**利用机器学习的正文抽取开源工程：https://github.com/ziyan/spider**

1. **Web Document Text and Images Extraction using DOM Analysis and Natural Language Processing.pdf**

Main content，根据文本块的长度以及公共父结点来定义main content。对于image的抽取，非常强调image caption抽取，然后对image caption进行命名实体识别，和正文段落进行相似性匹配，然后将图片和文本段进行相匹配。**对有效图片召回影响不大，不过这同样带来一个后期需要解决的问题，就是文章和图片并不是瀑布流的形式进行，有可能是是类似下面的效果：**



1. **Eliminating Noisy Information in Web Pages for Data Mining**

以site为单位，构建dom树，通过找到相同tag，来剔除潜在的广告信息。

1. **一种短正文网页的正文自动化抽取方法**

该方法先将网页进行分类，分成：长文本网页和短文本网页。然后有针对性的处理短文本网页。短文本网页也是类似采用单节点文本密度的方式进行采集。

**通过这篇论文启发到：短文本和正文图片在html中的距离应该不远，所以在定位到短文本后，可以通过距离来进一步定位正文图片。考虑到百度优质数据相当于有人手动替我们标注样本数据。**

**对于推荐和正文的过滤，可以通过计算子树的相似度来进行过滤聚类。**

**考虑一下html5，对于html5网页需要考虑特殊的html5标签，比如section，article等**

**采集一下网页，分析一下正文中，属性命名的规律性**

1. **Web Content Extraction – a Meta-Analysis of its Past and Thoughts on its Future**

Survey性质的文章，里面给了一些经典的算法论文和开源代码实现。提示了html5、js、css的重要性。

重点看下利用启发规则的算法，以及如何联合这些启发规则的算法。

1. **Boilerplate Detection using Shallow Text Features**

Many features that can be used for the classification of Web page segments have already been described [15, 16, 23, 26].一些正文提取属性。

Tag在html源码中的位置也是很重要，因为太靠后一般不会是正文。

非正文周围的信息（文字/图片也可能是非正文的），所以，给图片打权重时，可以将这个进行量化

1. （博士论文）EXTRACTING THE MAIN CONTENT FROM WEB DOCUMENTS

属性构造：

In order to do the classiflcation tasks, we need to collect all necessary features from  
DOM nodes or other features than can be constructed from the existing features. Most  
of the features that we used are from DOM properties and similar to Song et al [19],  
they used the features to classify segment importance into three levels of importance.  
In addition to that, we added some new features other than DOM properties namely  
stop word ratio, header around, and DOM height. We also incorporate link to text  
ratio [22] to the features. The features that we present here are used for both single  
phase classifler and two-phase classifler.

**innerTxtLength**The innerTxtLength is the length of all of the text string in all of the text nodes in a  
DOM node. The length of the inner text in a DOM node can be an indication that  
it contains main content.  
**innerHtmlLength**The innerHtmlLength is the length of all HTML string between start tag and end  
tag of an object. The length of the inner HTML in a certain DOM node can be  
used as heuristic to determine whether a DOM node is a good segment in segment  
classiflcation or main content in content classiflcation.

**imgNum**The imgNum is the number of image elements inside a DOM node. Typically, headers  
and footers of web pages have images without text elements on it. Meanwhile, the  
main content sometimes has image in it as an illustration. Therefore, number of  
images can be used as a feature to determine whether a DOM node is a main content.

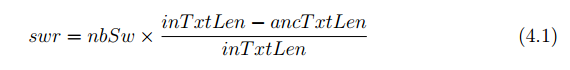
**interactionNum**The interactionNum is the number of input, select elements inside a DOM node.  
HTML elements such as input, select usually used to provide interaction between  
the user and the application. These kinds of interaction components are rarely part  
of the main content. Therefore, it could be useful as a feature to distinguish whether  
an element is a main content or noisy content.

**formNum**The formNum is the number of form element inside a DOM node.  
**optionNum**The optionNum is the number of option element inside a DOM node. The motivation  
to choose optionNum as a feature also similar to interaction components and forms  
since options only provide interaction between the user and the application which  
usually is not the part of the main content.

**tableNum**The tableNum is the number of table, tr, td elements inside a DOM node. Before  
the extensive use of div tag and Cascading Stylesheet (CSS) in the modern web  
documents, tables are often used to deflne the layout structure of the document and  
as a placeholder for main contents. However, sometimes we still can flnd ad hoc web  
pages that still use tables for layout purposes. **Therefore we use this as a feature  
to determine whether a DOM node is a good segment and whether it contains main  
contents.**

**paraNum**The paraNum is the number of p element inside a DOM node. In web documents  
such as blogs, news websites, the main content usually is placed inside the paragraph  
element.  
**linkNum**The linkNum is number of anchor element (hyperlink) inside a DOM node. If a DOM  
node contains quite a lot of links most likely it could be a navigational elements and  
not a main content.

