CETR - Content Extraction via Tag Ratios

Tim Weninger [*]
Dept. of Computer Science
University of Illinois at
Urbana-Champaign
Urbana, IL
weninge1@illinois.edu

William H. Hsu
Computing and Information
Sciences Dept.
Kansas State University
Manhattan, KS
bhsu@cis.ksu.edu

Jiawei Han
Dept. of Computer Science
University of Illinois at
Urbana-Champaign
Urbana, IL
hanj@cs.uiuc.edu

ABSTRACT

We present Content Extraction via Tag Ratios (CETR) – a method to extract content text from diverse webpages by using the HTML document's tag ratios. We describe how to compute tag ratios on a line-by-line basis and then cluster the resulting histogram into content and non-content areas. Initially, we find that the tag ratio histogram is not easily clustered because of its one-dimensionality; therefore we extend the original approach in order to model the data in two dimensions. Next, we present a tailored clustering technique which operates on the two-dimensional model, and then evaluate our approach against a large set of alternative methods using standard accuracy, precision and recall metrics on a large and varied Web corpus. Finally, we show that, in most cases, CETR achieves better content extraction performance than existing methods, especially across varying web domains, languages and styles.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: [Information filtering]; H.3.1 [**Content Analysis and Indexing**]: [Abstracting methods]; I.5.3 [**Clustering**]: [Algorithms]

General Terms

Algorithms, Experimentation

Keywords

Content extraction, tag ratio, world wide web

1. INTRODUCTION

As the Internet matures the amount of data available continues to increase. The artifacts of this ever-growing media provide interesting new research opportunities that explore social interactions, language, art, politics, and so on. Many of these new research directions require the content of the Internet to be gathered, processed and stored quickly and efficiently. These efforts are often hampered by the inclusion of non-content text and images, *i.e.*, navigation links and advertisements. Furthermore, HTML tags and other non-content related HTML characters – images not included – typically comprise the majority of each page's size, and yet, web crawlers are required to download, store and compute each webpage in its entirety. In order to effectively manage this ever-growing and ever-changing media, content extraction methods have been developed to remove extraneous information from webpages.

When beginning this investigation, we observed that the typical webpage contains a title banner (or something similar) towards the top of the page, a list of hyperlinks on the left or right side of the page with advertisements interspersed. Most usually the meaningful content of the page is located in the middle. Of course, this layout is not standard among all webpages; therefore a flexible, robust content extraction tool is necessary.

T.V. Raman recently observed that in newer webpages, "...there is a clean architectural separation among content, visual-presentation, and interaction layers" [31]. Our observations concur with those of Raman; specifically, we find that modern webpages have largely abandoned the use of structural tags within a webpage and adopted an architecture which makes use of stylesheets and `<div>` or `<span>` tags for structural information. While this is a welcome advancement for many reasons (*e.g.*, ease of development, more conducive to AJAX-oriented design) one aspect which has failed to keep up with these changes is content extraction methods. Most current content extraction techniques make use of particular HTML cues such as tables, fonts, sizes, lines, etc., and since modern webpages no longer include these cues, many content extraction algorithms have begun to perform poorly. One difference between our approach and other related work is that we make no assumptions about the particular structure of a given webpage, nor do we look for particular HTML cues. We only assume that a given webpage maintains *some* structure.

This is a nontrivial task because of the difficulty in determining which part of a webpage is meaningful and which part is not. Our solution, called Content Extraction via Tag Ratios (CETR - pronounced cedar), is partially based on previous work in Web content extraction [32]. In the CETR algorithm we construct a tag ratio (TR) array with the contention that for each line i in the array, the higher the tag ratio is for the ith line the more likely that i represents a line of content-text within the HTML document.

In this and in previous work [32], we observe that the TR array can be represented as a histogram, wherein each histogram bucket represents the tag ratio of a line in the document. By that observation the problem is reduced to a histogram clustering task wherein appropriate clusters should

[*] Weninger performed early work on this topic at Kansas State University, Manhattan, KS

WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA.
ACM 978-1-60558-799-8/10/04.

discriminate between TR-lines which correspond to webpage content and those TR-lines which do not. Three clustering approaches are investigated in this work. The first two approaches either apply a water-level *i.e.*, minimum cutoff (CETR-TM) or a partition clustering approach (CETR-KM) which operate on similarities from the tag ratio alone. We find that the TR-histogram is not just a set of values, but is rather an *ordered sequence* of values wherein additional information may be gained by examining the surrounding values and the manner in which these values evolve as the sequence is iterated through. By this intuition, we expand our model to include the sequence information by way of an absolute derivative array. The result is a fast, accurate and general content extraction algorithm which outperforms current, even supervised and specialized, approaches.

Contributions. Five main contributions can be claimed in our paper:

1. The introduction of Content Extraction via Tag Ratios (CETR).
2. Methods for clustering the Tag Ratio histogram in one dimension: CETR Threshold Method (CETR-TM) and CETR k-Means (CETR-KM), and discussion as to why we believe this approach to be limited.
3. A novel two dimensional webpage content model and its application to the content extraction task.
4. A tailored partition clustering approach designed to operate on the two dimensional webpage content model.
5. An empirical study which compares CETR with 10 alternate content extraction approaches across a large and varied corpus.

Orgainization. The remainder of this paper first discusses related work and applications before describing the CETR algorithm and giving examples of it's use. We pay special attention on the smoothing, two-dimensional model and clustering methods, as well as possible worst case scenarios. Three distinct algorithms are presented in this paper. The first two (CETR-TM, CETR-KM) are preliminary versions of the approach and the final version (CETR) is claimed to be the most advanced and best performing. We test each algorithm against many of the techniques examined in the related works section and discuss the results. Finally we offer our conclusions and plans for future research.

2. RELATED WORK

Extracting content from HTML documents has been well-studied and numerous methods have been developed.

Perhaps the most simplistic approaches are seen in hand-crafted web scrapers which specifically know how to extract article text by looking for known HTML-cues with regular expressions written in Java or Perl or with specialized tools designed for content extraction such as NoDoSE [2] or XWRAP [4]. An obvious disadvantage of this approach is that different rule expressions need to be manually created for each website. Furthermore, an individual website may also change its structure or layout over time making this approach in need of continuous maintenance.

The term *Content Extraction* was introduced by Rahman et al. [30] in which the authors describe a basic content extraction algorithm. Shortly thereafter Finn et al. [13] introduced the Body Text Extraction (BTE) algorithm wherein the authors extract content-text by identifying the single, continuous region which contains the most words and the least amount of HTML tags. Gottron [14] applied the Document Slope Curves (DSC) [29] extension to the BTE algorithm to create Advanced DSC (ADSC) in which a windowing technique is used to locate document regions in which word tokens are more frequent than tag tokens.

