ABSTRACT
In this paper, I propose a refined and incremental centroid-based approach for genre categorization of web pages. My approach is based on the construction of genre centroids using a set of training web pages. These centroids will be used to classify new web pages. The originality of my approach is the implementation of two new aspects, which are refining and incrementing. My approach is based on the combination of three information sources, which are the URL address, the logical structure and the hypertext structure. Conducted experiments show that the proposed approach is very fast and provide micro-average accuracy more than 95%, which is better than those obtained by other works on genre categorization of web pages and other machine learning techniques.

Categories and Subject Descriptors
I.2 [Artificial Intelligence]: Natural Language Processing, Learning; H.3 [Information Storage and Retrieval]: Information Storage.


Keywords: Categorization, genre, centroid, refining, incrementing, combination.

1. INTRODUCTION
In front of the explosive growth of the number of web pages, users cannot quickly find desired information among the huge list of web pages returned by a search engine. To deal with this problem, many recent works have interested with genre categorization of web pages [17].

Generally, the genre reflects the style of the document. Many definitions of genre have been proposed, which concerns non-digital documents. For web pages, the most known definition is proposed by Shephered and Watters [20] which characterized the genre of web pages, (also called cybergrenre) by the triple <content, form, functionality>. Content attribute is the topic of web page, and the form represents the physical and linguistic features of the web page. Functionality attribute is used only for web pages and he describes interaction between users and web pages.

Due to the fluidity and the fast-paced evolution of the web, many authors studied genre evolution [4, 19, 17]. Crowston and Williams used a collection of web pages randomly selected to study genre evolution [4]. They identified four types of genres: reproduced genres, adapted genres, novel genres and unclassified web pages. Reproduced genre is often borrowed from other media. Adapted genre means the adaptation of a traditional genre to the needs and capabilities of a new media. The authors report that most web genres are reproduced and adapted. An equal number of new genres and unclassified web pages are recorded.

Shepherd and Watters [19] identified two types of genres: extant and novel genres. Extant genres are genres that exist in some other medium. Novel genres can be evolved from extant genres or they can be completely original.

Santini combined the genre types obtained in [4] and [19] to define a new schema of web genres that contains five types of genres: reproduced genres, adapted genres, novel genres, spontaneous genres and unclassified web pages [17].

To take into account the evolution of web genre over time, I propose in this paper a new approach for web page genre categorization which based on two new aspects: refining and incrementing. Refining means that my approach eliminates noisy web pages to construct genre centroids. A noisy web page is a web page whose similarity to the centroid of the corresponding genre is below some refining threshold value. Incrementing consist in classifying web pages one by one. At each web page my approach refines current genre centroids.

This paper is organized as follows: Section 2 presents in chronological order recent works on genre categorization of web pages; Section 3 describes the principal of centroid-based categorization; Section 4 describes the different steps of my approach; Section 5 presents conducted experiments and discuss obtained results; finally, in section 6, I draw my conclusion and outline my future work.

2. RELATED WORKS
The feature set and the machine learning technique are two factors that distinguish between all studies on genre
categorization. In this section, I present, in chronological order recent works on genre categorization of web pages.

Meyer Zu Eissen and Stein [14] compiled the KI-04\(^1\) dataset, which contain 1209 web pages distributed over eight genres. They used different kinds of features including presentation-related features (html tag frequencies), closed word features (names, dates, etc.), text statistics (punctuation mark frequencies) and syntactic features (part-of-speech frequencies). Using SVM technique, they report 70% average classification accuracy.

Kennedy and Shepherd [10] consider only one genre and its sub-genres. Using a dataset of 321 web pages and a neural network, they attempted to discriminate between home pages from non home pages. On a second level, they classify home pages into three categories (personal, corporate, and organization). Their feature set comprises features about the content (e.g., common words, meta tags), form (e.g., number of images), and functionality (e.g., number of links, use of Javascript). The best reported results were for personal home pages.

