Using Visual Features for Fine-Grained Genre Classification of Web Pages

Ryan Levering, Michal Cutler, and Lei Yu
SUNY at Binghamton
ryan.levering@binghamton.edu, cutler@binghamton.edu, lyu@cs.binghamton.edu

Abstract

The field of automatic genre classification has primarily focused on extracting textual features from documents. The goal of this research is to investigate whether visual features of HTML web pages can improve the classification of fine-grained genres. Intuitively it seems that this should be helpful and the challenge is to extract those visual features that capture the layout characteristics of the genres. A corpus of Web pages from different e-commerce sites was generated and manually classified into several genres. Three different sets of features were compared - one with just textual features, one with HTML level features added, and a third with visual features added. Our experiments confirm that using HTML features and particularly URL address features can improve classification beyond using textual features alone. We also show that adding visual features can be useful for further improving fine-grained genre classification.

1. Introduction

Web search, as one much studied field, uses query terms to attempt to match a user's information need to the content of a particular document. Only recently have commercial search engines focused on goal-oriented interpretations of queries. The main advantage of this is that most users' web goals are satisfied through an intersection of document content and document purpose. Genre classification is an attempt to categorize the purpose of a document.

For example, a query for a particular brand of printer would seem a very well-defined search query. Yet while the content of the needed page is defined, the user's need is not. One could be searching for a driver, looking for a help manual, looking to buy the printer, or looking to post a review of the printer. This dimension is not very well addressed in modern search techniques. The addition of a content "snippet" (a small piece of the page returned with the result) does assist in this task, but can often still be ambiguous. Using the classified genre of a document to assist in retrieval tasks could potentially be very useful. This research does not investigate the practical application of this classification, but rather the improvement of current classification techniques.

The concept of "genre" has been around in literature for many years, as in [2]. Research from information science, like Crowston's seminal work in [4], attempted to define genre in its Web context by empirically evaluating the genre of random Web pages. The authors work also contains an in-depth discussion on the communications background of genre analysis. For a larger study on user's perceptions of genres, their usefulness, and work on defining a more complete genre palette, see [13]. [14] suggests some applications of a mapping of Web page genres to search goals.

The goal of this research was to investigate whether visual features of HTML pages can improve the classification of fine-grained genres. Intuitively it seems that visual features should be helpful to identify the genres of Web pages which often have characteristic layout patterns. To capture these patterns we needed to determine what are the important layout features that should be extracted, design methods to extract them and finally investigate whether they improve classification and by how much. In this paper we describe many such features and some classification benefits. We also provide some discussion on the additional cost of obtaining these features.

Shepherd and Watters argue in [16] that the number of genres on the Web keeps growing, and that some genres are difficult to classify, even for viewers. We concur with their opinion that it may be impossible to come up with a comprehensive set of Web genres, so that every future Web page will fall neatly into one or more of the genres in the set. We also believe that it is useful to classify those genres for which search engine users are commonly looking.

Because of the wide variation in genre palettes that are currently used by researchers, there is a similarly wide variety of granularity of such palettes.

1530-1605/08 $25.00 © 2008 IEEE

This is mentioned by several researchers, with some good thought on this referenced by and in Santini's [15]. She believes that a consistent granularity has a positive impact on classification accuracy. Her research, and indeed most research that attempts to identify a universal set of web genres, typically focus on coarse genres. While we see applications for a coarse genre classification, we believe fine-grained classification has more directly practical uses.

In this research we limited the corpus to web pages of e-commerce stores. While e-commerce stores contain many genres, we chose to classify only three of them which we consider to be useful: homepages, product lists and product descriptions.

We experimented with two different classification scenarios. In the simpler scenario the corpus includes only documents that belong to the selected subset of genres and the goal is to classify each page as belonging to one of the genres. In the more realistic scenario, the corpus is clouded with a large number of documents that do not belong to any of the chosen genres. For both of these problems we classify each genre against its complement. Then we use a simple ensemble voting scheme to apply the binary classifications to multi-class problems.

Current implementations of automatic genre classification are usually based on the simple scenario where the corpus includes only documents that belong to the chosen subset of genres. In the pure text area of research, [5] gets significant results in a multi-class genre classification problem. [17] gets satisfactory results on a genre classification task with only a common word list, which is great evidence of the difference of genre classification from categorical content classification. [1] gets very accurate classification on Web pages, but focuses on an easier classification problem, with fairly particular genres not specific to the Web. [11] use a more realistic Web related set of genres, but get less accurate results. Research in the area consists of a wide variation in the genres selected for classification and the features that are used to classify. The most closely related experiment may be [7], which uses an interesting set of HTML and text features to do home page classification. The main difference in this research, beyond the genre palette, is the use of rendered features and the presence of a much larger noise set.