Mantratzis et al. presented an approach to identify navigation lists by identifying DOM elements which have a high ratio of text residing in anchor tags [25]. This aptly named Link Quota Filter (LQF) approach can be applied to content extraction by it's inverse, that is, by removing the resulting link blocks from the document.

Han et al. developed the Largest Size Increase (LSI) algorithm [20] which identifies the DOM subtree which contributes most strongly to the visible content in a rendered document.

Debnath et al. developed the FeatureExtractor (FE) [11] and K-FeatureExtractor (KFE) [12] approaches based on block segmentation of the HTML document. Each block is analyzed for particular features like the amount of text, images, script code, etc. Content text is extracted by selecting blocks which meet some criteria, *e.g.* most text.

Gottron presented an approach most similar to CETR by way of Content Code Blurring (CCB) [16], wherein content regions are identified by homogeneously formatted source code character sequences.

An attempt to combine different content extraction methods into one system was made by the Crunch framework [18, 19, 17]. Crunch showed that a combination of different methods can provide better results than a single approach on its own. A more recent ensemble method called the CombineE framework [15] was recently developed to more easily configure ensembles of content extraction algorithms.

Yet another approach is to induce a wrapper from labeled examples. One such approach was studied by Kushmerick [22], however this approach could not handle complex or unexpected structures. Muslea et al. [27] present a similar approach by taking a hierarchical description of the fields to be extracted along with user defined labels on example documents in order to induce a set of extraction rules. However, like the manual or pattern matching approaches mentioned above, wrapper induction techniques still require up-to-date, tediously labeled examples for each data source.

The Visual Page Segmentation (VIPS) [6] heuristically segments documents into a tree where the nodes are visually grouped blocks. The major problem with this approach is that the result of the VIPS algorithm does not label the nodes as content or non-content. The results presented in later sections show that if the best possible parameters are selected and a perfect mechanism is provided to label the nodes then VIPS can extract article text with a high degree of accuracy. However, there exists no such labeling mechanism; furthermore, VIPS must partially render a page in order to analyze it including retrieving all external style sheets, etc. Therefore, compared to other techniques, VIPS is very resource intensive.

Template detection algorithms [23, 33, 21, 10, 7, 9] are a different approach to content extraction in which collections of training documents based on the same template are used to learn a common structure. Specifically, Bar-Yossef et al. present an approach which automatically detects templates from the largest pagelet (LP) [3]. In general template detection algorithms find the main content by removing identical parts across all web documents. This is an accurate approach but has been found to be too time consuming and burdensome because a model must be built for each individual website and therefore for each site multiple pages known to have the same template are required. In the CleanEval content extraction competition only a few pages are available from the same site thus mandating a more general approach.

The winner of the CleanEval task [26] splits pages by their tags into a sequence of blocks and then labels each block as content or non content using conditional random fields with a number of block-level features.

A hybrid approach of the heuristic and supervised learning methods is the Maximum Subsequence Segmentation algorithm (MSS) by Pasternack and Roth [28] wherein they extract content by a "method of global optimization over token-level local classifiers." Despite being a supervised learning approach, MSS seems to be less susceptible to the problems of similar approaches because it bases its learning largely on character sequence statistics rather than on specific tags. However, MSS still requires training and is therefore susceptible to bias from the training examples which is evident by it's results. For example, when trained on news article data MSS can extract news article content quite well, but when that model is given data such as the CleanEval corpora the performance suffered significantly (results not reported). Only after several adjustments were made were the CleanEval results reported.

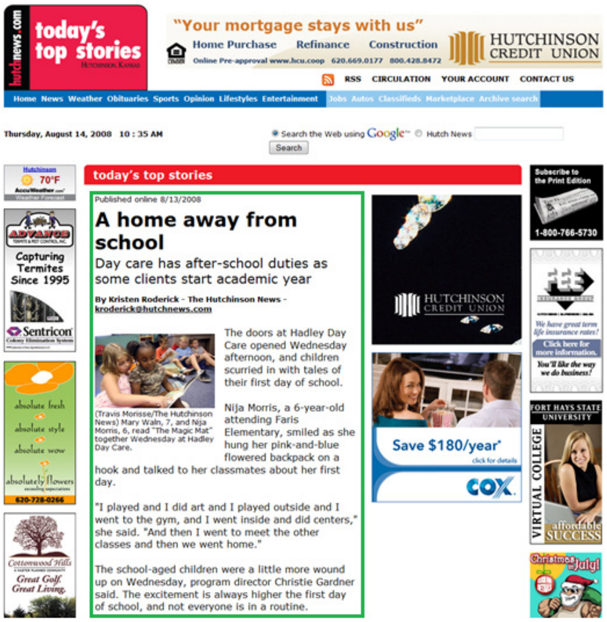
2.1 Applications

There are a number of applications where content extraction is an essential task or could improve overall performance. Pocket-sized devices with small screens such as mobile phones or PDA's are ubiquitous and therefore adapting webpages for these devices is an important task [4, 8]. Other tasks include automatically generating RSS news feeds from blogs or article pages. The general field of information retrieval may benefit from this work: by removing irrelevant text from a webpage a keyword-based search is less likely to return irrelevant hits. For example, Cai et al [5] increased IR performance when his VIPS algorithm was employed to process webpages' visual blocks separately. VIPS is used again to aid in query expansion by segmenting webpages and selecting additional query terms from only the "best" blocks.

Specifically, in a future task we wish to employ a general content extraction method as a preprocessing step to *clean* input text to subsequent steps in a pipeline. For example, an interesting research task aided by content extraction tools could be named entity extraction, disambiguation and reconciliation wherein we wish to infer relationships between entities by their semantics and relative location in the webgraph. In order to save such a system from the onslaught of irrelevant entities confounding the model, we realize the absolute necessity for a fast, accurate, general purpose content extraction algorithm.

3. TAG RATIOS

Let's take, as a running example, a news article from The Hutchinson News[1] that appeared on Wednesday, March 19, 2008 and is shown in Figure 1. This webpage is similar to many pages on the Web; the title banner, navigation and advertisements take up most of the space on the page while the content of the page is confined to a relatively small space in the middle. At the bottom of the page more advertisements and images are displayed along with links to copyright and other administrative information.


hutchnews.com
today's top stories
Hutchinson, Kansas
"Your mortgage stays with us"
Home Purchase Refinance Construction
Online Pre-approval www.hcu.coop 620.669.0177 800.428.8472
HUTCHINSON CREDIT UNION
RSS CIRCULATION YOUR ACCOUNT CONTACT US
Home News Weather Obituaries Sports Opinion Lifestyles Entertainment Jobs Autos Classifieds Marketplace Archive search
Thursday, August 14, 2008 10 : 35 AM
Search the Web using Google Hutch News
Search
Hutchinson
70°F
AccuWeather.com
today's top stories
Subscribe to the Print Edition
1-800-766-5730
ADVANCE
Capturing Termites Since 1995
Sentricon
absolute fresh
absolute style
absolute wow
absolutely flowers
620-728-0266
Cottonwood Hills
Great Golf. Great Living.
HUTCHINSON CREDIT UNION
Save $180/year
COX
FORT HAYS STATE UNIVERSITY
VIRTUAL COLLEGE
affordable SUCCESS
Christmas in July!