Boese and Howe [1] studied the effects of web page evolution on genre classification using KI-04 and WebKB datasets\(^2\). They used three measurements of change, which are accessibility, last modification dates and degree of page similarity. They used features including style (e.g., part-of-speech frequencies), form (e.g., html tag frequencies) and content (e.g., bag-of-words). Using logistic regression technique, they report 79.6% accuracy for a new version of WebKB dataset.

Santini [17] studied web page genre classification based on three different feature sets including frequencies of common words, part-of-speeches and part-of-speech trigrams, html tags, punctuation marks etc. using the KI-04 dataset and the naïve bayes technique, she reported 70.2% accuracy.

Kanaris and Stamatatos [9] used character n-grams extracted from both text and structure. Using KI-04 and WebKB datasets and the SVM technique, they achieved accuracy between 90% and 95%.

3. CENTROID-BASED CATEGORIZATION

Most of machine learning techniques consider each document in the training set individually during the training step and consider each one every time a new document will be classified. Only, SVM technique [21] finds a description for each category to distinguish it from other categories.

Like SVM, centroid-based techniques find a category “prototype” that summarizes all documents belonging to the given category, which called category centroid. Several models have been proposed in the literature to calculate centroids. To identify the best model, Cardoso-Cachopo and Oliveira [2] proposed a study to compare Rocchio, average, sum and normalized sum models. They report that normalized sum model outperform all other models.

Generally, the centroid of a particular category \( c_j \) is represented by a vector \( C_j \), which is a combination of the document vectors \( d_i \) belonging (or not) to \( c_j \).

Using normalized sum model, the centroid \( C_j \) is represented by a vector, which is the sum of all the vectors of documents belonging to the category \( c_j \). This is normalized so that it has unitary length. Then, the centroid \( C_j \) is defined as follow:

\[
C_j = \frac{1}{|c_j|} \sum_{d_i \in c_j} d_i
\]

During categorization step, the cosine similarity between each new document vector \( d_i \) and each centroid \( C_j \) is calculated as follow:

\[
sim(d_i, C_j) = \frac{d_i \cdot C_j}{||d_i|| ||C_j||}
\]

The document is assigned to the category having most similarity.

Note that, the time and memory required by centroid-based models are proportional to the number of categories instead of the number of training documents like other machine learning techniques, such as Naive Bayes, K nearest neighbors, decision trees, etc. Also, in centroid-based models you can add more training documents and easily recalculate centroids [18].

4. MY APPROACH

In this paper, I propose a refined and incremental centroid-based approach to classify web pages by genres. My approach is based on the combination of three homogenous classifiers, which uses three heterogeneous sources of information. These sources are the URL, the logical structure (called also internal structure) and the hypertext structure (called also external structure). Each classifier is based on the generation of genre centroids using a training set of web pages. The originality of my approach is the implementation of three new aspects: refining, incrementing and combination.

4.1 Representation

In my approach I performed a pre-processing step to extract the features, which are the URL address, the logical structure and the hypertext structure. The URL is processed as a one line of text. The logical structure is represented by the words between \(<\text{title}>\) and \(</\text{title}>\), and between \(<\text{Hn}>\) and \(</\text{Hn}>\), where \( n = 1, \ldots, 6 \). The usefulness of this kind of structure is studied in [6]. The hypertext structure is represented by the words, which are underlined in all hyperlinks contained in the web page. A study of the usefulness of different HTML tags is presented in [8].

Each feature will be processed to remove special symbols and special characters. For example, for the URL feature, the symbols \( :, ;, /\) and \( .\) are removed because they are founded in all URLs. For logical and hypertext structure, I can find some HTML characters like ‘&amp;’ or ‘&lt;38;’. This kind of characters should be removed.

Stop words should be removed to reduce vocabulary size. These stop words are automatically identified for each feature using frequency threshold technique, which based on Zipf theory [23]. In this theory, a word is considered a stop word if its frequency is less than a minimum threshold and greater than a maximum threshold. The list of stop words depends on each feature. For the

\(^1\)http://www.irit.brighton.ac.uk/~Marina.Santini/
\(^2\)http://www.cs.colostate.edu/~boese/Research/Corpora.html
URL, words that appear more than twice and more than 5 times are considered a stop word. For logical and hypertext structures, words that appear less than 5 times and more than 10 times, are considered a stop word.

