Genre Classification of Websites Using Search Engine Snippets

Suhit Gupta
Columbia University
500 W. 120th Street
New York, NY 10027
001-212-939-7184
suhit@cs.columbia.edu

Gail Kaiser
Columbia University
500 W. 120th Street
New York, NY 10027
001-212-939-7081
kaiser@cs.columbia.edu

Salvatore Stolfo
Columbia University
500 W. 120th Street
New York, NY 10027
001-212-939-7080
sal@cs.columbia.edu

Hila Becker
Columbia University
500 W. 120th Street
New York, NY 10027
001-212-939-7100
hb2143@cs.columbia.edu

ABSTRACT

Web pages often contain clutter (such as ads, unnecessary images and extraneous links) around the body of an article, which distracts a user from actual content. Automatic extraction of "useful and relevant" content from web pages has many applications, including browsing on small cell phone and PDA screens, speech rendering for the visually impaired, and reducing noise for information retrieval systems. Prior work has led to the development of Crunch, a framework which employs various heuristics in the form of filters and filter settings for content extraction. Crunch allows users to tune these settings, essentially the thresholds for applying each filter. However, in order to reduce human involvement in selecting these heuristic settings, we have extended this work to utilize a website's classification, defined by its genre and physical layout. In particular, Crunch would then obtain the settings for a previously unknown website by automatically classifying it as sufficiently similar to a cluster of known websites with previously adjusted settings - which in practice produces better content extraction results than a single one-size-fits-all set of setting defaults. In this paper, we present our approach to clustering a large corpus of websites by their genre, utilizing the snippets generated by sending the website's domain name to search engines as well as the website's own text. We find that exploiting these snippets not only increased the frequency of function words that directly assist in detecting the genre of a website, but also allow for easier clustering of websites. We use existing techniques like Manhattan distance measure and Hierarchical clustering, with some modifications, to pre-classify websites into genres. Our clustering method does not require prior knowledge of the set of genres that websites fit into, but instead discovers these relationships among websites. Subsequently, we are able to classify newly encountered websites in linear-time, and then apply the corresponding filter settings, with no noticeable delay introduced for the content-extracting web proxy.

Categories and Subject Descriptors

I.7.4 [**Document and Text Processing**]: Electronic Publishing; H.3.5 [**Information Storage and Retrieval**]: Online Information Services – *Web-based Services*

General Terms

Human Factors, Algorithms, Standardization.

Keywords

Website classification, clustering, content extraction, reformatting, HTML, context, accessibility, speech rendering.

1. INTRODUCTION

Web pages are cluttered with guides and menus attempting to improve the user's efficiency, but they often end up distracting from the actual content. These "features" may include script- and flash-driven animation, menus, pop-up ads, obtrusive banner advertisements, unnecessary images, or links scattered around the screen. The automatic extraction of useful and relevant content from web pages has many applications, including enabling end users to access the web more easily over constrained devices like PDAs and cellular phones, providing better access to the web for the visually impaired, providing less noisy data for information retrieval and summarization algorithms, and generally improving the web surfing experience.

Gupta et al. [4] [5] [6] have developed a framework, Crunch, which employs various heuristics in the form of filters and filter settings for content extraction. In order to analyze a web page for content extraction, they pass it through an HTML parser, which corrects the markup and creates a Document Object Model tree. The Document Object Model (www.w3.org/DOM) is a standard for creating and manipulating in-memory representations of HTML (and XML) content. The Crunch system applies the relevant filters and their appropriate settings to this DOM tree and the resulting HTML page provided to the user is a clean, clutter-free page.

Crunch allows users to tune the settings, essentially the thresholds for applying each filter. However, while the proxy works very well on a large variety of webpages, the filters sometimes need to be manually configured by the user in order to properly extract content for a given site. We found that the settings for the proxy needed to be adjusted only when the user moved from one class of site to another. In order to reduce human involvement in

SIGIR August 15-19, 2005 Salvador, Bahia, Brazil.

selecting the heuristic settings for the appropriate content extraction, we considered utilizing a website's classification. Crunch could then obtain the settings for a previously unknown website by automatically classifying it as sufficiently similar to a cluster of known websites with previously adjusted settings - which in practice produces better content extraction results than a single one-size-fits-all set of setting defaults. To come up with the classification for any site, we decided to use its genre as one of the two criteria; the other was its physical layout. We noted that filter settings did not change when browsing among sites that belonged to the same genres. For example, the link density, table/cell structure and advertisement layout remained consistent among sites with news genre vs. shopping genre. Therefore, settings that worked well for the various genres could be dynamically loaded once the genre for a site was identified.

With a good document clustering method, computers can automatically organize a document corpus into a meaningful cluster hierarchy, which enables an efficient browsing and navigation of the corpus. [17] However, performing genre analysis in real-time is too computationally expensive for web applications. Additionally, most current clustering algorithms require a priori knowledge of the number of clusters to properly classify a set of documents; however, given the vast quantity and enormous variety of documents on the web, this does not seem like a suitable approach. So, the tractability of classifying a document at runtime into an indeterminate number of categories remains vague at best. We found it necessary to identify a basic set of clusters so that new websites encountered could simply be compared to sites in those pre-defined sets.