Published online 8/13/2008

A home away from school

Day care has after-school duties as some clients start academic year

By Kristen Roderick - The Hutchinson News - kroderick@hutchnews.com

(Travis Morisse/The Hutchinson News) Mary Waln, 7, and Nija Morris, 6, read "The Magic Mat" together Wednesday at Hadley Day Care.

The doors at Hadley Day Care opened Wednesday afternoon, and children scurried in with tales of their first day of school.

Nija Morris, a 6-year-old attending Faris Elementary, smiled as she hung her pink-and-blue flowered backpack on a hook and talked to her classmates about her first day.

"I played and I did art and I played outside and I went to the gym, and I went inside and did centers," she said. "And then I went to meet the other classes and then we went home."

The school-aged children were a little more wound up on Wednesday, program director Christie Gardner said. The excitement is always higher the first day of school, and not everyone is in a routine.

Figure 1: The Hutchinson News webpage article

To extract the content from this webpage a naive approach would use regular expressions to remove all of the HTML tags from the document and return the result. This approach would achieve 100% recall, however all of the text advertisements, title, menus, etc. would remain.

The majority of the algorithms listed in Section 2 look for HTML cues which likely indicate a content section. For example, many algorithms look for specific structural elements of the webpage and match these elements to a set of rules to derive the content section. The shortcoming of these methods is that, with the widespread adoption of cascading style sheets in recent years, the structure of the webpage has become separated from the content (For an interesting review of this phenomenon see Michael Wesch's *The Machine is Us/ing Us*[2]). As a result, modern webpages have switched from using structural tags to mostly `<div>` tags with the structural information provided by the style sheets. With this change, most of the current extraction techniques perform poorly on modern webpages even if they previously performed well.

Of course, any new content extraction algorithm is still required to handle the old-style HTML markup. With this in mind, we studied the general features that the old and new paradigms have in common, and from this investigation we find that the number of tags per line of HTML markup has generally remained the same even though the type and function of those tags has changed. From this observation we developed the general concept of Tag Ratios.

Tag Ratios (TRs) are the basis by which CETR analyzes

[1] `http://www.hutchnews.com`

[2] `http://www.youtube.com/watch?v=NLlGopyXT_g`

a webpage in preparation for clustering. TRs, essentially, are the ratios 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 then the ratio is set to the length of the line. The TR algorithm is described in Algorithm 1 where D is the document being analyzed and T is the resulting histogram containing the tag ratios for each line i in D.

Before TRs are computed, `script`, `remark` and `style` tags are removed from the HTML document because this information would be treated as non-tag text by the algorithm and likely skew the results. Empty lines are also removed because their inclusion would potentially hinder the performance of the clustering procedure.

Algorithm 1 Compute Tag Ratios

```
INPUT: $D$
OUTPUT: $T$
$D \leftarrow removeScriptTags(D)$
$D \leftarrow removeRemarkTags(D)$
$D \leftarrow removeStyleTags(D)$
for all $i \leftarrow 1$ to $|D|$ do
    $x \leftarrow$nonTagChars$(D_i)$
    $y \leftarrow$tags$(D_i)$
    if $y = 0$ then
        $y \leftarrow 1$
    end if
    $T_i \leftarrow x/y$
end for
```

Computing the TR-histogram is a straight forward task as evident from the simplicity of Algorithm 1. Example 1 shows a snippet of code from an article published on The Hutchinson News' website with the corresponding tag ratios.

EXAMPLE 1. *Below is a brief snippet of a webpage news article.*

```
1. <div id="topnav">
2.   <div id="storyPageContent2">
3.   <div id="author">James Smith</div>
4.   OKLAHOMA CITY - Police were told that...
5.   ...The Oklahoman reported Sunday. <br><br> Jones had...
6. </div></div>
```

The Tag Ratios for these six lines are computed as follows:

1. Text=0, Tags=1, TR=0
2. Text=0, Tags=1, TR=0
3. Text=11, Tags=2, TR=5.5
4. Text=37, Tags=0, TR=37
5. Text=41, Tags=2, TR=20.5
6. Text=0, Tags=2, TR=0

The running time is linear in the number of HTML lines, that is, $O(|D|)$. Figure 2 shows the resulting TR-histogram T. We see that between lines 220 and 260 there exist lines with a relatively high tag ratio; intuitively, we acknowledge this high tag ratio portion to be indicative of the webpage's content.

4. THRESHOLD METHOD

In this section we describe the threshold partitioning technique. Originally described in [32] the principle of this approach is to determine a threshold τ which discriminates TRs into content and non-content sections. That is, any TR value greater than or equal to τ should be labeled content, and conversely, any TR value less than τ should be labeled not content. The problem then becomes a task of finding the best value for τ. Discussion on parameter tuning is in Section 6.5.

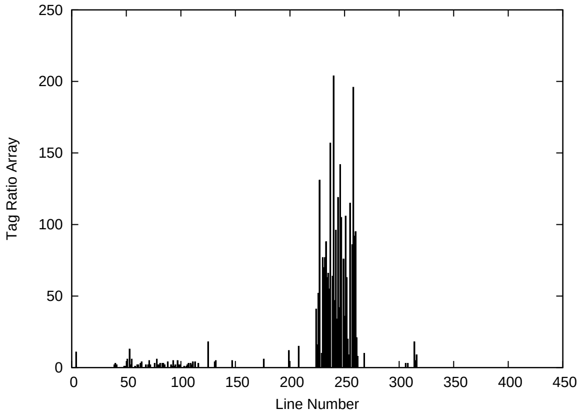


Figure 2: Tag Ratios line by line from Hutchinson News webpage article

4.1 Smoothing

After the TR-histogram T is calculated a smoothing pass is made on the histogram. This is done because without smoothing many important content lines might be lost. In our experience, these lost content-lines typically include the page title, the news article byline or dateline, short or one sentence paragraphs, or other lines such that the TR is abnormally different relative to the surrounding lines. As a pathological example, consider a webpage containing a document such as the American Declaration of Independence[3], which contains TR-spikes corresponding to the relatively long preamble and proclamation sections. However, many of the abuses of the king are listed in short, single sentence phrases, and relative to the rest of the document their TRs may therefore be errantly excluded in the clustering phase.