The remaining words will be stemmed using the porter stemmer [15]. For each term, I calculated its information gain [22]. To select terms, I sort all terms by descending values of information gain and then select the top n terms, where n is determined through an experimental study.

The selected terms are weighted using the TF-IDF technique [16]. The TF-IDF value increases with the number of times that the term occurs in the document and decreases with the number of times the term occurs in the collection. This technique favors big documents instead of small documents. Recently, Lertnathee and Theramunkong proposed a new approach to ameliorate TF-IDF term weighting. This approach is based on term distributions within a particular class, between classes and within the entire collection [13]. In this paper, I have used term distribution approach with exponent factors $\alpha=0.5$, $\beta=-1$ and $\gamma=-0.5$.

### 4.2 Construction of Centroids

I observed that the training web pages that are far away from its genre centroid tend to reduce the performance of categorization. My hypothesis is that these kinds of web pages are noisy examples and not considered as a useful training examples. So, it’s suitable to be excluded during centroid computation.

In my approach, I first calculate the centroids using all training web pages using the normalized sum formula presented in section 3. Then, I obtain a set of centroids $C = \{C_1, ..., C_j, ..., C_k\}$, where $k$ is the number of genres. Next, I discarded web pages that have a similarity with a centroid less than a predefined threshold $S_0$. For each category $c_p$, I calculate a new set of training web pages $s_j$ as follow:

$$s_j = \{p_i \in C_j \text{ and } \text{sim}(p_i, C_j) \geq S_0\}$$

Where $p_i$ is a web page and sim is the cosine similarity presented in section 3. An experimental study is conducted in the next section to discuss the choice of the appropriate refining threshold value.

The sets of training pages obtained after refining, will be used to recalculate the centroids using the normalized sum formula as follow:

$$S_j = \frac{1}{\|S_j\|^2} \sum_{p_i \in s_j} p_i$$

Finally, the refined centroids are applied to classify new web pages.

Notice, that the complexity of centroids construction is linear in the number of training web pages $m$ and the number of predefined categories $k$, hence, learning time is bounded by $O(km)$.

### 4.3 Web Page Categorization

In my approach, the categorization of new web pages is performed one by one. For each new web page $p$, I calculate its cosine similarity with all centroids. Then, I refine the centroids, which have a similarity with the page $p$, greater or equal than $S_0$. The refining step consists in adding the new page $p$ to the normalized centroid of the corresponding genre and renormalizes the centroid. For this reason, I maintained with each normalized centroid $S_p$, the non-normalized centroid $NS_p$, so that refining the centroid $S_p$ can be performed by the following operations:

$$NS_j = NS_j + p \text{ And then } S_j = \frac{NS_j}{\|NS_j\|}$$

If all similarity values between the new page $p$ and centroids are less than the refining threshold $S_0$, then I discard the new web page $p$ because it considered as a noisy web page. Note that the value of the refining threshold $S_0$ is the same as used to construct centroids.

Classifying a new web page is linear in the number of centroids $k$ and the number of new web pages $n$; hence, running time for classification is bounded by $O(kn)$.

### 4.4 Combination

The aim of combining the outputs of several classifiers is to achieve a performance better than obtained by each classifier individually. Many combining methods were proposed in the literature [12].

In my approach I propose to combine three classifiers. These classifiers are based on the same learning algorithm presented in the previous section, but they use heterogeneous sources of information, which are the URL, the logical structure and the hypertext structure. In previous paper, I have used many aggregation operators to combine different HTML tags [7]. In this paper, I have used decision templates (DT) method [11]. The principal of this method is explained below.