Our goal was to use a simple algorithm for the pre-clustering of a large corpus of web documents. Our key insight was utilizing the results (defined as "snippets") generated by sending the website's domain name to search engines as well as the website's own text towards determining the genre of websites. We find that exploiting these snippets not only increased the frequency of function words that directly assist in detecting the genre of a website, but also allow for easier clustering of websites. We found snippets to be descriptive of the function of the websites being accessed and especially useful since they added relevant knowledge in the form of function words which could then be used in the analysis of the appropriate genre. We use existing techniques like Manhattan distance measure and Hierarchical clustering, with some modifications, to pre-classify websites into genres. Our clustering method does not require prior knowledge of the set of genres that websites fit into, but instead discovers these relationships among websites. Subsequently, we are able to classify newly encountered websites in linear-time, and then apply the corresponding filter settings, with no noticeable delay introduced by our content-extracting web proxy. In this paper, we explain our method for clustering websites and briefly describe the ensuing classification of individual new websites in order to get good content extraction results.

The following sections describe the related work in the field of information retrieval and document clustering, followed by our approach and the associated implementation details, and we will conclude with results from our experiments and a summary of our contributions.

2. RELATED WORK

Document clustering has been extensively investigated over the years for many domains [3] [16].The use of clustering in IR appears mostly to be driven by the cluster hypothesis, which states that "closely associated documents tend to be related to the same requests". [1] Xu et al. point out that document clustering methods can be mainly categorized into two types: document partitioning (flat clustering) and agglomerative (bottom-up hierarchical) clustering. Although both types of methods have been extensively investigated for several decades, accurately clustering documents without domain-dependent background information is still a challenging task. [17] They provide a novel document partitioning method using non-negative factorization of the term-document matrix. However, we have found that web document clustering can be done with little or no background information available about the domains, so long as there are enough function words available to help classify the documents.

There is a large body of related work in information retrieval and genre classification. Lee et al. [7] describe their method for text genre classification by using two different class sets, genre classes and subject classes, in the training data. However, their method would not work well for web documents since the number of genres need to be identified and fixed before the categorization. This, as pointed out before, is an impediment for the vast number and incredible variety of the corpus of web documents. For the same reason, traditional partitioning methods like K-Means clustering [9] will not work. Similarly, clustering using sequential information maximization, presented by Slonim et al. [14] requires prior knowledge of the number of clusters for the data to be classified.

Spectral clustering, an approach demonstrated by Ng. et al. [10], clusters points using eigenvectors of matrices derived from the data contained in the documents. However, like with K-Mean clustering, the number of clusters is expected to be a known quantity before the process. Additionally, spectral clustering works hard at performing tight fitting of all data points within a cluster, and later experiments clearly show that this is not a hard requirement when classifying web documents by genre.

Cutting et al. [3] describe a cluster based approach to browsing large document collections, called Scatter/Gather. It provides a mechanism for user-driven organization of data in a fixed number of clusters, but the users need to be in the loop and the computed clusters do not guarantee accuracy. Moreover, the technique is designed to work well for finding similarity between articles based on subject. We are interested in identifying broad topic genres, like news, shopping, and sports, for top-level domains.

Zhang et al. [21] present BIRCH, a clustering method for extremely large databases. The main optimizations in this technique are made to reduce the number of I/O transactions and to increase space efficiency. In order to truly gain the benefits from this model, multiple passes on the data are required. However, this increases the runtime complexity of the system. Additionally, we observed that a relatively small number of documents (as few as two, as demonstrated later in this paper) were enough to define a cluster and its associated genre.

Siersdorfer et al. [13] show an interesting approach to clustering by working on only a subset of the available data. The key element to their approach is to construct restrictive meta-methods at the moderate cost loss of uncertain samples. However, their approach, called restrictive clustering, which clusters with high accuracy, is prone to miss recognizing failed clustering attempts.

Zamir et al. [20] describe Grouper, a clustering system that employs term based clustering, an approach that is similar to ours. They use Suffix Tree Clustering, which is shown to be fast and domain independent. However, they have applied their methods to only grouping the results returned by search engine by analyzing the snippets produced by the appropriate search. We would like to actually like to automatically classify whole websites.

Lastly, Crammer et al. [2] describe a technique for performing online classification that focuses on online additive algorithms for classification tasks. Their approach was designed to save a user from needing vast computational resources. We tend not to suffer from this problem either since we define a fixed range of words (described in Sections 3 and 4) to classify the various websites. Additionally, the system described needs a training period to produce the level of classification achieved while we use a static set of data and are able to achieve similar results.

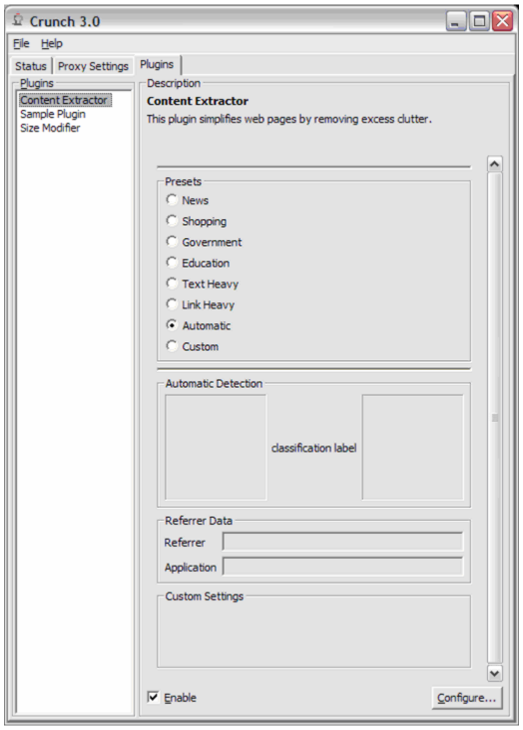
Finally, most of the work described here tends to be used towards analysis and clustering of documents or search engine results, but not for web pages. One general problem with using the strategies mentioned here is that most web pages contain extremely noisy data that may lead to incorrect clustering results and our results, which had to be used for the purpose of content extraction, had to be resilient to such noise.