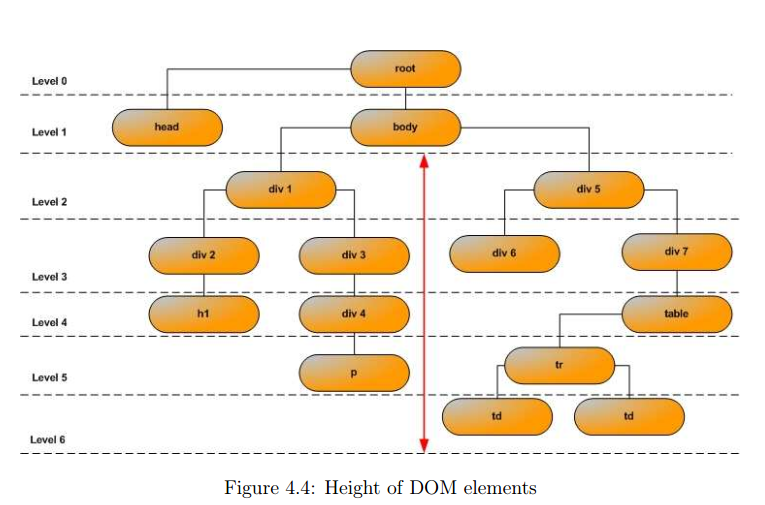
**divNum**The divNum is the number of div elements inside a DOM node.In the modern web  
documents, div tags are frequently used to deflne the layout structure of the document. **The number of div tags inside a DOM node can be used as a hint whether it  
is a good segment and main content.**

**stopWordRatio**The stopWordRatio is the number of stop words that contained in all of the text nodes  
of a particular DOM node. Based on our observation, typically, there are many stop  
words occur in the main content. However, there is a possibility that the stop words  
occur in the noisy contents such as recent comments section of blogs which rich of  
anchored text. Therefore, we introduce a penalty for stop words. Given the inner  
text length (*inTxtLen*), anchored text lengh (*ancTxtLen*), and number of stop words  
(*nbSw*), the stopWordRatio (*swr*) is calculated using the formula in Equation (4.1).  


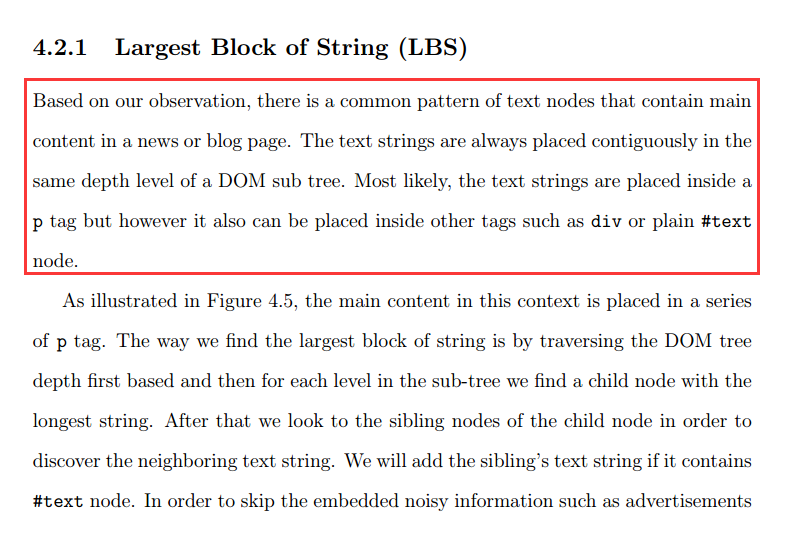
**linkToTextRatio**The linkToTextRatio in a DOM node is the ratio between the length of the anchored  
text and the total length of the text in the DOM node. A DOM node which has high  
value of link to text ratio most likely would be navigational elements and not main  
contents.

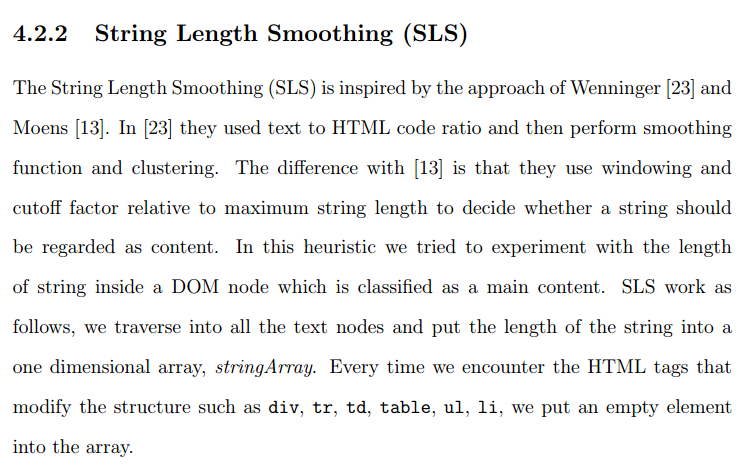
**tagName**The tagName is the name of the HTML tag of the DOM node. Different HTML tags  
usually have different purposes for document presentation. For example, li and ul  
tag usually used to make a linked list of hyperlinks or navigational elements while  
div and table used to deflne the document layout structure and content. Therefore,  
HTML tag can be used as a feature to identify whether a DOM node is a good  
segment and whether it contains the main content.