To resolve this problem we apply a Gaussian smoothing pass to T. Standard Gaussian smoothing algorithms are generally implemented for image processing, are continuous and thus do not suit our purposes. Therefore the algorithm used in this approach was re-implemented as a discrete function operating in a single dimension. Equation 1 shows the construction of a Gaussian kernel k with a radius of 1 standard deviation 1σ, giving a total window size of $2(\lceil\sigma\rceil) + 1$.

$$k_i = \sum_{j=-\lceil\sigma\rceil}^{\lceil\sigma\rceil} e^{\frac{-j^2}{2\sigma^2}}, 0 \leq i \leq 2(\lceil\sigma\rceil). \qquad (1)$$

The size of and values within k vary according to σ because as the variance of T increases, smoothing necessity also increases. Next, Equation 2 shows that k is normalized to form k'.

[3] e.g. http://www.ushistory.org/declaration/document/index.htm.

$$k'_i = \frac{k_i}{\sum_{j=0}^{\lceil\sigma\rceil} k_j}, \lceil\sigma\rceil \le i \le 2(\lceil\sigma\rceil). \quad (2)$$

Finally, the Gaussian kernel k' is convolved with T in order to form a smoothed histogram (T') as shown in Equation 3.

$$T'_i = \sum_{j=-\lceil\sigma\rceil}^{\lceil\sigma\rceil} k'_{j+\lceil\sigma\rceil} T_{i-j}, \lceil\sigma\rceil \le i \le len(T) - \lceil\sigma\rceil. \quad (3)$$

Compared to Figure 2, T', shown in Figure 3, is better suited for clustering because of the increased cohesiveness within sections and strict differences between sections. Furthermore, T' has a lower variance because outlying peaks and valleys are smoothed. Similarly, outliers, such as advertisements, that may occupy a single high-TTR line among many low-TTR lines, are smoothed to below the threshold.

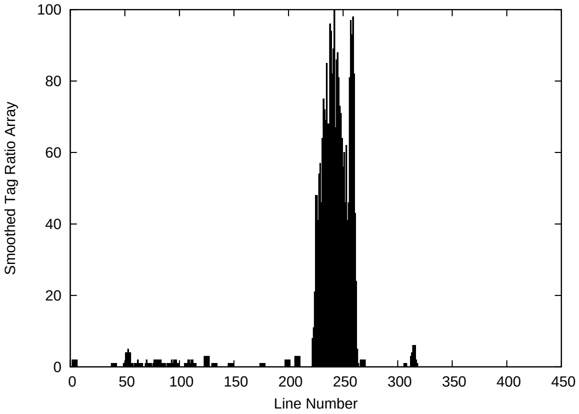


Figure 3: Smoothed Tag Ratios line by line of Hutchinson News webpage article

4.2 Selecting Content from the Threshold

Finally, let C be the set of content lines such that $D_i \in C$ iff $T'_i \ge \tau$ where $D_i \cong T'_i$ and $\tau \leftarrow \lambda\sigma$ where λ is a parameter and σ is the standard deviation. The parameter λ is discussed further in Section 6.5. After elements of C are selected, each content-line is stripped of all remaining HTML tags – usually paragraph and anchor tags. Then the cleaned lines are combined and output to a file for storage, indexing, etc. This threshold method is hereafter referred to as CETR-TM.

4.3 Selecting Content via Clustering

Alternatively, we apply the k-means clustering method to group content C and non-content N lines by using T' as the only similarity measure. Empirically, we set $k \leftarrow 3$. The resulting k clusters $S_1, S_2 \dots S_k$ are labeled by selecting the cluster which has its centroid closest to the origin (*i.e.* zero in 1-dimensional space) S_{min} and assigning $N \leftarrow S_{min}$. The remaining clusters are assigned to C. The content-lines in C are stripped of all HTML tags and output. This 1-dimensional k-means clustering method is hereafter referred to as CETR-KM.

5. 2D MODEL

One shortcoming of the Threshold Clustering and k-Means methods is that they view the TR histogram as a set of values rather than an *ordered sequence* of values, and as a result they are not sensitive to *jumps* in the TR histogram. This *ordered sequence* information should be considered in a general purpose algorithm because significant *jumps* in the histogram (moving left to right or right to left) provide more information on the borders of the content section(s).

This section presents a unique approach to clustering 1-dimensional histograms. We contend that by transforming the histogram data so that it may be represented in 2-dimensions we can capture the ordered nature of the histogram data and obtain more accurate results. For this task, we define the two dimensions to be (1) a smoothed TR histogram (T'), and (2) the absolute smoothed derivatives of the smoothed TR histogram ($\hat{G}$).

To compute G, first smooth T in the same manner as described in Equations 1-3 to get T'. Next, find the derivatives for each element in the array; specifically, we subtract T'_i from the mean of the next α elements in order to differentiate on the moving average as shown in Equation 4 instead of line-by-line. Note: all experiments presented in this paper use $\alpha = 3$.

$$f'(T'_i) = G_i = \frac{\sum_{j=0}^{\alpha} T'_{i+j}}{\alpha} - T'_i, 0 \le i < len(T') - \alpha. \quad (4)$$

Note that $len(G) \ne len(T')$. Instead $len(G) = len(T') - 1$ because G is essentially an array of differences. Next we again smooth G by way of Equations 1-3 to get G'.

Finally we compute $\hat{G}$ such that $\hat{G}_i = |G'_i|$ for each i in G'. These values are shown in Figure 4.

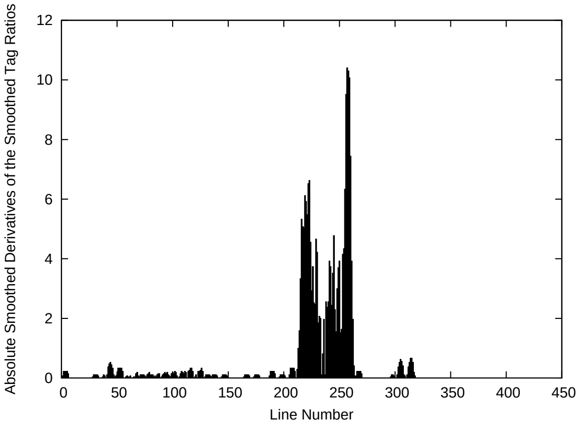


Figure 4: Absolute Smoothed Derivatives of Smoothed Tag Ratios line by line of Hutchinson News webpage article

Notice that there are two spikes in Figure 4. The first spike at line 220 corresponds to the beginning of the content section, and the second spike at line 267 corresponds to the end of the content section. In any given webpage there may exist more than one content section therefore a clustering method is needed to appropriately categorize our model.