3. APPROACH

Clustering involves the grouping of similar objects, and has been practiced, consciously or unconsciously, for many thousands of years. [16] Distance coefficients, such as Euclidean distance, have been used very extensively in cluster analysis, owing to their simple geometric interpretation. Our solution employs multiple techniques that incorporate the advantages of the previous work on clustering and information retrieval. However, in order to motivate the clustering, let us briefly describe the application.

3.1 Crunch

Extraction of “useful and relevant” content from web pages has many applications, including cell phone and PDA browsing, speech rendering for the visually impaired and reducing noise for information retrieval systems. Gupta et al. [4] [5] [6] suggested the key insight to work with DOM trees, rather than raw HTML markup. Access to various tunable heuristics, in the form of filters, are provided to the user through Crunch, their content extraction web proxy. Crunch extracts content from webpages based on filter settings that are typically applied by the user, based on the site they are browsing and the content they wish to see. Figure 2 shows an example of a typical CNN page without and with filtering.



Content Extractor

This plugin simplifies web pages by removing excess clutter.

Presets: News, Shopping, Government, Education, Text Heavy, Link Heavy, [✓ Automatic], Custom

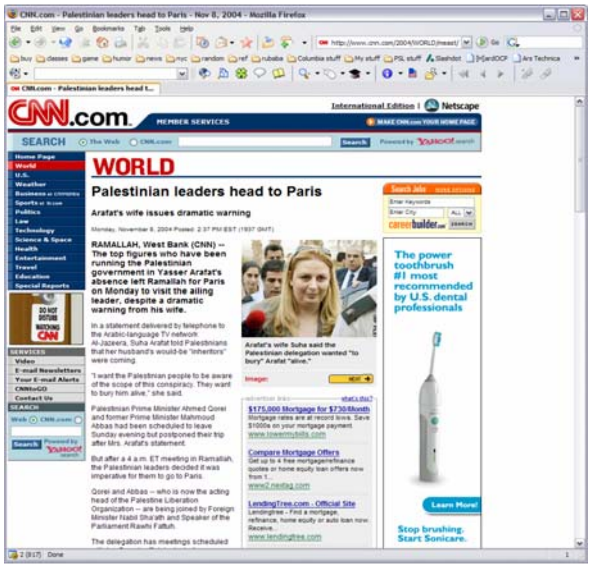
Referrer:

Application:

[✓ Enable]

Figure 1 - Crunch Control Panel

However, the Crunch settings (shown in Figure 1) require manual tweaking for various sites. Our goal was to automate the application of filter settings for a varied range of websites by detecting the classification of websites – defined by content genres and the physical layout. We noticed that, while certain settings worked well for particular genres of sites, for example most news sites shared the same filter settings that produce the best extraction of content; those same settings did not work well with other genres like shopping, sports or astronomy. This was mainly due to two reasons – (i) sites from the same genre generally followed similar layouts; and (ii) users defined content similarly across various genres. Therefore, we developed the hypothesis that detecting the genre for a site would enable better content extraction.


CNN.com - Palestinian leaders head to Paris - Nov 8, 2004 - Mozilla Firefox

WORLD

Palestinian leaders head to Paris

Arafat's wife issues dramatic warning

Monday, November 8, 2004 Posted: 2:37 PM EST (1937 GMT)

RAMALLAH, West Bank (CNN) -- The top figures who have been running the Palestinian government in Yasser Arafat's absence left Ramallah for Paris on Monday to visit the ailing leader, despite a dramatic warning from his wife.

In a statement delivered by telephone to the Arabic-language TV network Al-Jazeera, Suha Arafat told Palestinians that her husband's would-be "inheritors" were coming.

"I want the Palestinian people to be aware of the scope of this conspiracy. They want to bury him alive," she said.

Palestinian Prime Minister Ahmed Qorei and former Prime Minister Mahmoud Abbas had been scheduled to leave Sunday evening but postponed their trip after Mrs. Arafat's statement.

But after a 4 a.m. ET meeting in Ramallah, the Palestinian leaders decided it was imperative for them to go to Paris.

Qorei and Abbas -- who is now the acting head of the Palestine Liberation Organization -- are being joined by Foreign Minister Nabil Sha'ath and Speaker of the Parliament Rawhi Fattuh.

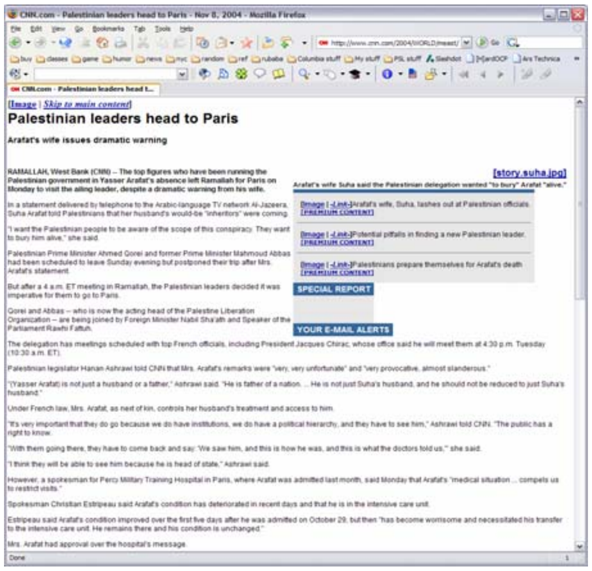
The delegation has meetings scheduled

Arafat's wife Suha said the Palestinian delegation wanted "to bury" Arafat "alive."