#### 4.4.1 Decision Templates

Let $E = \{E_1, ..., E_L\}$ be a set of classifiers. Each of these classifiers produces the output $E_i(x) = [d_{i1}(x), ..., d_{ic}(x)]$ where $d_{ij}(x)$ is the membership degree given by the classifier $E_i$ that $x$ belong to the class $j$. The outputs of all classifiers can be represented by a decision profile DP matrix as follow:

$$DP(x) = \begin{bmatrix} d_{11}(x) & ... & d_{1c}(x) \\ ... & ... & ... \\ d_{L1}(x) & ... & d_{Lc}(x) \end{bmatrix}$$

Using the training set $Z = \{Z_1, ..., Z_N\}$, I compute the fuzzy template $F$ of each class $i$, which represented by a L*c matrix $F_i = \{f_i(k, s)\}$. The element $f_i(k, s)$ is calculated as follow:

$$f_i(k, s) = \frac{\sum_{j=1}^{N} \text{Ind}(Z_{j},i) \cdot d_{ks}(Z_{j})}{\sum_{j=1}^{N} \text{Ind}(Z_{j},i)}$$
Where $\text{Ind}(Z_j, i)$ is an indicator function with value 1 if $Z_j$ comes from class $I$ and 0 otherwise.

At this stage, the ranking of classes can be achieved by aggregating the columns of DP using fixed rules (minimum, maximum, product, average, etc.). Another method calculates a soft class label vector with components expressing similarity $S$ between the decision template $DP$ and the fuzzy template $F$. The final classification $CLV$ is defined as follows:

$$CLV(x)_{ci} = \mu_i(x) = \min_{1 \leq i \leq L} \left( \frac{1}{c} \sum_{k = 1}^{c} \sum_{s = 1}^{s} (f_i(k, s) - d_k(x))^2 \right)$$

5. EXPERIMENTS AND RESULTS

I this section, I presented many experiments to evaluate my approach. In the first paragraph I describe the datasets used in my experiments. Next, I describe our experimental setup and finally I discuss the obtained results.

5.1 Datasets

Experiments should be conducted using datasets of HTML web pages grouped by genres. To evaluate URL classifier, I should know the URL address of the web page. According to these conditions, I can use only two datasets, which are KI-04 and WebKB datasets.

The KI-04 dataset was built following a palette of eight genres. It includes 1205 web pages. These genres were suggested by a user study on genre usefulness [14] (see Table 1).

<table>
<thead>
<tr>
<th>Genre</th>
<th># Of web pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Article</td>
<td>127</td>
</tr>
<tr>
<td>Download</td>
<td>151</td>
</tr>
<tr>
<td>Link collection</td>
<td>205</td>
</tr>
<tr>
<td>Private portrayal</td>
<td>126</td>
</tr>
<tr>
<td>Non private portrayal</td>
<td>163</td>
</tr>
<tr>
<td>Discussion</td>
<td>127</td>
</tr>
<tr>
<td>Help</td>
<td>139</td>
</tr>
<tr>
<td>Shop</td>
<td>167</td>
</tr>
</tbody>
</table>

The WebKB dataset [3] comes from the WebKB project at Carnegie Mellon University. It is composed of 8282 web page gathered from computer science department web sites of four American universities. These web pages are issued from seven categories, but I used only six categories (course, department, faculty, project, staff and student) as usually done. After discarding the category other and empty web pages, I obtained only 4249 web page. (see Table 2).

<table>
<thead>
<tr>
<th>Genre</th>
<th># Of web pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student</td>
<td>1541</td>
</tr>
<tr>
<td>Faculty</td>
<td>1063</td>
</tr>
<tr>
<td>Staff</td>
<td>126</td>
</tr>
<tr>
<td>Department</td>
<td>170</td>
</tr>
<tr>
<td>Project</td>
<td>474</td>
</tr>
<tr>
<td>Course</td>
<td>875</td>
</tr>
</tbody>
</table>

5.2 Experimental Setup

In my approach, a web page is assigned to the genre having most similarity. The suitable performance measure to this kind of classifier is the accuracy [18]. Since, my datasets are unbalanced, I used the micro-average accuracy. In my experiments, I used 5*2 cross-validation methodology, which consists in splitting the dataset into two blocs. One bloc is used for training and the other for testing. This process is repeated 5 times. For each time, I calculated the micro-average accuracy. Then, global accuracy is the average accuracy over the 5 times.