5.1 Constructing 2D Tag Ratios

By combining the Smoothed Tag Ratios T' from Figure 3 and the Absolute Smoothed Derivatives of Smoothed Tag Ratios $\hat{G}$ from Figure 4 we observe that good clustering properties are revealed. As illustrated in Figure 5, if we manually label each point to be either content ($\times$) or non-content ($+$) we see that the dense collection of points near the origin are non-content lines and the remaining points are content lines. This 2D model presents a clear separation of content from non-content lines which can be explicitly obtained with the appropriate clustering technique.

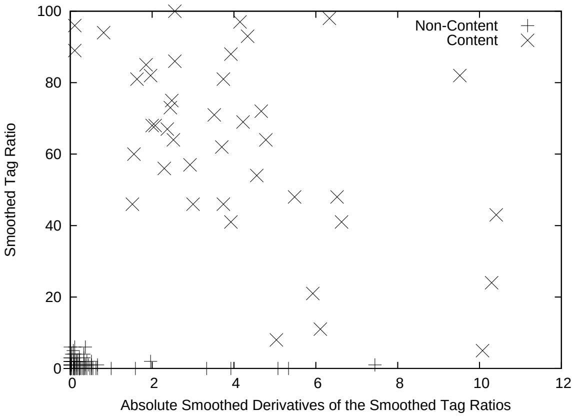


Figure 5: Scatterplot depicting Smoothed Tag Ratios by Absolute Smoothed Derivatives of the Smoothed Tag Ratios of the Hutchinson News webpage article. Manual labels of the data show content × and non-content + lines.

5.2 Clustering Tag Ratios

After the 2D model is created it is necessary to cluster the TR points $(T'_i, \hat{G}_i)$ into two sets: content (C) or non content (N). This section describes our clustering algorithm, which is based on the k-Means algorithm originally proposed by MacQueen [24]. The standard k-Means algorithm operates by assigning objects (*i.e.* $(T'_i, \hat{G}_i)$-points) to k clusters $S_1, S_2, \ldots, S_k$ randomly at first, and then by iteratively reassigning objects according to the cluster centroids' nearest neighbors. Empirically, we set $k = 3$.

Our approach to clustering is similar except that one cluster has a centroid which is always set to the origin. Specifically, we define m_i^j to be the centroid of S_i at iteration j and then force $m_1^j = (0, 0)$.

This approach is beneficial in 2 ways: (1) it forces the remaining clusters to migrate away from the origin where the non-content points are located, and (2) it provides an easy means for labeling the resulting clusters; specifically, the cluster with the origin-centroid will always be labeled non-content because points near the origin most likely represent non-content points, *i.e.* $N \leftarrow S_1$. All remaining clusters are therefore labeled content, *i.e.* $C \leftarrow S_2, \ldots, S_k$.

5.3 Implementation details

There exist some implementation details which are not discussed as part of the overall algorithm formulation.

First, we do assume that a given webpage does have some tag structure. Without HTML tags we cannot calculate the Tag Ratio array and the method will fail. To cope with these instances we assume that tagless webpages contain only content and we return the entire text.

Second, there exist some webpages wherein the HTML markup is written in a single line. Without multiple lines the computed Tag Ratio array would only contain one element and CETR would be forced to either return all text or no text. Fortunately, we are able resolve this issue by detecting these instances and inserting line breaks every 65 characters. If the 65^{th} character is located within a tag, then the line break is inserted at the next non-tag location.

6. EXPERIMENTS

In this section we conduct experiments on real world data from various Internet corpora to demonstrate the effectiveness of CETR.

6.1 Data Set

In our experiments we use data from two sources: (1) news site data from Pasternak and Roth's 2009 WWW paper [28] on maximum subsequence segmentation (MSS) and (2) training and test data sets from the CleanEval competition.

MSS: In order to appropriately compare to the maximum subsequence segmentation method, which we were unable to obtain or implement, we retrieved identical data from Pasternak and Roth's repository[4]. This dataset contains labeled webpages where labels mark the beginning and end of the content section(s). Labels in this data set were gathered by examining a few pages per news source and a template-based wrapper was manually written. Even with this semi-automated approach, this was still a tedious process taking nearly 12 hours to complete. It is also noted that, in order to eliminate non-news article pages, any webpages which contained less than fifty words and symbols as well as any webpages which contained more than 20% tags were discarded. The authors made no attempt to manually check for nor correct errors in the 24,000 wrapper-produced samples.

This data set contained 45 individual websites which were further separated into two non-overlapping sets. (1) the *Big 5*: Tribune, Freep, NY Post, Suntimes, and Techweb, and (2) the *Myriad 40* which were chosen randomly from from the Yahoo! Directory. The *Myriad 40* contains "an international mix of English-language sites of widely varying size and sophistication" [28].

For our purposes we arbitrarily selected 50 documents from each of the *Big 5* and 206 documents total from the *Myriad 40*. Aside from these sources, we also selected 50 additional pages from the BBC and NY Times websites each because we felt that these two sources are highly popular and should be explicitly included in our evaluation.

CleanEval: The CleanEval project is a shared task for cleaning arbitrary webpages. This was started by the ACL's SIGWAC and initially took place as a competition during the Summer of 2007. This corpus includes four divisions: a development/training set and an evaluation set in both English and Chinese languages which are all hand-labeled. Besides extracting content, the original CleanEval competition also asked participants to "markup" the webpage. This task scored the participants on how well their algorithm identified lists, paragraphs and headers; we consider this addi-

[4] `http://l2r.cs.uiuc.edu/~cogcomp/Data/MSS/`

tional task outside the scope of our approach and therefore do not consider it further.

Because our approach does not require training there is no need to separate between training/development and evaluation documents. Therefore, we effectively have two CleanEval sets: (1) 741 English documents and (2) 713 Chinese documents.

6.2 Performance Metrics

For evaluation, standard metrics are used to evaluate and compare the performance of different methods. Specifically, precision, recall and F$_1$-scores are calculated by comparing the results/output of each method against a hand-labeled gold standard. Let W_P be the set of words in the extraction result and W_L be the set of words in the labeled extraction. Precision and recall then follow from:

$$P = \frac{|W_P \cap W_L|}{|W_P|}, R = \frac{|W_P \cap W_L|}{|W_L|} \qquad (5)$$

The F$_1$-scores are computed as usual and all results are calculated by averaging each of the metrics over all examples. We also present the scores from the *Big 5*, BBC and NY Times individually. It is important to note that every word in the document is considered to be distinct even if two words are lexically the same. One exception to this is the VIPS results, which often moves or removes text from its output; this makes it impossible to align words with the original page and therefore forces us to treat W_P and W_L as a bag of words, *i.e.*, where two words are considered the same if they are lexically the same. The bag of words measurement is more lenient and as a result VIPS scores may be further inflated.