$175,000 Mortgage for $730/Month
Mortgage rates are at record lows. Save $1000s on your mortgage payment.
www.lowermybills.com

Compare Mortgage Offers
Get up to 4 free mortgage/refinance quotes or home equity loan offers now from 1...
www2.nextag.com

LendingTree.com - Official Site
Lendingtree - Find a mortgage, refinance, home equity or auto loan now. Receive...
www.lendingtree.com


CNN.com - Palestinian leaders head to Paris - Nov 8, 2004 - Mozilla Firefox

[Image | *Skip to main content*]

Palestinian leaders head to Paris

Arafat's wife issues dramatic warning

[story.suha.jpg]

Arafat's wife Suha said the Palestinian delegation wanted "to bury" Arafat "alive."

RAMALLAH, West Bank (CNN) -- The top figures who have been running the Palestinian government in Yasser Arafat's absence left Ramallah for Paris on Monday to visit the ailing leader, despite a dramatic warning from his wife.

In a statement delivered by telephone to the Arabic-language TV network Al-Jazeera, Suha Arafat told Palestinians that her husband's would-be "inheritors" were coming.

[Image | -Link-]Arafat's wife, Suha, lashes out at Palestinian officials [PREMIUM CONTENT]

"I want the Palestinian people to be aware of the scope of this conspiracy. They want to bury him alive," she said.

[Image | -Link-]Potential pitfalls in finding a new Palestinian leader. [PREMIUM CONTENT]

Palestinian Prime Minister Ahmed Qorei and former Prime Minister Mahmoud Abbas had been scheduled to leave Sunday evening but postponed their trip after Mrs. Arafat's statement.

[Image | -Link-]Palestinians prepare themselves for Arafat's death [PREMIUM CONTENT]

But after a 4 a.m. ET meeting in Ramallah, the Palestinian leaders decided it was imperative for them to go to Paris.

SPECIAL REPORT

Qorei and Abbas -- who is now the acting head of the Palestine Liberation Organization -- are being joined by Foreign Minister Nabil Sha'ath and Speaker of the Parliament Rawhi Fattuh.

YOUR E-MAIL ALERTS

The delegation has meetings scheduled with top French officials, including President Jacques Chirac, whose office said he will meet them at 4:30 p.m. Tuesday (10:30 a.m. ET).

Palestinian legislator Hanan Ashrawi told CNN that Mrs. Arafat's remarks were "very, very unfortunate" and "very provocative, almost slanderous."

"(Yasser Arafat) is not just a husband or a father," Ashrawi said. "He is father of a nation. ... He is not just Suha's husband, and he should not be reduced to just Suha's husband."

Under French law, Mrs. Arafat, as next of kin, controls her husband's treatment and access to him.

"It's very important that they do go because we do have institutions, we do have a political hierarchy, and they have to see him," Ashrawi told CNN. "The public has a right to know.

"With them going there, they have to come back and say: 'We saw him, and this is how he was, and this is what the doctors told us,'" she said.

"I think they will be able to see him because he is head of state," Ashrawi said.

However, a spokesman for Percy Military Training Hospital in Paris, where Arafat was admitted last month, said Monday that Arafat's "medical situation ... compels us to restrict visits."

Spokesman Christian Estripeau said Arafat's condition has deteriorated in recent days and that he is in the intensive care unit.

Estripeau said Arafat's condition improved over the first five days after he was admitted on October 29, but then "has become worrisome and necessitated his transfer to the intensive care unit. He remains there and his condition is unchanged."

Mrs. Arafat had approval over the hospital's message.

Figure 2 - Content Extraction results on a typical cnn.com article (original page vs. extracted page)

Since this is a web application, detection of the content genre needs to be done in near real-time, for the overall content extraction to be done in a reasonable amount of time. However, as pointed out before, performing genre-analysis on individual documents can be extremely computationally expensive. But, if the content extraction proxy already had data on the existence of various genre clusters (and the various heuristic filter settings that work well for those genres), then matching individual websites to those clusters and applying the appropriate settings could be an efficient process. Therefore, we focused on the offline identification of clusters of genres from a large corpus of websites.

3.2 Clustering

We found very few classification algorithms applied to the clustering of web documents (as shown in Section 2). These few algorithms suffered from two basic problems – (i) they had high run-time complexity (cubic or higher running time), and/or (ii) they classified webpages into a fixed number of pre-determined clusters rather than finding the affinity among websites and discovering where the clusters lay. Our goal was to overcome these two basic constraints.

We use an external module as a preprocessor to classify almost 200 unique and frequently visited (by the members of our group) websites, covering genres like news (international as well as regional), shopping, astronomy and technical weblogs. Our key insight was to use results returned by search engines (defined as "snippets"), obtained only from the first page of the results returned by the various search engines, when searching for the domain name of the site in question, towards determining the genre of websites. We found that snippets contain words that are highly descriptive of the function of the websites being accessed. Snippets are especially useful since they add relevant knowledge in the form of function words which could then be used in the analysis of appropriate genres. Furthermore, we found that adding snippet data does not increase the complexity of the clustering algorithm, instead simply adds the access time to the particular search engine to the overall running time of the system.