**domHeight**The domHeight is the maximum depth that can be reached a particular DOM node  
to a certain leaf node. The example of DOM tree with its associated depth level  
is shown in Figure 4.4, in the flgure, the DOM height of element body is flve. The  
explanation as follows, from body element, we can reach four leaves, namely h1 (at  
Level 4), p (Level 5), div 6 (Level 3), and two td elements (Level 6). Thus, the  
maximum depth is the path from body to td elements which give flve units height.  
In order to do segment classiflcation, we should select a certain DOM node which  
has good granularity level. The DOM height could be used as a feature to determine  
whether a DOM node is a good segment or not.

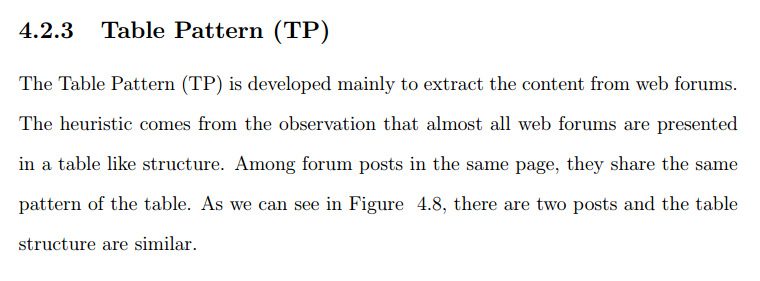


**headerAround**This feature indicates whether there are any header elements near a particular DOM  
node. In order to do this, we check the parent node, sibling nodes, and children nodes  
of the DOM node whether any header element exist. In this case, the header element  
is denoted by h1, h2, h3, h4, and h5 tags. Based on our observation, most of the main  
contents such as news articles, blog posts have title around them. The title usually  
represented by header tags.

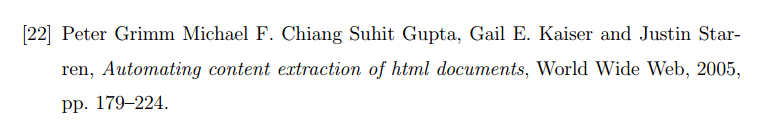




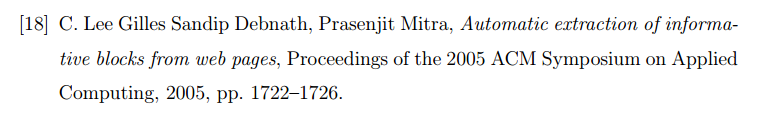
该方法构造一个一维向量以后，然后利用平滑技术，对长文本周围的短文本进行平滑处理，最后根据一个阈值判断哪些结点是否是正文结点。阈值选择是根据标准方差：We followed [23] to use the standard deviation of the string xlength as the threshold value.



