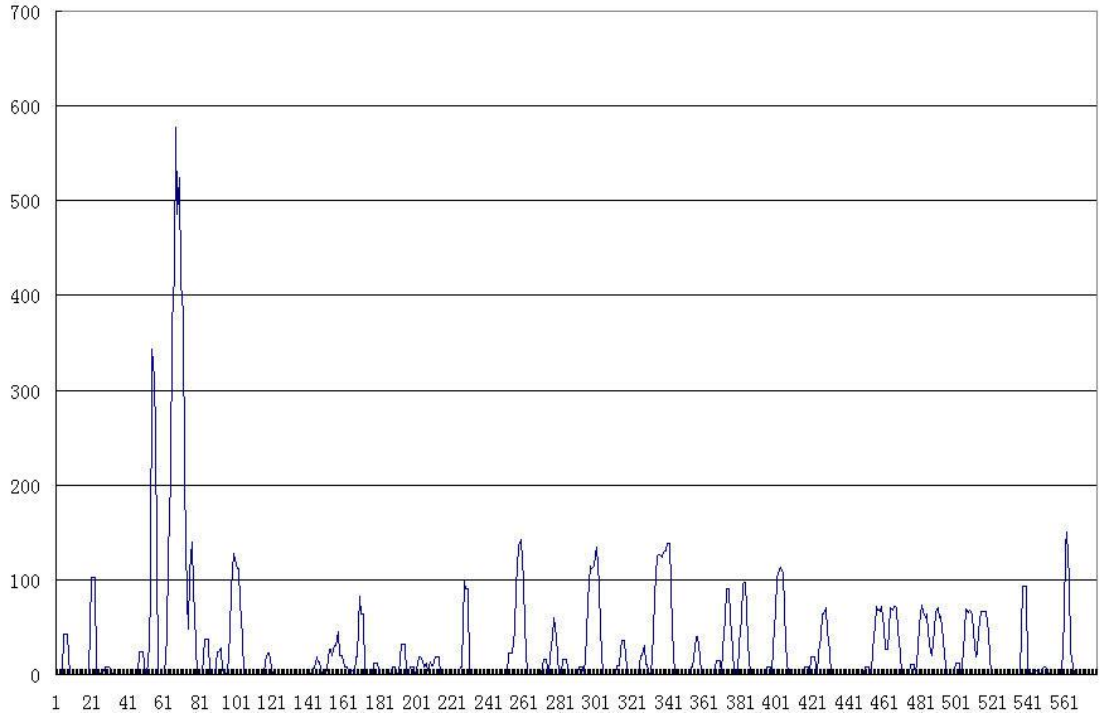

图 6. 网易新闻，正确文本区域行号为 62——79

http://news.163.com/09/1109/02/5NL6V0VB000120GU.html

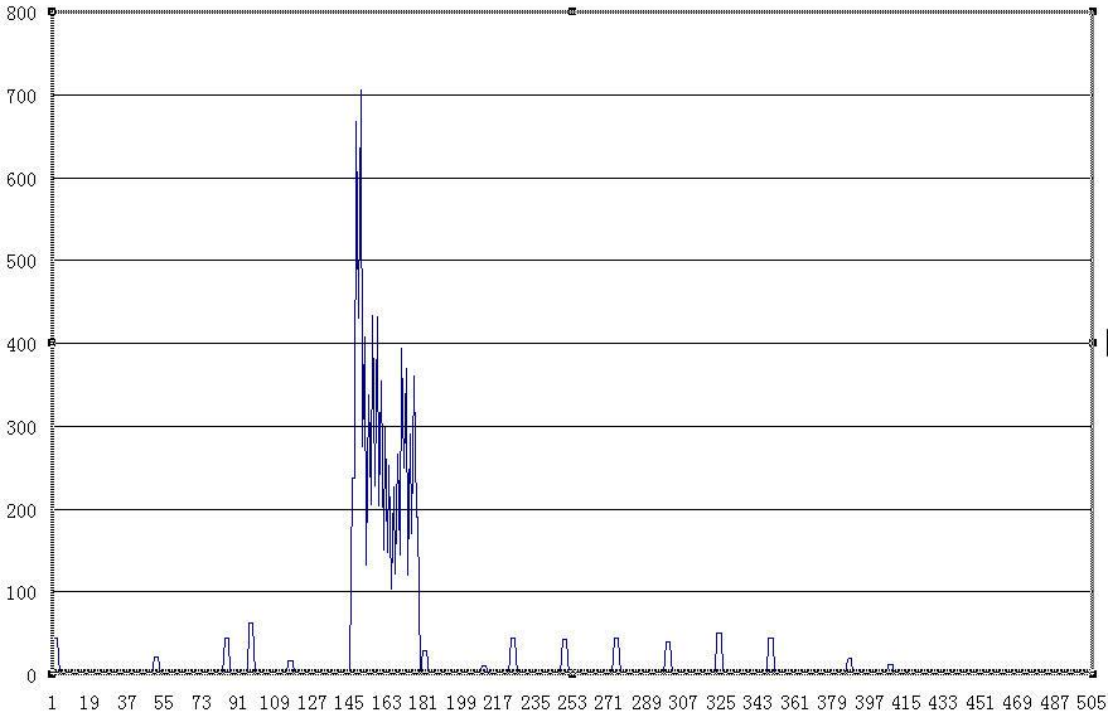

图 7. 中华人民共和国中央人民政府新闻，正确文本区域行号为 145——182

http://www.gov.cn/ldhd/2009-11/08/content_1459564.htm

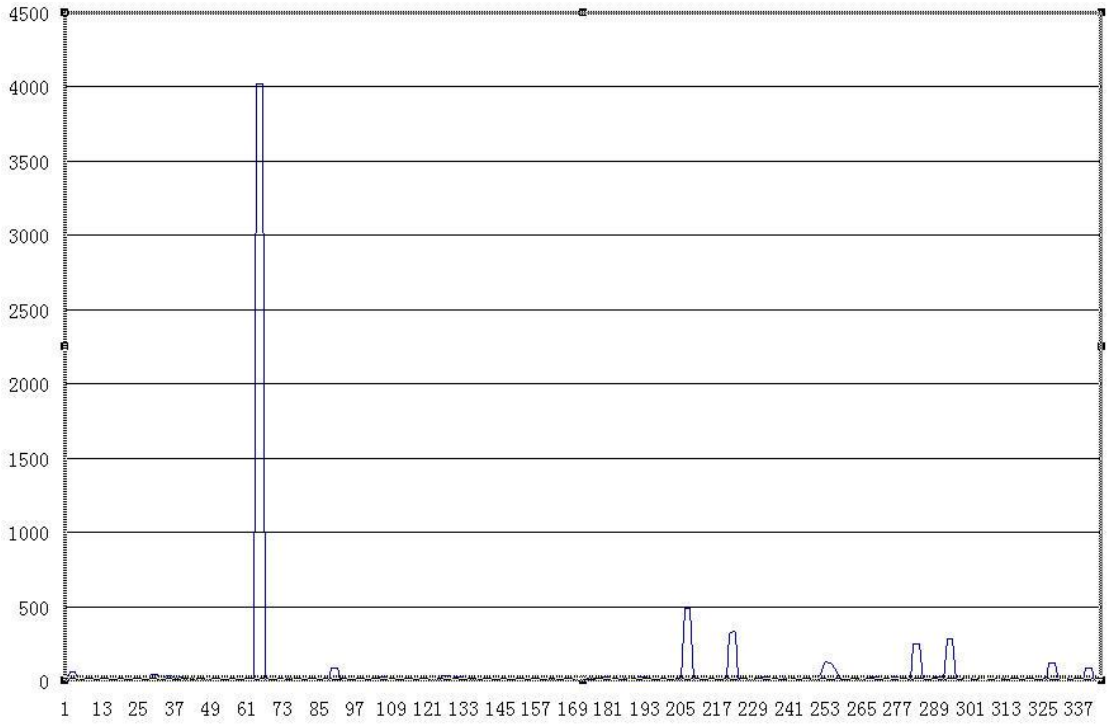

图 8. 人民网新闻，正确文本区域行号为 64——66

http://politics.people.com.cn/GB/14562/10341706.html

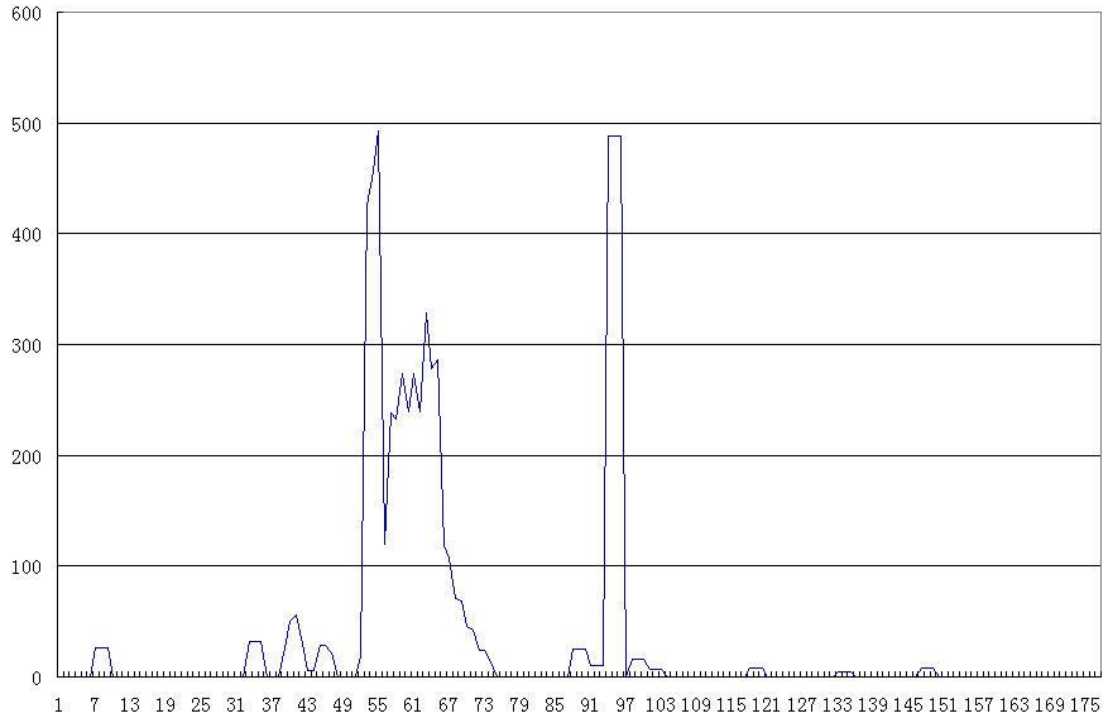

图 9. 工大校内新闻，正确文本区域行号为 52——74

http://today.hit.edu.cn/articles/2009/10-27/1015135739.htm