For each site slated for clustering, we create a word frequency map consisting of the textual data of the webpage as well as the snippets produced by six popular search engines (Google, Yahoo, Dogpile, MSN, Altavista and Excite). For instance, searching for cnn or nypost, when accessing these sites respectively, increased the frequency of words such as "news" and "business", while searching for Amazon resulted in a frequency increase of the word "shop" and "books". From the frequency map of each document, we prune all words deemed insignificant – a stop word list – containing words that typically are parts of speech (prepositions, articles, pronouns, etc.), and other words that may appear frequently in a document but that do not add information to the genre of the site. We also remove all non-dictionary words; our dictionary contains 23,000 words and their variations, including some of common prefixes, suffixes and tenses. From the frequency maps, frequent (greater than 10) and unique words are added to a *Word Key* vector, if they were not already added by a previous site. We found that each site added, on an average, six new words to the *Word Key* list.

The frequency maps are then re-graphed across this *Word Key*. The re-graph against this new set of keywords produces an accurate content genre *identifier* for each of the websites (examples of graphs for the sites are shown in Figures 3 and 4 respectively). The next step, in order to perform clustering, was to find the distance of each *identifier* from all the other identifiers in our corpus. It has been pointed out that a major limitation of the Euclidean distance in the information retrieval context is that it can lead to two documents being regarded as highly similar to each other, despite the fact that they share no terms at all in common. [16] However, we found that due to the addition of snippet data, and consequently the increase in the frequency of genre-specific function words, we did suffer from this pitfall.

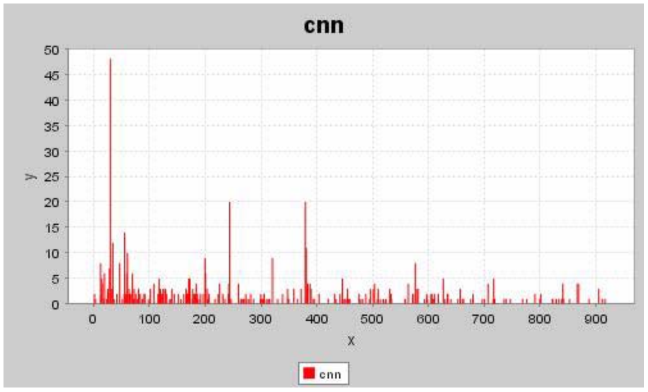


Figure 3 - Graph of CNN.com across the *Word Key*

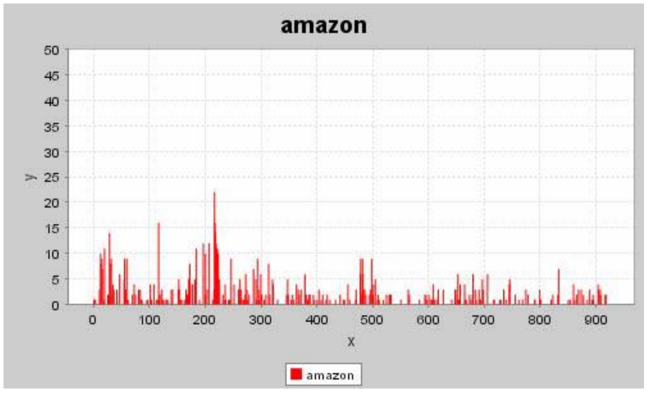


Figure 4 - Graph of Amazon.com across the *Word Key*

With all the distances in place, we sort the sites by this distance to create a list of sites that range from closest association to further. Here we employ the hierarchical clustering algorithm [16], with a slight variation (as described in the next section) to perform clustering. The clusters created are then manually tagged by the appropriate genre and the best heuristic settings, that produce the best content extraction results for that genre, are stored.

At run-time, when a new unclassified site is encountered, it is put through the same process as described above –the text from the site and the associated snippet data is aggregated, cleaned and graphed across the *Word Key*. Then using the same distance measuring algorithm, we find the site that is a closest match to the current webpage and its cluster is identified. The associated genre settings are loaded in Crunch and content for the site is extracted.

4. IMPLEMENTATION

We then use the *Manhattan histogram distance measure algorithm* to measure the distance between the website in question and our original classifications. The formula is defined as

$$D_1(h_1, h_2) = \sum_{i=0}^{n-1} | h_1[i] - h_2[i] |$$

The histogram (h_1 , h_2) is represented as a vector, where n is the number of bins in the histogram (i.e., the number of words in our Key Word Set). h_1 and h_2 must first be normalized in order to satisfy the above distance function requirements. In Crunch, the sum of the histogram's bins is normalized before computing the distance. We use the settings associated with the website whose distance is closest to the one being accessed. Calculating the Manhattan distance for every site from every other site is an O(n^2) operation.

Once the distance between any two sites is determined, we sort all distances in time O(n log n). We then proceed to find clusters based on the assumption that similar sites contain similar words. We do not force a specific number of clusters, but rather let clusters form naturally using our clustering algorithm. The clustering method that we use is a slight variation on the hierarchical clustering algorithm that runs in time O(n^2). [16] Each cluster is viewed as a tree with the closest pair of sites as its root. Starting with the closest pair of sites, our algorithm selects the next closest available pair of sites at each iteration from the set of sorted site distances. If neither site belongs to an existing cluster, they become the root of a new cluster. However, one of our customizations to the algorithm is that the root sites may pull in additional sites into the cluster if the algorithm encounters a pair of sites where one is the root and the other is not already clustered. A site that was pulled in directly by the root may pull in additional sites. However, any site that was not clustered directly by the root of that cluster may not pull in any other site, even if the algorithms encounters a site not yet been clustered. This restriction is imposed in order to prevent chains of more than three sites from forming in the same cluster, which ultimately prevents all the clusters from merging into one gigantic one. We found that sites associated with the root via more than one link are often far enough to be potentially closer to a different cluster. In many cases, sites that were initially rejected were still pulled into the cluster by a site that is directly connected to the root.