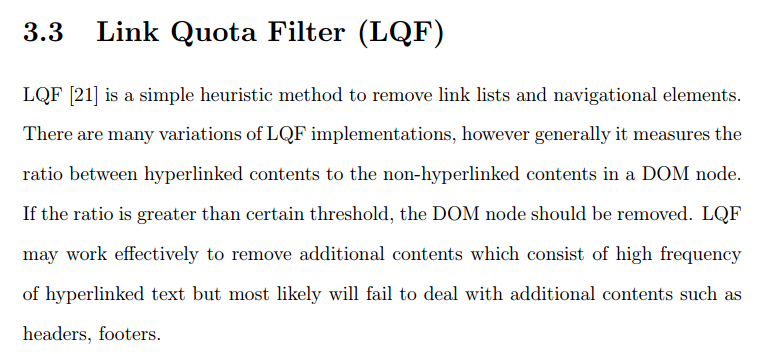
**Table Pattern是针对论坛页面处理的**



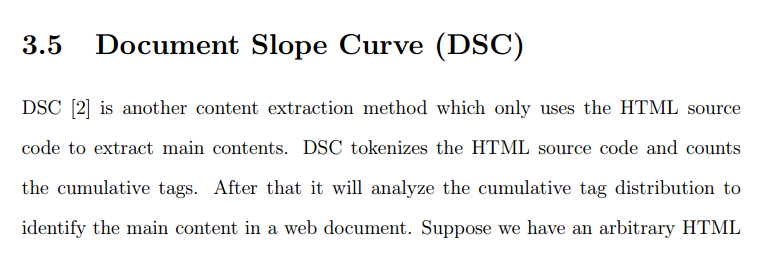
[22]这篇文章是利用关键词进行页面分类的



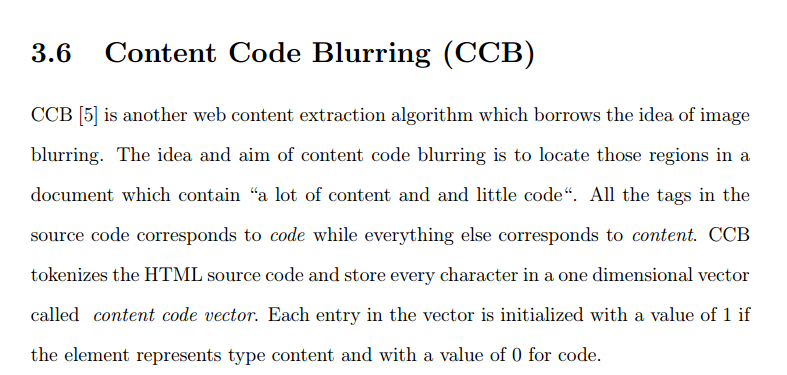
[18]利用临近结点的属性进行提取（这个描述不对，不知道为什么这么描述）这篇文章就是K-feature Extraction



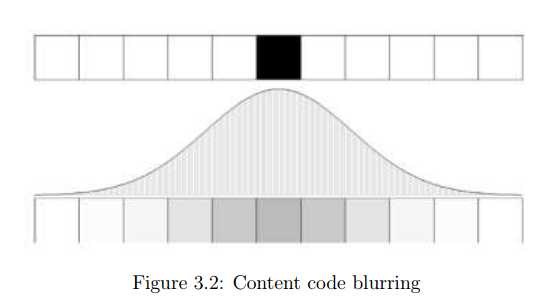
通过hyperlinked text比例来过滤addition content。可以统计图片周围这些文本的情况。这个属性对于header，footers识别效果不好。