由上述行块分布函数图可明显看出，正确的文本区域全都是分布函数图上含有最值且连续的一个区域，这个区域往往含有一个骤升点和一个骤降点。

于是，网页正文抽取问题转化为了求行块分布函数上的骤升骤降两个边界

点，这两个边界点所含的区域包含了当前网页的行块长度最大值并且是连续的。用数学语言描述就是：

求正文区域所在的起始行块号 X_{start} 和终止行块号 X_{end}（ X 为行号，Y(X)是以 X 为轴的行块长度 ），需要满足以下四个条件：

1、$Y(X_{start}) > Y(X_t)(Y(X_t)$是第一个骤升点，骤升点必须超过某一阈值)；

2、$Y(X_n) \neq 0(n \in [start+1, start+K], K$是行块厚度, 紧随骤升点的行块长度不能为0，避免噪声)；

3、$Y(X_m) = 0(m \in [end, end+1]$，骤降点及其尾随的行块长度为0，保证正文结束)；

4、$\exists X$,当取到$\max(Y(X))$时,$X \in [X_{start}, X_{end}]$(保证此区域是取到行块最大值的区域)。

我分别用 Java 和 Perl 实现了上述算法。

六、系统评价（主要针对中文）

从国内主流网络媒体如新浪、搜狐、腾讯、网易等网站随机抓取有关经济、政治、军事等主题类网页各 300 篇做测试，经实验对比，除个别对话类文章在边界语句处有问题外，系统的正文提取准确率在 95%以上，可移植性和通用性都达到了实用水平。

系统特点如下：

- 与 HTML 是否良构无关；
- 不用建立 Dom 树，与 HTML 标签无关；
- 只需求出行块的分布函数即可抽取出正文；
- 只需对脱过标签的文本扫描一次，处理效率高；
- 去链接群、广告信息容易；
- 扩展性好，通用抽取采用统计方法，个别网站辅以规则，做到统计与规则相结合。

七、结论

1000 篇主题类网页（每篇网页的 HTML 源码至少是 1000 多行）的正文抽取时间，含 IO 操作，一共用时 21.29s，准确率在 95%以上。实验结果说明，这种基于行块分布函数的抽取方法对中文主题类网页有很好的通用性和较高的准确性。这种好的效果主要依赖于以下推论：

- ✓ HTML 每一行都表示一个完整的语义；
- ✓ 正文代码在物理位置上会靠的很近；
- ✓ 正文代码的一行中大都是文字；
- ✓ 正文代码的一行中非 HTML 标签的文字数量较多；
- ✓ 正文代码的一行中超链接长度所占比率不会很大。