If the algorithm encounters a pair of sites which belong to different clusters, it simply proceeds to the next iteration since each of these sites was found to be closer to one classification than the other. The algorithm halts when the distance between the next pair of sites exceeds a preset threshold or when all possible pairs of sites have been examined. The threshold is used in order to prevent extremely unrelated sites from contaminating existing clusters. These sites are either manually inserted into an existing genre cluster or form their own cluster.

The overall runtime complexity of the offline clustering system is O(n^2) due to the quadratic level running time of both the distance measurements as well as the clustering. However, at runtime, when a new site is encountered and its genre needs to be determined, the complexity drops to O(n). This is so because once the textual data from the site and the snippet data are collected, the time to create the frequency graph is a constant time process and it needs to be distance matched to the n different pre-clustered sites to find the closest cluster that it belong to. Once this has been determined, the associated settings are loaded into Crunch.

5. EXPERIMENT & RESULTS

We chose 171 unique sites to test our system. Most of the sites chosen were the top sites visited by our group on a daily basis, with a few more unusual sites chosen for comparison purposes. In Table 1, we show results of our primary hypothesis – that using search engine snippets towards clustering produces better and tighter results. In Table 2, we actually show the top six clusters produced by the clustering system. The search engines used to extract snippets are Google, Yahoo, Dogpile, MSN, Altavista and Excite.

	Using Snippets	Without Snippets
No. of Sites	171	171
No. Search Engines	6	0
No. Clusters Found	14	5
Max. Cluster Size	71	159
Min. Cluster Size	2	2
Avg. Cluster Size	12.21	34.2
Time to cluster (min)	25	11

Table 1 - Cluster Results

From Table 1, we see that the number of clusters using snippets far exceeds the run of the system without using snippets. While the runtime of the system more than doubled, we found that the increase was only due to the increase in access time and data gathering from six more sites per website being clustered. Upon manual inspection, we found that the sites clustered by the first experiment, i.e. the one using snippets, categorized sites extremely accurately. Evidence of this comes from the fact that upon manual clustering websites into various genres, we came up with 16 distinct clusters and using snippets identified 87.5% of those clusters while only 31% of clusters were identified when snippets were not used.

	Cluster Type	# of Sites (w/snippets)	# of Sites (no snippets)	# of Sites (manual)
Cluster #1	International News	71	159	65
Cluster #2	Shopping	27	-	25
Cluster #3	Regional News	10	-	13
Cluster #4	Tech News	7	-	10
Cluster #5	Tech Blogs	7	-	8
Cluster #6	Astronomy	7	6	7

Table 2 – Typical Cluster Examples

Table 2 shows some of the top clusters that are created by our approach as well as the comparable approach without using snippets as part of the clustering data. Additionally, we manually inspected all the websites and assigned them to clusters based on personal analysis. International news sites were the most common sites and those made their way into one cluster. Other notable genres are shopping, regional news (all the sites related to news from India), and astronomy sites. An interesting observation was the separation of technical news sites vs. technical blogs. They clustered into to separate genres mainly because blogs tend to be updated more often than general tech news sites. Overall, it is worth pointing out that the approach without using snippets produces results that are far worse than the approach with. Moreover, the approach with snippets is far more in tune with what is observed upon human-based inspection.

Figures 5-6 show the graphs and their general similarity of various websites that were determined as being part of a cluster. Figure 5 shows international news websites (the news genre) like CNN.com, drudgereport.com, washingtontimes.com and chicagotribune.com. Here, Crunch's heuristic settings would be far more text specific and would aggressively remove links and advertisements. Figure 6 shows popular shopping sites (the shopping genre) like Amazon.com, ebay.com, pricewatch.com and streetprices.com. Here, the settings would be sensitive to associated link-heavy sites with large amounts of form data and advertisements.