One common element in these approaches is that they use only text or low-level HTML features in their classification. Several of the researchers anticipate that presentation features may be important but they either only address low-level presentation features or leave the work for the future. We believe that users often use visual cues in their interpretation of page genre. In Figure 1, two product description pages from mainstream commercial sites are shown on the left. On the right is an outline of the pages with elements that may contribute to the users' detection of the genre. Semantically similar elements have been labeled with the same letter (w, x, y, or z). These elements tend to occur in recognized patterns, such as the presence of a large image (labeled x) near the top of the page for the product picture.

While using HTML tags to approximate visual elements might achieve some improvement over text alone in classification, including features based on a visual rendering of the page should be more beneficial. In this research, we use several methods to construct visual features from a rendered HTML document, using both the *position* of more traditional features (links, text, emphasis tags, etc.), as well as a look at other non-textual content. [10], a survey of the web, found that non-textual content accounts for a majority of many web pages' visual area. Therefore, being able to use features to represent non-textual content is important.

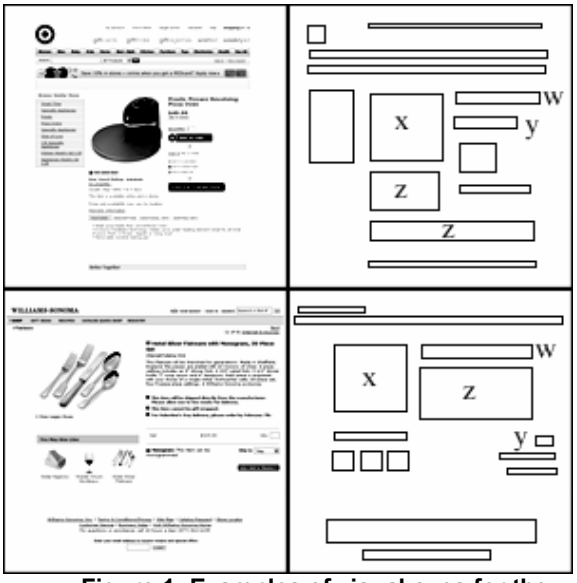

Figure 1: Examples of visual cues for the Product Description genre

In section 2, we outline the classification task we chose to investigate i.e., the usefulness of visual features. The methodology for the corpus creation and manual class labeling is also discussed. In section 3, we explain the features we chose to use and the methods we followed to collect the visual ones. In section 4, we evaluate the worth of the features in distinguishing our chosen genres. This is done through a logical series of classification tasks and feature selection methods.

2. The Framework

In order to focus on the feature construction, we purposely chose a task where the genres were relatively easy to determine and limited in their generality. The goal was to train classifiers so that spiders that crawl a commercial store domain could recognize particular genres, in order to eventually extract information from it or decide when to return for updated information, etc. Systems constructed to recognize a genre on specific sites may achieve the most practical results, but we wanted to achieve acceptable results on a more general level.

We chose three genres for the task: store homepages, store product lists, and store product descriptions. These are very frequently occurring genres for online shopping sites, for which many users share perceptions in terms of page content and layout. Two people manually compiled a list of agreed upon URLs for each genre after given an example of each type of genre. In addition, they collected a larger set of URLs that did not fit into one of these genres. These coders did not know the goal of the research and thus did not "clean" the list or purposely avoid poor examples. The websites were all large, popular stores (that had a non-Web presence as well) that sold or at least advertised their products online. Typically, for each website chosen we included one example of each genre.

While using visual features to improve genre classification is the main focus of the paper, we also planned our experiment to be a better representation of a useful web genre classification task. As a classification task, general purpose genre classification on the Web is an incredibly unbalanced, noisy goal. Most of the previous research has achieved good results by setting up their experiment to be a question of page classification when the page is known to belong to one of the classes. This is sometimes known in the literature as a genre sorting problem. The weakness with this approach is that either 1) the genres are too general to be classified with much accuracy, or 2) the genres chosen do not cover the whole universe of pages to be classified in a practical system. Therefore, as discussed above, we believe that it's far more useful to have finer-grained, more precise binary classifiers that can distinguish particular genres from a large negative class. In our experiment, we collect a large set of pages to serve as a negative class to better train and evaluate a general-purpose classifier.

Table 1 lists the number of documents collected for each of the three genres. It also shows the number of documents in the "store other" subset. One benefit of restricting our task to a certain type of commercial website was that we could define a much more complete negative class, something a general purpose classifier couldn't always count on. Several examples of negative class pages are: help/FAQ, special offers, department entries, company policy, and "about the company".