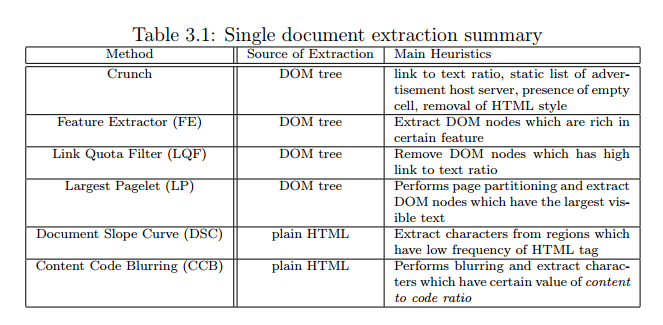
3.5表示通过累加计算相应tag数量变化，对于变化平缓的，可能是正文区域

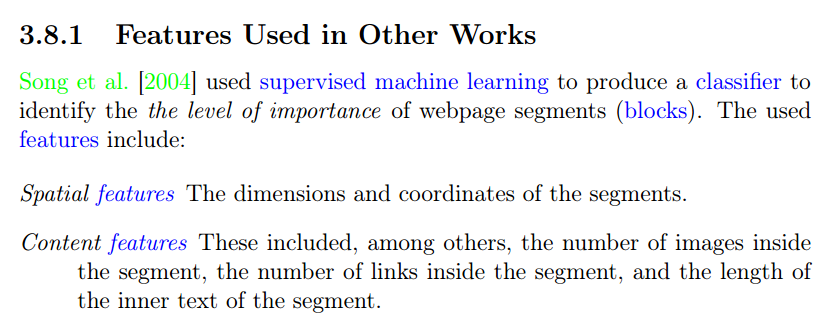


3.6 Content Code Blurring是将code和content进行0/1编码，然后利用函数进行调整。



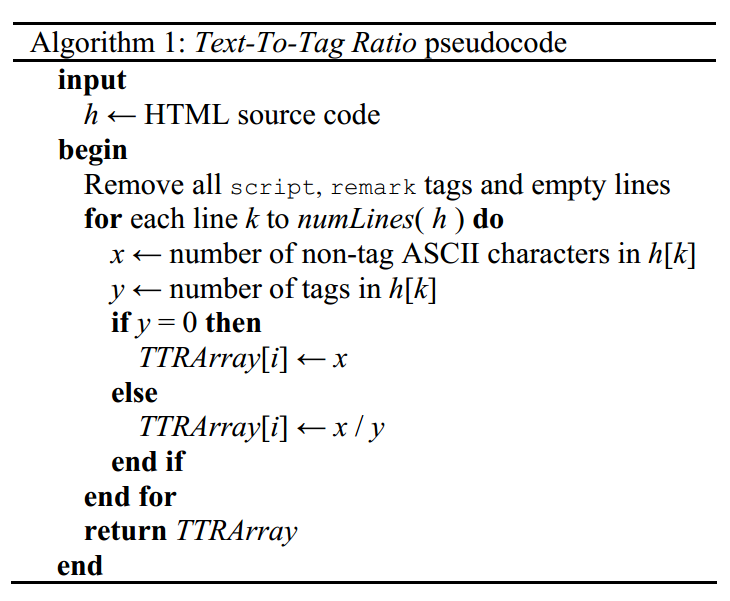
中间灰色部分的将是正文。





**Text Extraction from the Web via Text to Tag Ratio**

It, essentially, is the ratio of the count of non-HTML-tag characters to the count of HTML-tags per line. In the likely event that the count of HTML-tags on a particular line is 0 the ratio is set to the length of the line. The TTR algorithm is described in Alg. 1.



注意，上面是统计每行文字和tag的比例。而不是以结点为单位。

Before TTRs are computed, script and remark tags are removed from the HTML document because this  
information would be treated as non-tag text by the algorithm and thus skew the results. Empty lines are  
also removed because their inclusion would potentially skew the performance of the clustering algorithms that are described in Section IV.

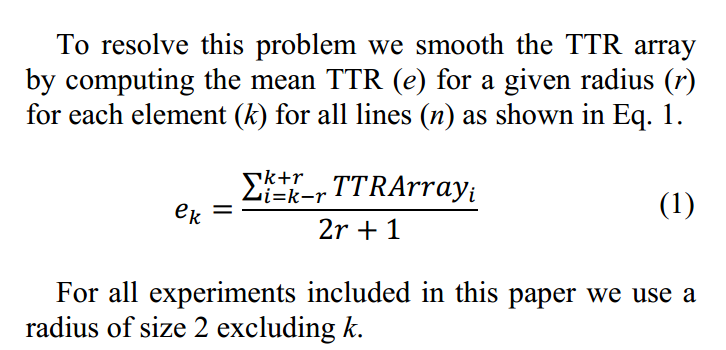
相对于平均值：

For each *k* in *TTRArray*, the higher the TTR is for an element *k* relative to the mean TTR of the entire array the more likely that *k* represents a line of content-text within the HTML-page.

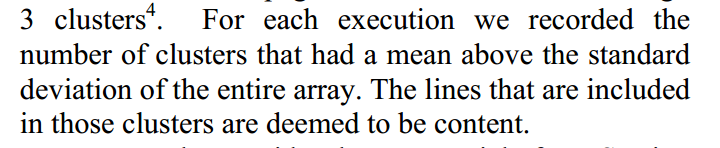
The standard deviation is used as the clustering threshold because it best represents the variance of the array while the mean is more likely to be skewed by large outliers and the median is more likely to pick a point that is too low because of the preponderance of low TTR scores among the lines of most Web pages.

做平滑的原因：

Before any clustering techniques are applied to the array a smoothing pass is made on the array. This is done because without smoothing many important content lines might be lost including the article title, by-line, short or one sentence paragraphs, etc. that would otherwise fall below the clustering threshold.



被看作main content的条件：

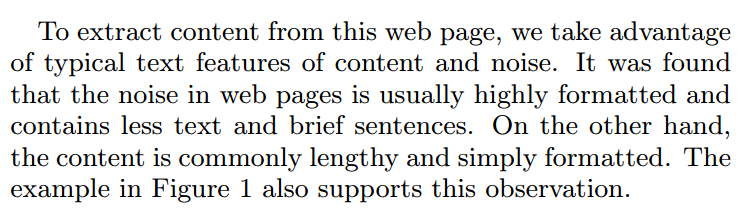


利用text to tag ratio的前提条件是：

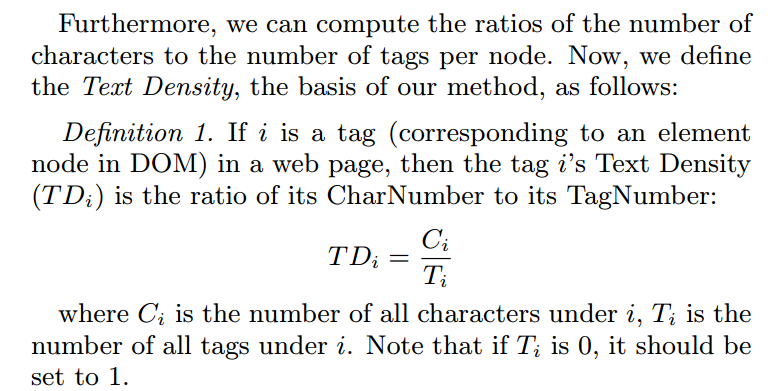
Our final clustering approach is very similar to the threshold clustering discussed in the previous subsection, in that, prediction clustering operates on the assumption that the TTR of the content lines of Web pages appear in stark contrast to the TTR of noncontent lines.（从文字比率分布上，正文和非正文区分度比较高）

**DOM Based Content Extraction via Text Density**

通过Text Density观察到的特点：



Text Density的定义如下：



In the likely case where the number of tags under a particular node is 0, the density is set to the number of characters the node contains.