The CleanEval competition uses a different approach when computing extraction performance. Their scoring method is based on the Levenshtein distance between the extraction algorithm and the gold standard divided by the alignment length. The Levenshtein distance is typically described as being the number of insertions and deletions of characters necessary to align two strings. The CleanEval version of the Levenshtein distance operates on the insertion and deletion of words rather than individual characters (presumably for either conceptual clarity or computation time). The alignment length is the number of insertion, deletion or align operations required to align two word sequences. The Levenshtein distance is relatively expensive to compute, taking $O(|A| \times |B|)$ time, which can be prohibitively large when $|A|$ and/or $|B|$ are sufficiently large. We find that our datasets typically include documents which are "sufficiently large" (*i.e.*, size greater than 10,000 words) and therefore we do not evaluate our performance using this metric.

6.3 Alternative Approaches

In order to properly evaluate the performance of CETR we compare it's performance with several other content extraction algorithms.

Several of the algorithms described in Section 2 have been implemented in Java (FE, KFE, BTE, DSC, ADSC, LQ, LP, CCB) for the CombineE framework [15]. None of these algorithms require training, so the evaluation is done by inputting each document one-by-one into each algorithm and gathering the results.

VIPS was evaluated similarly except for two major differences. First, VIPS was not implemented, rather the executable program was taken directly from the author's website. Second, the output from VIPS is a set of page segments rather than extracted text. As mentioned in Section 2, we exhaustively search for the perfect parameters for segmenting, and from the results, we exhaustively search for the best possible combination of page segments by comparing each combination with the gold standard and selecting the segment(s) with the best F$_1$-score. This certainly inflates the extraction performance over practical means.

MSS is neither implemented nor directly tested, instead the experiments described in this paper were deliberately designed to match those of [28]. In some instances, such as the Chinese language CleanEval, NY Times, and BBC, the MSS scores are missing because those datasets were not tested or not reported in the original work. We are confident that our results can be compared to MSS because we worked directly with the authors of the MSS experiments when preparing our experiments.

CETR is implemented in Java[5] and is divided into three distinct algorithms. The first is the one dimensional Threshold Method (CETR-TM) from Section 4.2. The second is the one dimensional method which is clustered with k-Means$_{k=3}$ (CETR-KM) from Section 4.3. The third iteration of this algorithm is the two dimensional method clustered with the tailored clustering technique (CETR) from Section 5.

6.4 Results

Table 1 presents the results of the Threshold Method (CETR-TM) when given the task of extracting content from the CleanEval, *Myriad 40*, *Big 5*, NY Times and BBC data sets. The *Big 5* is broken down into it's individual sources.

Table 1: Results for CETR-TM on various domains

Source	Precision	Recall	F$_1$-Measure
CleanEval-Eng	97.52%	90.92%	94.10%
CleanEval-Zh	89.03%	84.21%	86.55%
CleanEval	**93.27%**	**87.56%**	**90.33%**
Myriad 40	**87.86%**	**95.31%**	**91.44%**
NY Post	65.43%	100%	79.10%
Freep	63.93%	96.94%	77.05%
Suntimes	59.97%	100%	74.97%
Techweb	61.64%	100%	76.27%
Tribune	99.13%	98.74%	98.94%
Big 5	**70.02%**	**99.14%**	**81.23%**
NYTimes	100%	94.38%	97.11%
BBC	97.41%	99.12%	98.26%

With the CETR-TM method we observe a very high recall rate. This is because the threshold τ is set to 1.0σ, *i.e.*, $\lambda \leftarrow 1.0$. Intuitively, if λ is increased (*e.g.*, 1.1σ, 1.2σ) then the selectivity of the threshold would increase causing the precision to increase and the recall to decrease. Conversely, if λ is decreased (*e.g.*, 0.9σ, 0.8σ) then the selectivity of the threshold would decrease resulting in a lower precision and a higher recall. Tuning this parameter is left to the user, and Section 6.5 discusses λ in further detail.

Table 2 presents the results of the k-Means (CETR-KM) clustering method.

[5] The CETR implementation is available online at `http://www.cs.illinois.edu/homes/weninge1/`

Table 2: Results for CETR-KM on various domains

Source	Precision	Recall	F_1-Measure
CleanEval-Eng	96.85%	92.98%	94.88%
CleanEval-Zh	95.65%	78.95%	86.50%
CleanEval	**96.25%**	**85.96%**	**90.69%**
Myriad 40	**95.87%**	**92.54%**	**94.17%**
NY Post	76.64%	100%	86.78%
Freep	82.78%	92.44%	87.34%
Suntimes	96.28%	98.97%	97.61%
Techweb	78.21%	100%	87.78%
Tribune	100%	93.50%	96.64%
Big 5	**86.78%**	**96.98%**	**91.23%**
NYTimes	99.64%	97.18%	98.40%
BBC	100%	94.19%	97.01%

The CETR-KM method typically achieves either a high recall or a high precision but rarely both at the same time. Nevertheless, these results show that CETR-KM typically outperforms CETR-TM.

Table 3 presents the results of the complete CETR algorithm.

Table 3: Results for CETR on various domains

Source	Precision	Recall	F_1-Measure
CleanEval-Eng	96.66%	92.86%	94.72%
CleanEval-Zh	92.31%	81.72%	86.69%
CleanEval	**94.49%**	**87.29%**	**90.71%**
Myriad 40	**96.84%**	**92.68%**	**94.72%**
NY Post	83.57%	94.18%	88.56%
Freep	83.57%	92.44%	87.78%
Suntimes	99.86%	98.01%	98.93%
Techweb	76.34%	100%	86.59%
Tribune	99.61%	94.73%	97.11%
Big 5	**88.59%**	**95.87%**	**91.82%**
NYTimes	99.26%	97.18%	98.21%
BBC	100%	95.04%	97.46%

These results show that the complete CETR algorithm performs far better than CETR-TM and/or CETR-KM. There is some variability among the results, which have an F_1-Measure range of 98.93% for Suntimes to 86.59% for Techweb. Perhaps most importantly, the CleanEval scores were high relative to the highest score in the CleanEval competition, which scored an 84.1% on the English dataset. Remember, however, that the scoring metrics used in this paper, and in most similar literature, (precision, recall, F_1) are different from the scoring metrics used in the CleanEval competition (Levenshtein distance).

The relatively low precision reported by NY Post, Freep and Techweb is likely due to the fact that these sources contain user comments, feedback, etc. after each article. CETR typically does not include short comments as content whereas the gold standard extractions include these comments. Therefore, we contend that the actual precision of CETR is likely higher than the indicated precision. This *comment effect* becomes more evident when we see that more precise results are from sources such as NYTimes which hides comments by default, Suntimes which limits the comments to nine at a time, Tribune which limits the comments to three by default, and BBC which does not accept comments at all.