Table 1: Experimental Datasets

#	Dataset	# instances
1	Store Homepages	179
2	(Store) Product Lists	167
3	(Store) Product Descriptions	155
4	Store Others (Negative Class)	798

3. Feature Construction

As in most automatic classification goals, the quality of genre classification is highly dependent on the selected features. If the right features are not chosen, no amount of training or feature selection will ever improve accuracy without producing a spurious, potentially over fitted model. For example, a classifier to recognize FAQ documents will suffer greatly without the word "faq" as a feature.

3.1 Textual Features

For vocabulary, [17] established the importance of common words, stop words, and punctuation in determining genre. We included a stop word list, Stamatatos' common word list, and all of the punctuation characters as features. We also collected the full set of Porter-stemmed word counts in each page as dynamic vocabulary features. These would be important for finding useful key words like our "FAQ" example above.

Grammar analysis was done to recreate the feature set collected by [1]. A part-of-speech tagger, syllable counter, and tokenizer were used in combination to generate readability measures as well as a battery of statistics on the makeup of text. This ranged from the simple total word count feature to the number of sentences that begin with determiners. The complete list of text features is found in Table 2.

Table 2: Textual Feature List

Readability: KCD, ARI, C_L, FLESCH, FOG, LIX, and SMOG
Counts: Characters, Words, Unique Words, Sentences, Paragraphs
Averages: Sentence Length, Word Length, Paragraph Length, Syllable Length
Sentence Length: Maximum, Minimum, Over 28, Under 13

POS Counts: Conjunctions, Pronouns, Prepositions, Nominal verbs, Auxiliary verbs, "to be" verbs
Punctuation Characters (dynamic per page)
Every Stemmed Word (dynamic per page)

3.2 HTML Features

HTML features can be drawn from the raw DOM without significant processing. This included individual tag counts, tag depths, and "table" tag depths. In addition, we also included some script features (number of tags with JavaScript, unique function calls, etc.), an in-page link analysis (number of external vs. internal domain links, total number of links, etc.), and some form analysis (submission destination, element counts).

We also included features extracted from the URL address of the Web pages. These features included not only static counts of URL length and the number of path levels in the URL (i.e. the number of slashes), but also a dynamic list of the vocabulary found in the URL address. This was formed by taking any consecutive sequence of alphabetic characters, stemming it, and using it as a feature. The complete list of HTML features is found in Table 3.

Table 3: HTML Feature List

Link Counts: Total, Domain, External-domain, Server, Internal, Mailto, FTP, Other
Form Element Counts: Form Tags, Hidden Form Inputs, Combo-boxes, Lists, Text-fields, Passwords, Text areas, Buttons, Checkboxes, Radio buttons, File uploads
Form Link Counts: Domain, Extra-domain
Tag Counts: Total, Emphasis (bold, italics, underline), Font, Script, Table, Paragraph, Line-break, Block-quote, Image, Horizontal rule
HTML Depths: Max HTML depth, Max table depth
Script Counts: Unique events, Link events, External script tags, Internal script tags, onMouseOver, onSubmit, onClick, onMouseMove
URL Lengths: Total URL length, Directory path length
Every Stemmed Alphabetic URL Sequence (dynamic per page)

3.3 Visual and Visually Central Features

The common thread in all of the visual features is that they carry some context about the rendered location and size of an object from the Web page. Rendering engines have become very fast and while there will undoubtedly be a performance cost for implementations, this research will show potential benefits of having this information.

We currently do not feel confident enough in the efficiency of our analysis library to include a runtime comparison of the different feature sets. However, we believe that the critical cost of this approach is the cost of downloading of associated document content. Preliminary extensions to our web survey research [10] and examination of the test data collected for this research demonstrated that associated image, script, and style file sizes are a significant factor compared to the compressed size of an average HTML file. Laying out the document, in our experience, is not considerably slower than building HTML models of the document. Furthermore, the layout features we collect from the model are actually less time intensive to gather than the textual features, with readability analysis that is currently done.

It was important to our research to use an established, commercial renderer in order to get accurate tag position/size information. For our engine, we used JRex [9], which is a collection of Java wrapped JNI interfaces for the Gecko rendering engine that Mozilla uses. The document is rendered by Gecko and then a position enhanced DOM is returned via JavaScript from which to extract features. An added benefit to this method is that script-interpreted page contents are returned, which can often differ slightly from the text found in the HTML. The visual features fall into several categories: image counts and statistics, area based features, and visually central features.

3.3.1 Image counts and Statistics. As bandwidth increases on the Internet, it becomes easier for Web authors to include more image media. It is very common to find pages having text within images and pages without many words at all. The distribution of image elements on a page could be a visual cue that viewers use to determine genre. Using image features to improve textual web search was investigated in [19], though the image features they used were much more sophisticated image content features using latent semantic indexing. Because the type of image often indicates its usage (photographic JPEGs versus iconic GIFs) separate image counts were maintained by us for those types. We also detected animated images as well as images with progressive loading like interlacing in JPEGs.