5.3 Results

To evaluate my approach, I conducted many experiments. The aim of these experiments is to measure the effect of vocabulary size, refining, incrementing and combination on the global micro-average accuracy. I proposed also a comparison with other works on genre categorization and other machine learning techniques.

5.3.1 Effect of vocabulary size

Experiments presented in this paper are conducted using different vocabulary’ sizes according to the dataset and the feature used. The results are showed in Figure 1. These results are obtained when the number of terms is varied between 5 and 3000.

The curves presented in the Figure 1 show that micro-average accuracy depends on the number of terms. I observe that the number of terms, which provides better results, is between 30 and 200.

5.3.2 Effect of refining

To measure the effect of refining on genre categorization, I varied the refining threshold between 0 and 1 by step of 0.1. Zero value means that is no refining. The obtained results for all features and datasets are presented in the following figure.

3http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/
As illustrate in the Figure 2, the value of refining threshold affects the micro-average accuracy of genre categorization. For KI-04 dataset, I remark that the best results are obtained for the values 0.4, 0.5 and 0.6 as refining thresholds for respectively the URL, the logical and the hypertext structures. For the WebKB dataset, the values of refining threshold are less than those of KI-04 dataset. These values are 0, 0.1 and 0.2 for respectively the URL, the logical and the hypertext structures. According to these results I notice that in the case of noisy web pages like those contained in KI-04 dataset, the refining is very useful. On the other hand, for noiseless corpus like WebKB dataset, the refining is useless. So, I concluded that refining work very well for noisy datasets and affect the global performance of genre categorization.

5.3.3 Effect of incrementing

To Measure the effect of incrementing I performed an experiment using the best number of terms identified in the first experiment and the best refining thresholds identified in the second experiment. In this experiment I varied the proportion of testing web pages on each feature between 10% and 90% by step of 10%. For example, 80% means that I have used 20% of web pages for training and the remaining (80%) for testing. The results are illustrated in the Figure 3.

The curves presented in the Figure 3 shows that micro-average accuracy depends on the proportion of testing web pages and the feature used. According to these curves, I have concluded that genre categorization using the KI-04 dataset needs more testing web pages that those performed using the WebKB dataset. This result is explained according to the number of noisy web pages in each corpus as discussed in the previous experiment. Indeed, the best proportions of testing web pages for KI-04 dataset are 80%, 70% and 70 % for respectively the URL, the logical and the hypertext structures. For WebKB dataset, the best proportions of testing web pages are 30%, 20% and 30 % for respectively the URL, the logical and the hypertext structures.

5.3.4 Effect of combination

In this experiment I measure the effect of each classifier on the final micro-average accuracy. The results of combination are presented in the Figure 4. These results show that the combination of classifiers using DT technique yields more than 95% of accuracy. This result outperforms those obtained using each feature separately.

### Table 3. Micro-average accuracy for both KI-04 and WebKB and for web page genre categorization

<table>
<thead>
<tr>
<th>Author</th>
<th>KI-04</th>
<th>WebKB</th>
</tr>
</thead>
<tbody>
<tr>
<td>[14]</td>
<td>0.70</td>
<td>-</td>
</tr>
<tr>
<td>[1]</td>
<td>0.75</td>
<td>0.80</td>
</tr>
<tr>
<td>[9]</td>
<td>0.84</td>
<td>-</td>
</tr>
<tr>
<td>[17]</td>
<td>0.70</td>
<td>-</td>
</tr>
<tr>
<td>My approach</td>
<td>0.96</td>
<td>0.98</td>
</tr>
</tbody>
</table>

5.4 Comparison

In this section, I conducted experiments to compare my approach with other known works on genre categorization of web pages and with other used machine learning techniques.
5.4.2 Comparison with other machine learning techniques

In my experimentation I compare my approach with other categorization techniques implemented in the program Rainbow⁴. Among these techniques I have used Rocchio, Naïve bayes (NB), K Nearest Neighbors (KNN) with K=30, SVM with Fisher kernel and TreeNode because they are widely used in genre categorization of documents. Results are presented in Tables 4 and 5. These results show that my approach outperform all other techniques.