注意是通过中序遍历获得结果的

Img元素包含的非常规属性的个数。

**Empirically, we set the <body> tag’s text density as the threshold in our study. 将body的参数值作为参考。因为包含最多的正文文本，同时也包含最多的非正文文本。**

DensitySum, is based on the observation that a content block of a web page belongs to an ancestor node in the structure of DOM; and as mentioned before, the text density of content nodes is much higher than that of noise nodes.  
Therefore, the content block (*i.e.*, a node in DOM) will get a peak value if we add up its children’s text densities. Here, we define DensitySum as below。

文本密度阈值的定义：

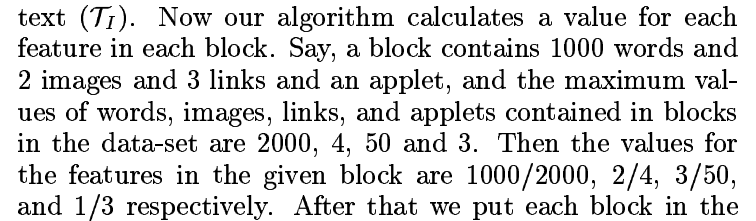
To resolve this problem, the algorithm first finds the maximum DensitySum tag in the whole page, without thresholding. Next, we set the minimum text density of the node in the path from the maximum DensitySum tag to <body> tag as threshold. The simple process of content extraction using DensitySum can be seen in Algorithm 2

计算图片和潜在标题或者重要文本的距离，比如相隔多少个TextUnit或者picture

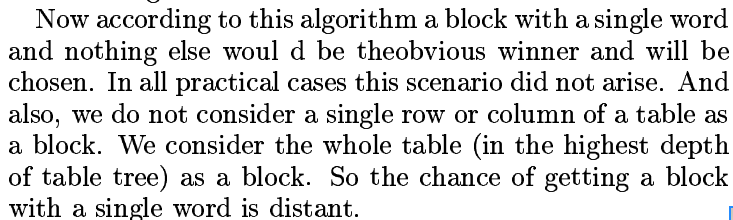
首先通过多页面线下挖掘一批非正文的图片，比如logo/icon/广告等信息。

**K-Features Extraction**

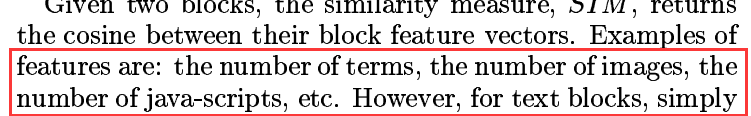
将各个计算得到的属性值进行归一化或者说规整化



解释了一些特殊情况的处理方法

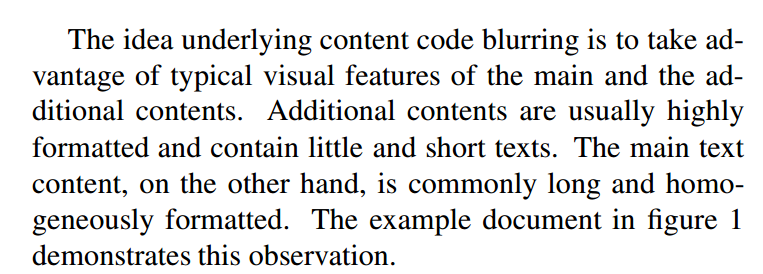


**属性：**



K-Feature Extraction就是讲述构造特征向量，然后计算相应阈值，大于阈值时，视为main content。同时对于多个main content，利用聚类算法进行聚类。