Several statistics were calculated on total image distribution as well as specifically for GIFs and JPEGs. Minimum and maximum image sizes could be useful for detecting deviant image sizes that indicate genre. We also included a standard deviation, average, median, and mean of all the image sizes. When combined with the central area filtering described subsequently, these give a characterization of the important image distributions.

3.3.2 Area-based features. For this set of features, we sought to use the measurement of the general page layout to assist classification. The idea was inspired by optical scanning research [6] where content density histograms were used to assist in top-down blocking of a document during text extraction.

The main content types on a Web page are defined as text, image, form, and object. This content breakdown was used in [9]. The actual areas of each element were used, rather than perceived area. For instance, a series of bullet points in a row may be perceived to be one continuous text block, but spaces between the rows would not be included in the totals.

A large number of different area based statistics were also collected. In table 4 we list the following features: page dimensions, the total areas of different object types, percentages of the area of each object relative to the total area of the object type, as well as relative areas of each object to the sum of the total areas used by text, image, form and object. By projecting content areas to a single dimension, we can get additional information of how the visual elements are spread along both the X and Y dimensions. We can also see how the content focus of the document changes across its width or length. Thus if the genre was the type of page that often had a single image at the top followed by a heavy text passage, the top would have a strong image projection and weak text projection to the top of the y-axis of the document and the ratio would reverse as we got closer to the bottom.

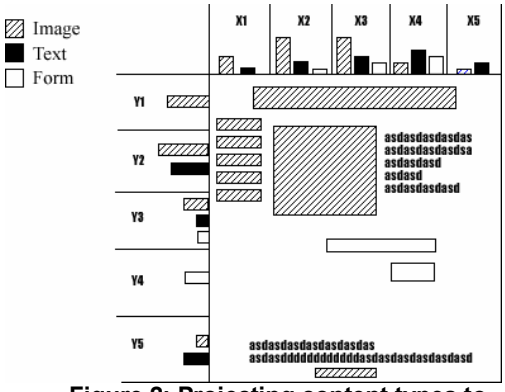

Figure 2: Projecting content types to dimensional buckets

Each page was broken into five proportional segments in each dimension as illustrated in Figure 2. The area of each content type within each segment was extracted to discretize the layout flow across the page. A feature for these non-normalized areas in each content type bucket was included. To assist in classification, two other features for each bucket were used: percentage bucket area (for example text bucket area*100 / total bucket area where total bucket area = (text bucket area + image bucket area + form bucket area + object bucket area)) and relative bucket area (for example, text bucket area / total bucket area).

3.3.3 Visually Central Features. Using the position of text in a page to increase its importance was first shown by [8], and later improved upon by the VIPS algorithm developed by Microsoft's [3] and visual style trees [18] which used more detailed page segmentation to achieve more accurate results. In our task, a similar idea was applied to the features. As shown in the last rows of Table 4 we took all the basic features and where it was possible and reasonable, extracted the location and area information for only those that occurred within the central area of the page. This has the main benefit of removing the noisy header/sidebar/footer information from the analysis of the page. The outside one-fifth of each dimension was used as the cutoff for unimportant area. Any element whose center existed in that border area was not included in the features.

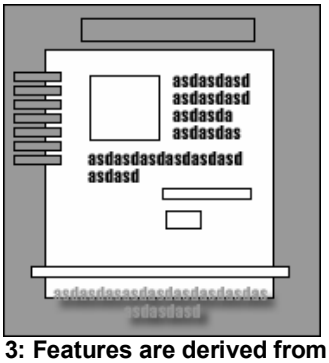

Figure 3: Features are derived from the center area

This filter was applied to all the link counts, visual form element counts, and to the image analysis that is discussed in the following section to produce additional features. A feature for each vocabulary term was also produced with the term frequency in the central area. It is important to note that we did not replace the original features, since it is more than possible that a genre might have visual cues on the border that are positive indicators.

Table 4: Visual Feature List

Image Counts: BMPs, GIFs, animated GIFs, progressive, JPEGs, PNGs, Unique GIFs, Unique JPEGs
Image Statistics: Minimum, Maximum, Mean, Median, and Standard Deviation for GIF, JPEG, and Total
Page Dimensions: Height, Width
Total Areas: Form, Link, Text, Object, Image

Above the Fold (top 800 pixels) Areas: Form, Link, Text, Object, Image
Percentage Areas: Form, Link, Text, Object, Image
Relative Areas: Form, Link, Text, Object, Image
Area projections for Image, Text, Form, Object, and Link Bucket Areas: X1-X5, Y1-Y5 (see Figure 2)
Image Placement: Every Central Image Count/Statistic Feature (see above)
Link Placement: Every Central Link Count (see Table 3)
Form Placement: Every Central Form Element/Link Count (see Table 3)

4. Empirical Study

The main goal of our experiments was to look into how different combinations of feature sets performed in both of our classification scenarios. At the same time, we wanted to make sure that classification performed well enough to be useful.