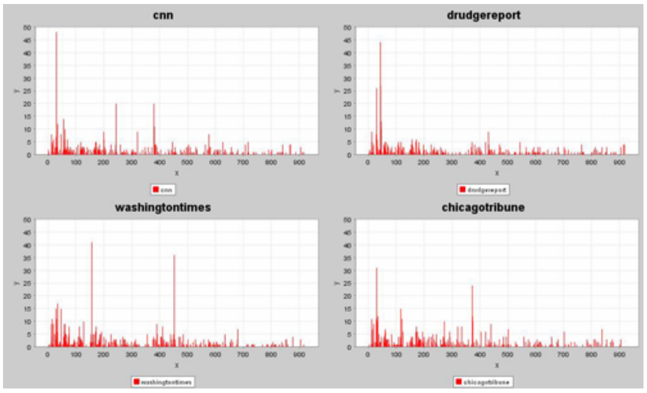


Figure 5 - Cluster containing CNN and other news sites

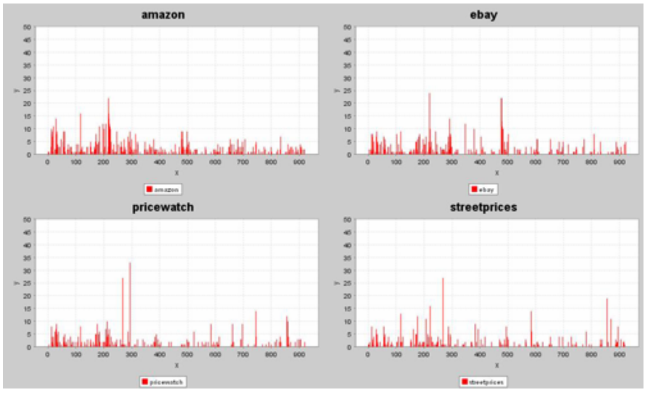


Figure 6 - Cluster containing Amazon and other shopping sites

While our system for classification is able to cluster similar sites, it is also able to clearly distinguish between singular sites that do not fit into any cluster, both in the pre-classification stage as well as the subsequent genre matching phase. Skinheadz.com is an example of such a site, and its frequency graph along the *Word Key* is shown in Figure 7. Similarly, sec.gov, whose frequency graph is shown in Figure 8, is a site that remained unmatched to the identified genres. (These two sites presumably fit into certain genres, but not those visited regularly by our group so their clusters do not appear in our base data.)

Our algorithm was also successfully able cluster sites whose frequency graphs look visually different but whose genre is similar. Examples of this are shown in Figures 9 and 10 in the Appendix.

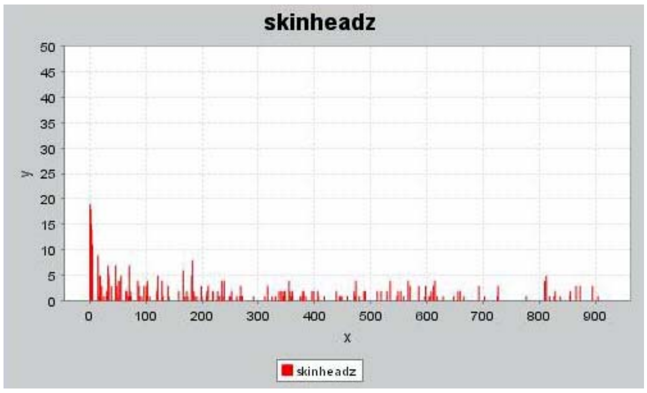


Figure 7 - Graph of skinheadz.com across the *Word Key*

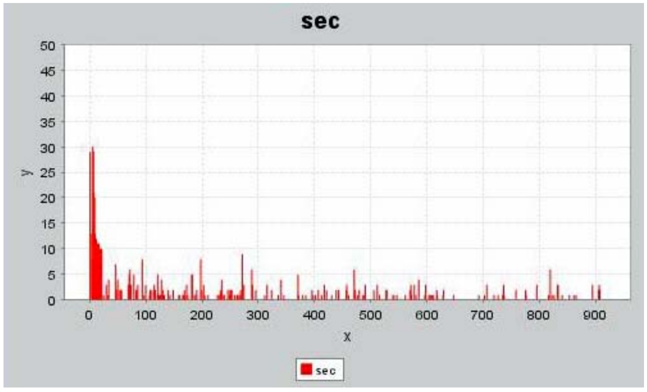


Figure 8 - Graph of sec.gov across the *Word Key*

We have also found that using snippets also makes use resilient to changes in the structure of clusters over time. Even though the content on the various websites may change, the snippets tend to remain the same. Therefore, the frequency of the most frequent words remains the same over time causing little fluctuations.

6. SUMMARY OF CONTRIBUTIONS

In this paper, we consider the problem of clustering websites based on their genre. However, web page classification is much more difficult than pure-text classification due to a variety of noisy information embedded in Web pages - which reminds of our original motivation for content extraction. [12] Utilizing snippets produced by search engine searches for the domain name of the website being classified, we are able to improve the frequency of the function words being considered to help classify that site. With these snippets as well as the textual content on the site itself, we use existing simple and proven techniques like Manhattan distance and a Hierarchical clustering, albeit with a slight variation, in order to successfully pre-cluster a large number of websites, in an efficient manner. This pre-clustering then allows us to classify individual new sites not already classified, in linear time, by comparing them to the existing clusters. We extended a content extraction proxy, Crunch [4] [5] [6], to use this information to produce better results in its content extraction goals. Further, our approach to identifying text genres should be beneficial to many text-based applications. For instance, if the genre of every document is known *a priori*, information retrieval results could be better presented to the user, depending on the user's preferences. [7]

7. ACKNOWLEDGEMENTS

The Programming Systems Laboratory is funded in part by National Science Foundation grants CNS-0426623, CCR-0203876 and EIA-0202063, and in part by Microsoft Research. Part of the work reported in this paper emanated from research from the Columbia IDS lab, which has been supported by grants form NSF and HS ARPA.

8. REFERENCES

[1] Javed Aslam, Ekaterina Pelehov, Daniela Rus, "A Star Clustering Algorithm for Static and Dynamic Information Organization", Journal of Graph Algorithms and Applications, vol. 8, no. 1, 2004

[2] Koby Crammer, Jaz Kandola, Yoram Singer, "Online Classification on a Budget", Seventeenth Annual Conference on Neural Information Processing Systems, 2003

[3] Douglass Cutting, David Karger, Jan Pedersen, John Tukey, "Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collection", Proceedings of SIGIR '92, 15th ACM International Conference on Research and Development in Information Retrieval, 1992

[4] Suhit Gupta, Gail Kaiser, David Neistadt, Peter Grimm, "DOM-based Content Extraction of HTML Documents", 12th International World Wide Web Conference, 2003