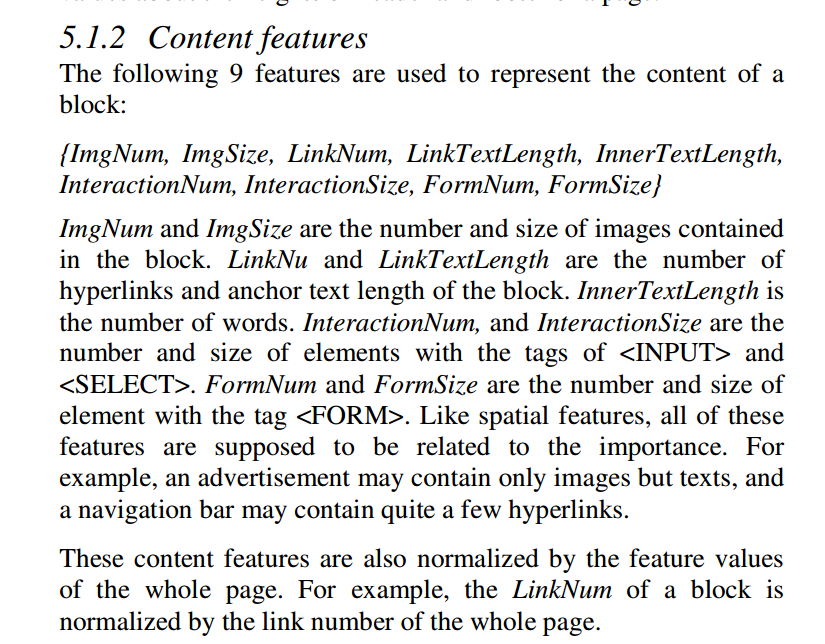
**Content Code Blurring: A New Approach to Content Extraction**



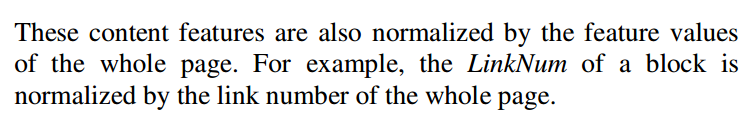
**这个文章的思路是将code和content进行tokens分割，然后，根据这个形成的向量，进行处理**

Learning Block Importance Models for Web Pages

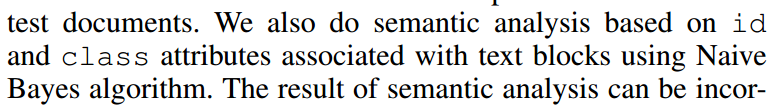
该算法是利用VIPS进行分割页面，然后进行判断的，对pic url提取影响不大。



正则化



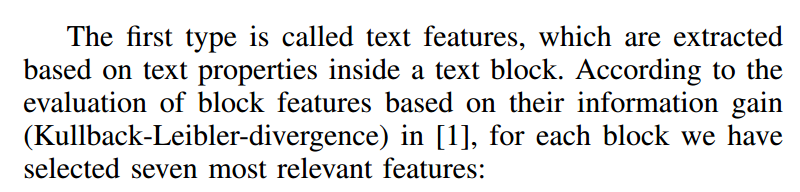
**A Machine Learning Approach To Webpage Content Extraction**



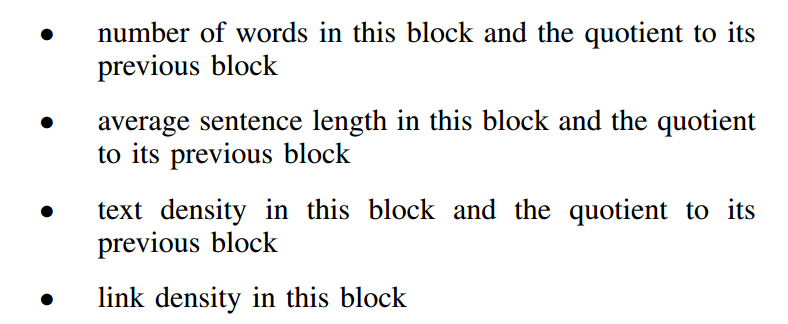
利用机器学习的方法进行main content的抽取。

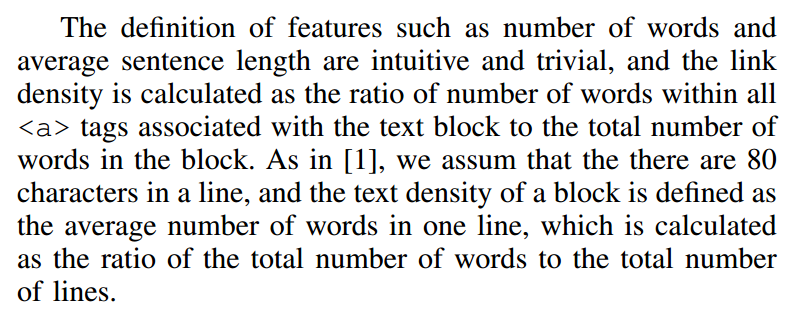
提取img到body的id列表/class列表，判断该列表中包含main content意思的关键词数量。