4.1 Feature Sets

We chose an incremental approach to the selection of feature sets in our experiments, based on the feature groups explained in the last section. Initially we use pure text as would be used in non-Web genre classification. Then we run an experiment using all the textual features, plus the HTML level features. Most Web genre classification research currently uses something very similar to a subset of these features. Finally, visual features are added to create another experimental set of features.

Table 5: Feature Sets compared in evaluation

Feature Set	# of Features
Text	~10,000
Text + HTML	~11,500
Text + HTML + Visual	~12,500

This incremental way of looking at the problem is a very practical approach. The higher level models of data, moving from a text model to an HTML model or especially going to a rendered model over an HTML model, carry with them an increase in processing complexity. The question is whether the benefit justifies the computation cost.

4.2 Feature Ranking

Feature ranking is the method of ranking each individual feature by its ability to differentiate between classes. We used information gain to pre-select features for classification. Information gain is a measurement of the change in entropy when a particular feature is used to divide the data. Features with high information gain most likely are highly important in determining the genre of a given page. We wanted to use the same process to select features for the harder problem.

Before doing so, we examined the features selected to see their usefulness in the comprehensive task; In other words, given all twelve thousand features, which features would be most likely chosen to distinguish a genre from the universe of documents.

One large disadvantage with feature ranking is that it does not take into account positive and negative feature relationships. Therefore, if you take the top one hundred features, it does not mean those are best one hundred as a set. It means that those are the top one hundred features selected incrementally. However, it does give an idea of which features have possibilities of being included in an eventual set and serves as an initial culling of the problem space.

The graphs shown in the figures below illustrate the ratios of the types of features selected as a percentage of the running total number of features selected. Therefore, the ratio at the very right side of the graph would be the ratio of features in the top 100 that were eventually used for classification.

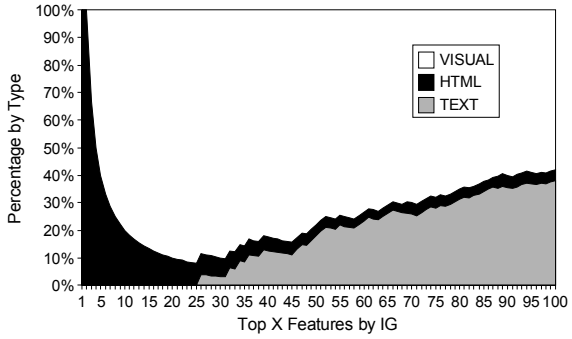

Figure 4: Feature Ranking Ratios in *Store Homepage* Discrimination

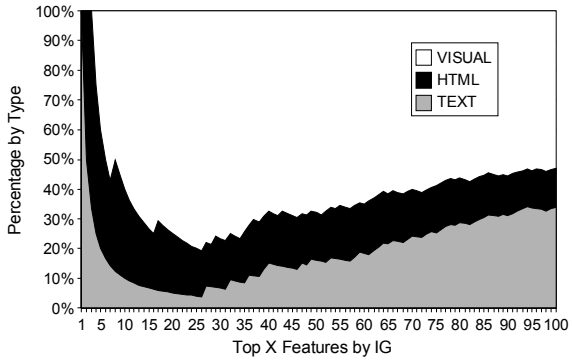

Figure 5: Feature Ranking Ratios in *Product List* Discrimination

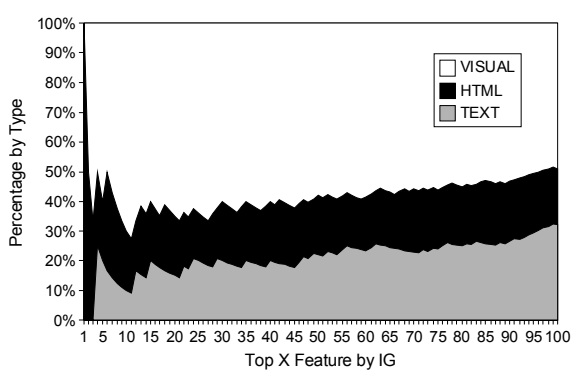

Figure 6: Feature Ranking Ratios in *Product Description* Discrimination

One can see that visual features are important at some level to all of the classification tasks. While the features weren't usually the very top features, they definitely seem to be able to play a role in explaining the data space. That being said, more of the visual features are naively redundant, which will bias these ranking results. In addition, the Homepages ranking was especially interesting in that with the addition of the negative class, text features do not show much worth and are not in the top twenty-five features at all. This probably indicates negative documents hurting the discriminating power of words found in store homepages.