6.4.1 Methods Comparison

In order to judge the veracity of CETR we compare its performance with the alternative approaches described earlier in this section. Table 4 presents these results with the winners for each data source in bold. Some of the MSS results are not listed because the original work did not perform experiments on these data sources. The CETR threshold method is abbreviated CETR-TM and the 1-dimensional CETR clustered with k-Means is abbreviated CETR-KM.

CETR is the highest performing algorithm in most data sets and overall. The MSS algorithm performs highest on the *Big 5* data set. We believe this is because of the *comment effect* mentioned earlier, for instance, if the low performing precision results from Table 3 were more in line with the median precision of CETR then the average F_1-measure would outperform MSS and VIPS by an even greater margin.

6.5 Discussion

Tables 3 and 4 clearly show that CETR is a viable and robust content extraction algorithm by performing relatively well even on non-news corpora (CleanEval) and across multiple languages (English and Chinese). Admittedly, MSS does perform relatively well especially on the news corpora, however, we should emphasize that, unlike MSS, CETR is a completely unsupervised algorithm and therefore does not require labeled training examples. We must also view the VIPS results with some hesitation because, as stated earlier, VIPS was evaluated assuming the perfect parameters and segments were selected for each webpage.

The results also show that occasionally CETR-TM or CETR-KM does perform the best. We believe this to be because of nuances among website-page architecture. For instance, NY Times and BBC websites have structures that are most conducive to CETR-TM and CETR-KM. However, the results of broader corpora, *i.e.*, *Myriad 40* and CleanEval, show that CETR performs the best in the general case.

Even though CETR-TM does not perform the best overall, for practical purposes end users may consider its use when recall is a top priority. By reducing the threshold coefficient λ users can see a marked increase in the recall and a sharp decrease in the precision. This precision/recall tradeoff is shown in Figure 6. When $\lambda = 0$ the recall is always 100% because all lines are included. For the NY Times domain, shown in Figure 6, a good tradeoff might be $\lambda = 0.5$. Finding a good threshold value is difficult because λ must be empirically found for each domain.

Although CETR-TM and CETR-KM perform relatively well overall, we find that these methods are highly susceptible to webpages which do not have smooth tag ratio sections. Taking the American Declaration of Independence example from earlier, we find that content text which is presented in lists are sometimes missed by the CETR-TM and CETR-KM methods. This is because the threshold/clustering procedures regard the Tag Ratio array as a bag of values instead of an *ordered sequence* of values. As a result low-lying ratios can be missed even after smoothing.

The complete CETR algorithm solves this problem by explicitly identifying the content's borders by way of the absolute derivative array. With this new information the CETR algorithm is better able to identify the beginning and end of

Table 4: F_1-measures for each algorithm in each source. Winners are in bold.

Algorithm	CleanEval-Eng	CleanEval-Zh	**CleanEval**	***Myriad 40***	***Big 5***	NyTimes	BBC	Average
FE	7.86%	3.50%	5.68%	4.63%	8.27%	2.35%	17.14%	7.29%
KFE	89.19%	45.68%	67.44%	71.41%	71.36%	94.30%	78.13%	75.01%
BTE	93.13%	18.52%	55.83%	68.97%	64.58%	93.49%	63.93%	67.10%
DSC	80.92%	5.00%	42.96%	84.59%	81.54%	89.69%	80.96%	70.45%
ADSC	86.70%	5.13%	45.91%	86.41%	80.27%	96.06%	96.64%	75.20%
LQ	91.96%	58.47%	75.22%	70.25%	54.96%	93.42%	64.00%	72.18%
LP	49.65%	55.41%	52.53%	83.11%	25.89%	97.35%	90.48%	66.98%
CCB	91.57%	58.99%	75.28%	77.05%	68.21%	98.09%	71.90%	77.64%
MSS	91.98%	–	–	94.64%	95.13%	–	–	93.92%
VIPS	93.41%	39.43%	66.42%	92.97%	**95.59%**	95.61%	84.77%	83.63%
CETR-TM	94.10%	86.55%	90.33%	91.44%	81.27%	97.11%	**98.26%**	91.45%
CETR-KM	94.68%	86.50%	90.59%	94.17%	91.23%	**98.40%**	97.01%	93.66%
CETR	**94.72%**	**86.62%**	**90.67%**	**94.72%**	91.82%	98.21%	97.46%	**93.93%**

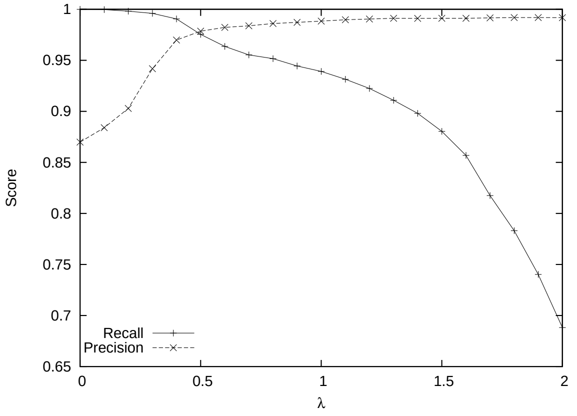


Figure 6: Precision and Recall tradeoff for NY Times corpora as the threshold coefficient (λ) is increased from 0 to 2

content sections. This information coupled with the original TR array create a novel model by which content sections can be identified.

Furthermore, there is no rule which states that a webpage may only have a single content section. There exists several instances in which content is divided by a menu or an advertisement which indicates the "fold" – referring to newspapers which are delivered folded in half. Unlike many current methods, VIPS especially, CETR is not affected by multiple content sections.

Despite the many advantages related to our algorithm, we do recognize some weaknesses. CETR does not perform well on portal home pages. For example, the *Yahoo!* homepage contains a vast array of menus and short news descriptions; CETR has difficulty discerning the content section(s) of these types of webpages. *Google News* is another webpage where content is difficult to discern; CETR typically extracts far more text than what users would consider content *i.e.*, recall is high and precision is low. Finally, we observe that webpages which do not have advertisements or menus, such as computer science professors' homepages, do not achieve high extraction accuracy. In these instances, CETR typically removes courses taught, patents awarded, and sometimes publications lists. The only way around this problem is to determine whether or not a given webpage contains non-content text, and then if it is determined that the webpage in question does contain non-content text invoke CETR to extract the content.

7. SUMMARY

The effectiveness of extracting content text from HTML documents using the Content Extraction via Tag Ratios (CETR) algorithm has been demonstrated. Furthermore, results show that when compared to several other leading content extraction methods CETR performs best on average.

Besides the demonstrated effectiveness of the algorithm, perhaps CETR's greatest strength over other methods is the simplicity of the concept, implementation and execution of the algorithm. The complete CETR algorithm contains no parameters to adjust ($k \leftarrow 3$ and $\alpha \leftarrow 3$ works for most cases), no training to be done, and no classifier models to build; all that is required is to give, as input, an HTML document and the approximate content will be returned.