Table 4. KI-04: Micro-average accuracy for each machine learning technique and for each feature

<table>
<thead>
<tr>
<th></th>
<th>URL</th>
<th>Logical Structure</th>
<th>Hypertext Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>My approach</td>
<td>0.81</td>
<td>0.85</td>
<td>0.83</td>
</tr>
<tr>
<td>SVM</td>
<td>0.79</td>
<td>0.85</td>
<td>0.80</td>
</tr>
<tr>
<td>Rocchio</td>
<td>0.77</td>
<td>0.82</td>
<td>0.82</td>
</tr>
<tr>
<td>NB</td>
<td>0.72</td>
<td>0.80</td>
<td>0.81</td>
</tr>
<tr>
<td>KNN</td>
<td>0.67</td>
<td>0.66</td>
<td>0.80</td>
</tr>
<tr>
<td>TreeNode</td>
<td>0.63</td>
<td>0.62</td>
<td>0.65</td>
</tr>
</tbody>
</table>

Table 5. WebKB: Micro-average accuracy for each machine learning technique and for each feature

<table>
<thead>
<tr>
<th></th>
<th>URL</th>
<th>Logical Structure</th>
<th>Hypertext Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>My approach</td>
<td>0.86</td>
<td>0.87</td>
<td>0.84</td>
</tr>
<tr>
<td>SVM</td>
<td>0.84</td>
<td>0.87</td>
<td>0.81</td>
</tr>
<tr>
<td>Rocchio</td>
<td>0.82</td>
<td>0.86</td>
<td>0.82</td>
</tr>
<tr>
<td>NB</td>
<td>0.81</td>
<td>0.81</td>
<td>0.81</td>
</tr>
<tr>
<td>KNN</td>
<td>0.70</td>
<td>0.74</td>
<td>0.76</td>
</tr>
<tr>
<td>TreeNode</td>
<td>0.62</td>
<td>0.64</td>
<td>0.62</td>
</tr>
</tbody>
</table>

To show that obtained results are really meaningful and not due to chance, I used the 5*2 cross-validation t-test [5]. The results are illustrated in the following table.

5.4.3 Comparing train and test times

Besides the categorization performance that categorization techniques yields, another important aspect to consider when comparing classification techniques is the time that they require to execute. Time is a very important aspect for comparison, especially when you wish to integrate my approach in a search engine. Figures 5, 6 and 7 shows a comparison of the time that each classification technique needs to execute, in both training and classification phases for each dataset and for each feature.

According to this table, I conclude that SVM approach achieve similar results than my approach, especially for WebKB dataset. Rocchio provide the nearest results among the other approaches.

The symbols used in this table are defined as follow:

- ≈ Indicates no significant differences
- < Indicates that the row approach achieves a significantly lower measurement then our approach with 0.05 as a significance level
- << Indicates that the row approach achieves a significantly lower measurement then our approach with 0.01 as a significance level
- <<< Indicates that the row approach achieves a significantly lower measurement then our approach with 0.005 as a significance level

⁴ http://www.cs.cmu.edu/~mccallum/bow/rainbow/
The X axis represents the time spent during the training phase and Y axis represents the time spent during the testing phase, both in seconds. By looking at the X and Y axis, I notice that my approach is the fastest in both training and testing phases. This can be explained by the fact that the time spent is proportional to the number of categories instead of the number of training web pages as for the other approaches. Among the other approaches, Rocchio and Naïve bayes are the fastest. Decision tree is the slowest classifier.

6. CONCLUSION AND FUTURE WORK
In this paper, I proposed a new approach for genre categorization of web pages. My approach uses three new features, namely, the URL address, logical and hypertext structures. Moreover, my approach implements three new aspects, which not explored in previous studies on genre categorization. These aspects are refining and incrementing. Also I have conducted many experiments to measure the usefulness of each aspect in genre categorization. The comparison with other approaches shows that my approach is the fastest and outperforms many known categorization techniques. In the future, I hope to integrate my approach in a web search engine.

7. REFERENCES
Proceeding of the 19th IEEE Int. Conf. on Tools with Artificial Intelligence. 2007.