4.3 Simple classification scenario - sorting

To begin with, we wanted to make sure our features performed well on the easier scenario of genre sorting (i.e., recognition). That is, given a document we know is in one of the three genres, select the correct genre. Given our focused genre set, we predicted this to be easily accomplished.

Before classification, we did a feature selection pre-processing step where we selected the top 100 features by information gain. While feature ranking is often used to improve classifier performance, our main goal was runtime performance. The comprehensive feature set that we use has over twelve thousand features, albeit many rare useless ones. Running a classifier on that many dimensions was impractical, especially when most of them were sparse vocabulary features.

For classification we used the Weka data mining API with an oversampling linear-kernel SVM classifier and ten fold cross validation. Since a SVM is fundamentally a binary classifier, to handle a multi-class problem it is necessary to do several binary classifications and then use a method to aggregate the results.

For the multi-class classification, we used a simple voting/consensus scheme between several one-versus-rest (complement) binary classifiers:

- Homepages: Dataset 1 v. Dataset 2+3
- Product Lists: Dataset 2 vs. Dataset 1+3
- Products: Dataset 3 vs. Dataset 2+1

An ten-fold cross validation was used on the overall process. Therefore, ninety percent of each dataset was used to train three binary classifiers as above. The remaining ten percent of each dataset was used to evaluate the ensemble approach. Each testing instance was classified against all three binary classifiers. If there was one positive classification, this was the chosen genre. If there were multiple positive classifications, the one with the highest probability according to the SVM was chosen. If none of the classifiers returned a positive classification, the classifier that had the lowest probability for the negative class was used. In the following tables, the percentage and number of instances of the genre indicated by the row were classified as the genre indicated by the column.

Table 6: Text Features – Confusion Matrix

	SH	PL	PD
Store Homepage	.816 (146)	.100 (18)	.084 (15)
Product List	.191 (32)	.713 (119)	.096 (16)
Product Description	.258 (40)	.052 (8)	.690 (107)

Table 7: +HTML Features – Confusion Matrix

	SH	PL	PD
Store Homepage	.955 (171)	.028 (5)	.017 (3)
Product List	.042 (7)	.844 (141)	.114 (19)
Product Description	.019 (3)	.135 (21)	.846 (131)

Table 8: +Visual Features – Confusion Matrix

	SH	PL	PD
Store Homepage	.972 (174)	.022 (4)	.006 (1)
Product List	.036 (6)	.868 (145)	.096 (16)
Product Description	.019 (3)	.090 (14)	.890 (138)

The results across feature sets give an idea of the general trend of classification improvement. Adding HTML level features noticeably increases the overall system F-measure from 85.2% to 93.9% and is especially useful with homepage classification. Adding visual features to these further increases the F-measure to 95.4%, demonstrating how useful visual features can be with some of the more difficult classification cases. These results were promising, but we still had to see if this worked for a more realistic scenario.

4.4 More realistic classification scenario - discrimination

In this experiment, we introduced a large negative class of documents that do not belong to any of the selected and labeled three genres. This should in theory, produce a classifier that would have a better chance of looking at a document within the sampled online store universe and deciding not only which genre to put it into, but also whether it belongs to one of the genres at all.

For this problem, we are again interested in using several binary classifiers to choose a single genre for a document. Therefore, all of the discussion in this section is based on forming a consensus between the following three unbalanced binary classifications:

- Homepages: Dataset 1 vs. Dataset 2+3+4
- Product Lists: Dataset 2 vs. Dataset 1+3+4
- Products: Dataset 3 vs. Dataset 1+2+4

For the actual classification, we first applied the information gain feature ranking discussed above to choose the top 100 features. Then, because this problem was a bit more difficult, we performed an additional feature subset selection using a combination of best-first correlation based feature selection and SVM-based wrapper subset selection. These feature selections were more time-intensive but were especially useful in eliminating many of the less useful features that impeded classification. After feature selection, we proceeded to use a similar classifier setup as discussed with the simpler task: linear SVM, balanced classification, with 10-fold cross validation on the overall process. We also did some early experiments with both a J4.8 decision tree classifier and a Naïve Bayes classifier. The NBC did not perform as well as the SVM, and the decision tree, while performing similarly well after feature selection was not included to keep the focus of the research on the feature sets.

In a real-world system, the voting/consensus step would not necessarily be required. If we allowed for multiple genre assignments for each document, we would just be using the individual classifiers. However, we wanted to take the extra step of ensuring that if a single class was required for a document, the results would be acceptable. This problem is actually an aggregation of the individual genre classification problem and is more difficult. Furthermore, the multi-class results better illuminate intra-class confusion as shown in a confusion matrix.