Ultimately, CETR provides a fast, accurate method for extracting content from a variety of sources with little effort.

7.1 Future Work

The task of automatic content extraction remains a hot topic especially with the colossal amount of information being added to the Internet every day. With that in mind, there some portions of this specific approach that need further exploration.

We intend to incorporate this method into standard search engines in order to see what effect, if any, it has on the result relevance. For instance, many webpages include word strings and links in order to boost their search engine rank, if we can filter the irrelevant text from the page during indexing then it may be possible to present more relevant search results.

Another area for further investigation is the clustering algorithm used in CETR. We do not claim that our clustering method is optimal, in fact, a linear max-margin clustering approach may give better results.

8. ACKNOWLEDGMENTS

The authors sincerely thank Thomas Gottron, author of the CombineE framework and by extension some of the implementations used in this work. We also thank Jeff Pasternack, author of MSS, for the fruitful discussions and data access, and the CleanEval team for providing a standard evaluation data set. This work is funded by an NDSEG Fellowship award to the first author.

9. REFERENCES

[1] *19th International Workshop on Database and Expert Systems Applications (DEXA 2008), 1-5 September 2008, Turin, Italy.* IEEE Computer Society, 2008.

[2] B. Adelberg. Nodose - a tool for semi-automatically extracting semi-structured data from text documents. In *SIGMOD Conference*, pages 283–294. ACM Press, 1998.

[3] Z. Bar-Yossef and S. Rajagopalan. Template detection via data mining and its applications. In *WWW*, pages 580–591, 2002.

[4] O. Buyukkokten, H. Garcia-Molina, and A. Paepcke. Accordion summarization for end-game browsing on pdas and cellular phones. In *CHI*, pages 213–220, 2001.

[5] D. Cai, X. He, J.-R. Wen, and W.-Y. Ma. Block-level link analysis. In *SIGIR*, pages 440–447. ACM, 2004.

[6] D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma. Extracting content structure for web pages based on visual representation. In *APWeb*, volume 2642 of *Lecture Notes in Computer Science*, pages 406–417. Springer, 2003.

[7] R. Cathey, L. Ma, N. Goharian, and D. A. Grossman. Misuse detection for information retrieval systems. In *CIKM*, pages 183–190. ACM, 2003.

[8] J. Chen, B. Zhou, and H. Zhang. Function-based object model towards website adaptation. In *In Proceedings of the 10th International World Wide Web Conference*, pages 587–596. ACM Press, 2001.

[9] L. Chen, S. Ye, and X. Li. Template detection for large scale search engines. In *SAC*, pages 1094–1098. ACM, 2006.

[10] D. de Castro Reis, P. B. Golgher, A. S. da Silva, and A. H. F. Laender. Automatic web news extraction using tree edit distance. In *WWW*, pages 502–511. ACM, 2004.

[11] S. Debnath, P. Mitra, and C. L. Giles. Automatic extraction of informative blocks from webpages. In *SAC*, pages 1722–1726. ACM, 2005.

[12] S. Debnath, P. Mitra, and C. L. Giles. Identifying content blocks from web documents. In *ISMIS*, volume 3488 of *Lecture Notes in Computer Science*, pages 285–293. Springer, 2005.

[13] A. Finn, N. Kushmerick, and B. Smyth. Fact or fiction: Content classification for digital libraries. In *DELOS Workshop: Personalization and Recommender Systems in Digital Libraries*, 2001.

[14] T. Gottron. Evaluating content extraction on html documents. In *ITA*, pages 123–132, 2007.

[15] T. Gottron. Combining content extraction heuristics: the *ombine* system. In *iiWAS*, pages 591–595. ACM, 2008.

[16] T. Gottron. Content code blurring: A new approach to content extraction. In *DEXA Workshops* [1], pages 29–33.

[17] S. Gupta, G. E. Kaiser, P. Grimm, M. F. Chiang, and J. Starren. Automating content extraction of html documents. *World Wide Web*, 8(2):179–224, 2005.

[18] S. Gupta, G. E. Kaiser, D. Neistadt, and P. Grimm. Dom-based content extraction of html documents. In *WWW*, pages 207–214, 2003.

[19] S. Gupta, G. E. Kaiser, and S. J. Stolfo. Extracting context to improve accuracy for html content extraction. In *WWW (Special interest tracks and posters)*, pages 1114–1115. ACM, 2005.

[20] W. Han, D. Buttler, and C. Pu. Wrapping web data into xml. *SIGMOD Rec.*, 30(3):33–38, 2001.

[21] H.-Y. Kao, S.-H. Lin, J.-M. Ho, and M.-S. Chen. Mining web informative structures and contents based on entropy analysis. *IEEE Trans. Knowl. Data Eng.*, 16(1):41–55, 2004.

[22] N. Kushmerick. Wrapper induction: efficiency and expressiveness. *Artificial Intelligence*, 118(1-2):15–68, 2000.

[23] S.-H. Lin and J.-M. Ho. Discovering informative content blocks from web documents. In *KDD*, pages 588–593. ACM, 2002.

[24] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Berkeley Symposium on Mathematics Statistics and Probability*, pages 281–297, 1967.

[25] C. Mantratzis, M. A. Orgun, and S. Cassidy. Separating xhtml content from navigation clutter using dom-structure block analysis. In S. Reich and M. Tzagarakis, editors, *Hypertext*, pages 145–147. ACM, 2005.

[26] M. Marek, P. Pecina, and M. Spousta. Template detection through conditional random fields. In *WAC3*, 2007.

[27] I. Muslea, S. Minton, and C. A. Knoblock. Hierarchical wrapper induction for semistructured information sources. *Autonomous Agents and Multi-Agent Systems*, 4(1-2):93–114, 2001.

[28] J. Pasternack and D. Roth. Extracting article text from the web with maximum subsequence segmentation. In *WWW*, pages 971–980. ACM, 2009.

[29] D. Pinto, M. Branstein, R. Coleman, W. B. Croft, M. King, W. Li, and X. Wei. Quasm: a system for question answering using semi-structured data. In *JCDL*, pages 46–55. ACM, 2002.

[30] A. F. R. Rahman, H. Alam, and R. Hartono. Content extraction from html documents. In *WDA*, pages 7–10, 2001.

[31] T. V. Raman. Toward 2w, beyond web 2.0. *Commun. ACM*, 52(2):52–59, 2009.

[32] T. Weninger and W. H. Hsu. Text extraction from the web via text-to-tag ratio. In *DEXA Workshops* [1], pages 23–28.

[33] L. Yi, B. Liu, and X. Li. Eliminating noisy information in web pages for data mining. In *KDD*, pages 296–305. ACM, 2003.