对文本属性的处理，利用信息熵

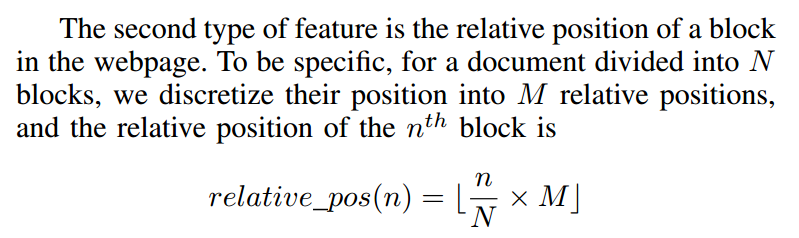


文本属性如下：（quotient to its previous block，感觉这种类型的特征对于抽取图片没有什么影响。）

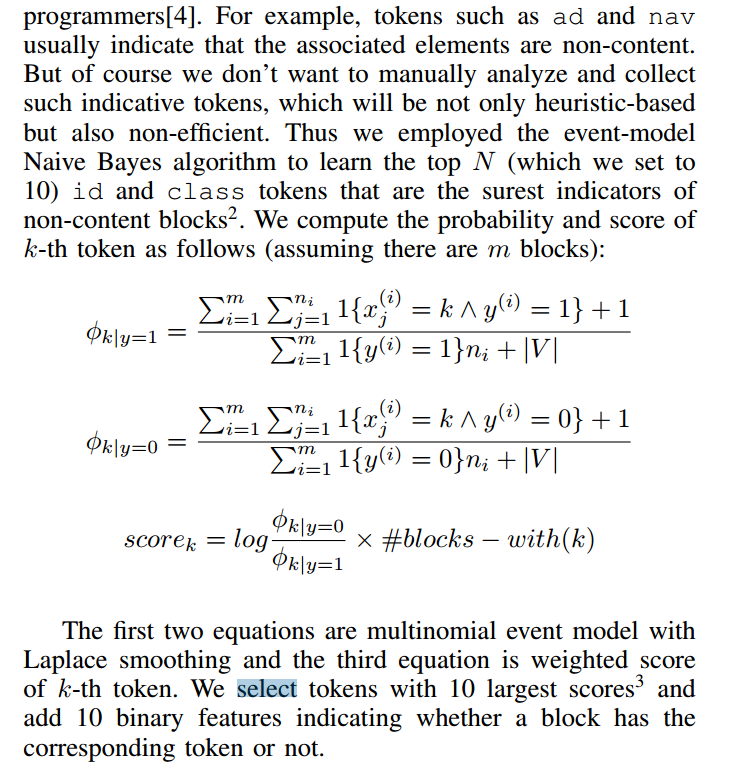




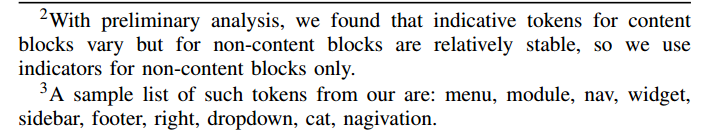
位置属性：



利用机器学习的方法获取none-content的id或者class。



为什么只学习none-content，因为id或者class属性对none-content辨识度比较高，对于content辨识度不高。

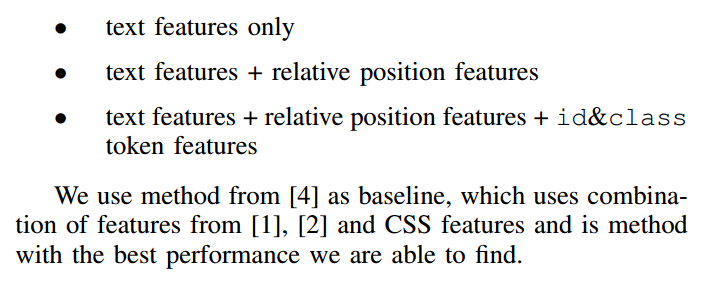


常见的none-content的id或者class:

A sample list of such tokens from our are: menu, module, nav, widget, sidebar, footer, right, dropdown, cat, nagivation.

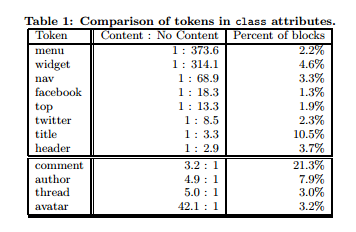
将获取的属性值作为单独的列，转换成one-hot编码。

训练模型时，利用属性进行训练时，训练的步骤：



Content Extraction Using Diverse Feature Sets

利用机器学习进行web content抽取，下表是包含重点内容的属性名称。



可能是水论文Main Content Extraction from Web Documents Using Text Block Context

提供了text block的上一个text block距离是指text block之间含有的tag数量，包含opentag和closetag

问一下图片文件名是否是有意义单词的技术。

图片后缀名。Icon可能较少是正文图片，除了专业的网站。

**页面中图片的Last-modify，可以用来纠正召回率的时效性。比如页面的时间比图片的时间晚，则应该以图片的时间为准。**