The classifier voting was done in a manner similar to the simpler scenario. The main difference was that instead of having to choose a classifier when none of them returned a positive classification, a "none of the above" classification was now allowed. Ideally, only the fourth noise dataset should have documents that produced such a classification result.

The confusion matrices for the classification with different feature sets are shown below.

Table 9: Text Features – Confusion Matrix

	SH	PL	PD	None
Store Homepage	.486 (87)	.095 (17)	.419 (75)	0
Product List	.204 (34)	.707 (118)	.323 (11)	.024 (4)
Product Description	.290 (45)	.077 (12)	.323 (50)	.310 (48)
Store Other	.312 (249)	.046 (37)	.026 (21)	.615 (491)

Table 10: +HTML Features – Confusion Matrix

	SH	PL	PD	None
Store Homepage	.994 (178)	.006 (1)	0	0
Product List	0	.754 (126)	.060 (10)	.186 (31)
Product Description	0	.071 (11)	.735 (114)	.194 (30)
Store Other	.008 (6)	.050 (40)	.030 (24)	.912 (728)

Table 11: +Visual Features – Confusion Matrix

	SH	PL	PD	None
Store Homepage	.989 (177)	.011 (2)	0	0
Product List	0	.832 (139)	.042 (7)	.126 (21)
Product Description	0	.058 (9)	.716 (111)	.226 (35)
Store Other	.008 (6)	.034 (27)	.028 (22)	.931 (743)

Immediately, it's evident that textual features perform very poorly on their own, particularly in the store homepage and product description classification. Features that could potentially be useful in the simpler problem, like the size of the page or the presence of certain general product words because far less useful in the presence of the larger set of negative documents.

Both of these poorly performing classifiers improve dramatically with the introduction of the HTML features, in particular the URL-based features. Classifying store homepages with URL features is a naïve problem within our universe as they will always have very short request URLs compared to the other genres. In the case of product descriptions, the addition of URL tokens such as "product" make it a much easier problem.

Interestingly enough, both of these genres gain very little from visual features. In the case of store homepages, the extra features can only make the classifier worse. In the case of product descriptions, this could be either because the genre does not have a consistent layout pattern or the features that we are extracting do not define this pattern strongly enough when the noise is added. We believe it to be the latter and are working on defining more complex analysis techniques to capture these patterns.

The classification task that benefits the greatest from visual features is the product list classification. This task actually improves much further from visual features than from HTML features The product list documents in our corpus are more consistent in their appearance and contain very few optional visual sections that make a classification like the product description one more noisy. In fact, after feature subset selection, more than half of the features chosen when all are available are visual features.

Overall, the usefulness of visual features in a very noisy classification appears to be very genre dependent. Only one of the three classifications showed significant gains. Our results indicate that while visual features can be useful, there is still further work to be done on capturing the semantics of fine-grained genres. Further conclusions on this are left to the following section.

5. Conclusion

Genre classification can be useful in web search, as well as other IR tasks. On the Web, genres are particularly visually-oriented. This paper deals with the challenging problem of using such visual features for improving genre classification. We have demonstrated that visual features can be beneficial for a noisy genre classification problem. In addition, in a less noisy problem with a smaller universe like in our first experiment, these features become more useful.

While we have shown that these features can be useful in certain genres, the more important question is whether these runtime intensive features are useful for genre classification on a Web-sized scale. The answer to that question depends largely on the way automatic genre classification will be used and defined in its transition to the World Wide Web.

Genres that do not depend on a *form* aspect will obviously not find area or image features very helpful. In these cases it is expected that at the least the use of visually central features and more complex segmentation algorithms to locate the genre-specific textual and HTML features could improve results as it has in other fields. In addition, it is highly unlikely that visual features will be as successful with very coarse genres. A fine-grained problem was chosen for this research because we believed it would show the most noticeable gains, having a stronger visually archetypal semantics. Coarse genres are composed of many sub-genres that may have their own unique visual characteristics. However, that does not mean that coarser genre problems could not benefit from these features indirectly in classification schemes.

For many fine-grained genre tasks though, it is our belief that the conclusions of this hypothetical experiment can be projected. The underlying reason for this is that author goals on the World Wide Web have changed from the traditional raw information transfer of text documents to a more complex level of interaction. Genres are innately tied to the communicative intent of the media author and as the technology to express that intent changes, analysis methods must improve to keep up. Authors use layout information on web pages to communicate relationships between the visual elements in ways that a basically one dimensional text document cannot. Just as text style and content patterns are used as a convenience to facilitate communication with the reader, layout patterns are used to facilitate interaction with the Web viewer. These layout patterns can be exploited to assist in determining the author's intent for the document and thus the document genre.