[5] Suhit Gupta, Gail E Kaiser, Peter Grimm, Michael F Chiang, Justin Starren, "Automating Content Extraction of HTML Documents", Columbia Univ. Dept. of Comp. Sci. TR CUCS-001-04, 2004

[6] Suhit Gupta, Gail Kaiser, "CRUNCH - Web-based Collaboration for Persons with Disabilities", W3C Web Accessibility Initiative, Teleconference on Making Collaboration Technologies Accessible for Persons with Disabilities, 2003

[7] Yong-Bae Lee, Sung Hyon Myaeng, "Text Genre Classification with Genre-Revealing and Subject-Revealing Features", Proceedings of SIGIR '02, 25th ACM

International Conference on Research and Development in Information Retrieval, 2002

[8] Tao Li, Sheng Ma, Mitsunori Ogihara, "Document Clustering via Adaptive Subspace Iteration", Proceedings of SIGIR '04, 27th ACM International Conference on Research and Development in Information Retrieval, 2004

[9] Tom Mitchell, "Machine Learning", McGraw-Hill Science/Engineering/Math, March, 1997

[10] Andrew Ng, Michael Jordan, and Yair Weiss, "On spectral clustering: Analysis and an algorithm", In Advances in Neural Information Processing Systems, 2001

[11] Fabrizio Sebastiani, "Text categorization", In Alessandro Zanasi (ed.), Text Mining and its Applications, WIT Press, Southampton, UK, 2005

[12] Dou Shen, Zheng Chen, Qiang Yang, Hua-Jun Zeng, Benyu Zhang, Yuchang Lu, Wei-Ying Ma, "Web-page Classification through Summarization", Proceedings of SIGIR '04, 27th ACM International Conference on Research and Development in Information Retrieval, 2004

[13] Stefan Siersdorfer, Sergej Sizov, "Restrictive Clustering and Metaclustering for Self-Organizing Document Collections", Proceedings of SIGIR '04, 27th ACM International Conference on Research and Development in Information Retrieval, 2004

[14] Noam Slonim, Nir Friedman, Naftali Tishby, "Unsupervised Document Classification using Sequential Information Maximization", Proceedings of SIGIR '02, 25th ACM International Conference on Research and Development in Information Retrieval, 2002

[15] C.J. van Rijsbergen, "Information Retrieval", Butterworths, London, 2nd ed., 1979

[16] Peter Willett, "Recent Trends in Hierarchic Document Clustering: A Critical Review", Journal Information Processing and Management, 1988

[17] Wei Xu, Xin Liu, Yihong Gong, " Document Clustering Based on Non-negative Matrix Factorization", Proceedings of SIGIR '03, 26th ACM International Conference on Research and Development in Information Retrieval, 2003

[18] Yiming Yang, Jian Zhang, Bryan Kisiel, "A scalability analysis of classifiers in text categorization", Proceedings of SIGIR '03, 26th ACM International Conference on Research and Development in Information Retrieval, 2003

[19] Oren Zamir, Oren Etzioni, "A Dynamic Clustering Interface to Web Search Results", Proceedings of Eighth World Wide Web Conference, 1999

[20] Oren Zamir, Oren Etzioni, "Web Document Clustering: A Feasibility Demonstration", Proceedings of SIGIR '98, 21st ACM International Conference on Research and Development in Information Retrieval, 1998

[21] Tian Zhang and Raghu Ramakrishnan and Miron Livny, "BIRCH: an efficient data clustering method for very large databases", ACM SIGMOD International Conference on Management of Data, 1996

9. APPENDIX

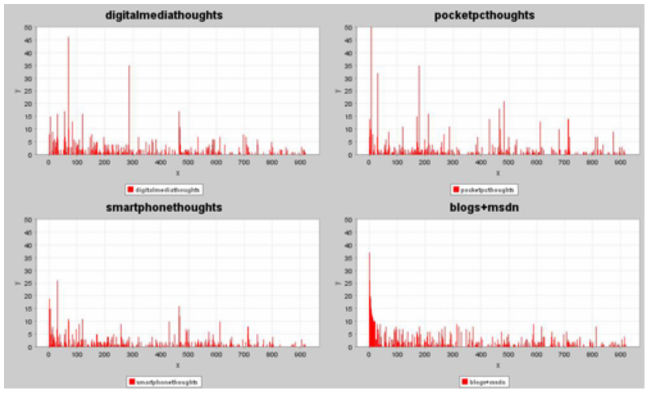


Figure 9 - Cluster containing tech blog sites like DigitalMediaThoughts, PocketPCThoughts, SmartphoneThoughts and MSDN Blog

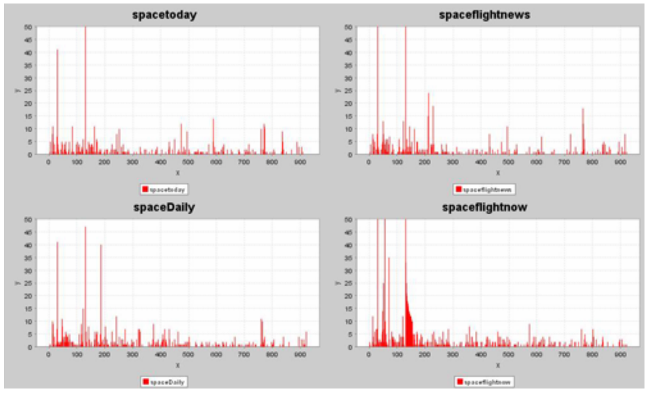


Figure 10 - Cluster containing several astronomy related sites like spacetoday.com, spaceflightnews.com, spacedaily.com and spaceflightnow.com