Further research is needed in using visual features within genre classification. This work should be expanded to include a wider range of corpuses with varying genre palettes. In addition, one of the strongest benefits of visual features is that they are inherently non-textual and therefore language independent. This idea should be investigated in the context of multi-lingual corpuses.

Additional general visual features need to be investigated and evaluated. In the process of our feature construction, there were several more semantically complex visual characteristics that we wanted to capture for this particular experiment. Expanding the work to other genre palettes may give some more insight into useful visual features.

Finally, as genre classification on the Web is still in its infancy, most research spends a lot of effort proving that it can be done and less time examining potential applications of the field. Further research is needed to evaluate practical applications of genre classification toward fields such as information retrieval.

6. References

[1] Boese, E. 2005. Stereotyping the Web: Genre Classification of Web Documents. Masters thesis, Dept. of Computer Science, Colorado State University, Boulder, Colorado.

[2] Butcher, S. H.. ed. 1932. *Aristotle's Theory of Poetry and Fine Arts with The Poetics*. 4th Edition. London: MacMillan.

[3] Cai, D., Yu, S., Wen, J.-R. and Ma, W.-Y. 2003. *VIPS: a vision based page segmentation algorithm*, Microsoft Technical Report, MSR-TR-2003-79.

[4] Crowston, K. and Williams, M. 1997. Reproduced and emergent genres of communication on the World-Wide Web. In *Proceedings of the 30th Hawaii International Conference on System Sciences*, 30. Washington, DC: IEEE Computer Society.

[5] Dewdney, N., VanEss-Dykema, C., and MacMillan, R. 2001. The form is the substance: Classification of genres in text. In *Proceedings of ACL Workshop on Human Language Technology and Knowledge Management*, 1-8. Morristown, NJ: Association for Computational Linguistics.

[6] Ha, J., Haralick, R.M., Phillips, I.T. 1995. Recursive X-Y cut using bounding boxes of connected components. In *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 2)*, p.952, August 14-15, 1995.

[7] Kennedy, A. and Shepherd, M. 2005. Automatic Identification of Home Pages on the Web. In *Proceedings of the 38th Hawaii International Conference on System Sciences*, 99. Washington, DC: IEEE Computer Society.

[8] Kovacevic, M., Diligenti, M., Gori, M. and Milutinovic, V. 2002. Recognition of Common Areas in a Web Page Using Visual Information: a possible application in a page classification. In *Proceedings of 2002 IEEE International Conference on Data Mining*, 250. Washington, DC: IEEE Computer Society.

[9] JRex 2007. http://jrex.mozdev.org/.

[10] Levering, R. and Cutler, M. 2006. The portrait of a common HTML web page. In *Proceedings of the 2006 ACM symposium on Document engineering*, 198-204. New York, NY:ACM Press.

[11] Meyer zu Eissen, S., Stein, B. 2004. Genre Classification of Web Pages. In *KI-2004: Advanced in Artificial Intelligence*, 256-269. Heidelberg: Springer Berlin.

[12] Peng, H., Long, F., & Ding, C. 2005. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(8), 1226-1238. Washington, DC: IEEE Computer Society.

[13] Rosso, M.A. 2005. Using Genre to Improve Web Search. Ph.D. diss., School of Information and Library Science, University of North Carolina, Chapel Hill, North Carolina.

[14] Roussinov, D., Crowston, K., Nilan, M., Kwasnik, B., Cai, J., Liu, X. 2001. Genre Based Navigation on the Web. In *Proceedings of the 34th Hawaii International Conference on System Sciences*, 4013. Washington, DC: IEEE Computer Society.

[15] Santini, M. 2006. Common Criteria for Genre Classification: Annotation and Granularity. Workshop on Text-based Information Retrieval (TIR-06), In Conjunction with ECAI 2006, Riva del Garda, Italy.

[16] Shepherd, M. and Watters, C. 2004. Identifying Web Genre: Hitting A Moving Target. In *Proceedings of the WWW2004 Conference. Workshop on Measuring Web Search Effectiveness: The User Perspective*.

[17] Stamatatos, E., Fakotakis, N., and Kokkinakis, G. 2000. Text Genre Detection Using Common Word Frequencies. In *Proceedings of the 18th International Conference on Computational Linguistics*, 808-814. Morristown, NJ: Association for Computational Linguistics.

[18] Yi, L. and Liu, B. 2003. Eliminating Noisy Information in Web Pages for Data Mining. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 296-305. New York, NY:ACM Press.

[19] Zhao, R. and Grosky, W.I. 2002. Narrowing the Semantic Gap – Improved Text-Based Web Document Retrieval Using Visual Features. In *IEEE Transactions on Multimedia*, 4(2):189-200. Washington, DC: IEEE Computer Society.