Chaker Jebari

A New Centroid-based Approach for Genre Categorization of Web Pages

In this paper we propose a new centroid-based approach for genre categorization of web pages. Our approach constructs genre centroids using a set of genre-labeled web pages, called training web pages. The obtained centroids will be used to classify new web pages. The aim of our approach is to provide a flexible, incremental, refined and combined categorization, which is more suitable for automatic web genre identification. Our approach is flexible because it assigns a web page to all predefined genres with a confidence score; it is incremental because it classifies web pages one by one; it is refined because each web page either refines the centroids or is discarded as noisy page; finally, our approach combines three different feature sets, i.e. URL addresses, logical structure and hypertext structure. The experiments conducted on two known corpora show that our approach is very fast and outperforms other approaches.

1 Introduction

As the World Wide Web continues to grow exponentially, web page categorization becomes increasingly important in web searching. Web page categorization, called also web page classification, assigns a web page to one or more predefined categories. According to the type of category, categorization can be divided into sub-problems: topic categorization, sentiment categorization, genre categorization, and so on.

Recently, more attention has been given to automatic genre identification of web pages because it can be used to improve the quality of web search results — see, for example, all the articles in this journal and Mehler et al. (2009).

However, although potentially useful, the concept of "genre" is difficult to define and genre definitions abound. Generally speaking, a genre is a category of artistic, musical, or literary composition characterized by a particular style, form, or content [1], but more specialized characterizations have been proposed. For instance, Kessler et al. (1997) defined a genre as a bundle of facets, focusing on different textual properties such as brow, narrative and genre. According to Shepherd and Watters (1998), while non-digital genre is defined by the tuple <content, form>, the genre of web pages (or "cybergenre") is characterized by the triple <content, form, functionality>, where the functionality attribute accounts for the interaction between the user and the web page. Rauber and Müller-Kögler (2001) defined the genre as a group of documents that share the same stylistic properties. In their experiment with digital libraries, documents of

[1] For example, see Merriam-Webster Online Dictionary `http://www.m-w.com`

the same genre are rendered with the same color. According to Finn (2002), genre is orthogonal to topic, and relates to polarities such as subjectivity/objectivity and positivity/negativity. For Boese (2005) a genre is characterized by the same style, form and content.

In this article, the word "genre" is loosely defined as a textual category that can be more or less related to the topic or content of a web pages. For this reason, we use two different collections. One created by genre researchers for whom the concept of genre is independent from topic (the KI-04 corpus, see Section 4); the other including a number of academic categories (the WebKB collection, see Section 4).

In order to comply with our view of genre, our approach is flexible, incremental, refinable and combines different feature sets. We devised it to be fast, so that in the future it can be applied to web search engines.

Currently, search engines use keywords to classify web pages. Returned web pages are ranked and displayed to the user, who is often not satisfied with the result. For example, searching for the keyword "machine learning" will provide a list of web pages containing the words "machine" and "learning". These pages are from different genres. Therefore, web page genre categorization could be used to improve the retrieval quality of search engines (e.g. see Meyer Zu Eissen (2007)). For instance, a classifier could be trained on existing web directories and be applied to new web pages. At query time the user could be asked to specify one or more desired genres so that the search engine would returns a list of genres under which the pages would fall.

However, a web page is a complex object that includes heterogeneous elements with different communicative purposes. Generally, a web page is composed of different sections organized in the form of headings and links. These sections belong to different genres. Graphical elements (search buttons, images, menus, forms, etc.) and text types, sizes and colors are used to mark sections in web pages. Our approach assigns a web page to all predefined genres with different confidence scores, which represent the similarity between the web page and the centroid of each genre.

It is worth noting that web genres evolve over time because of the continuous modification of the content and purpose of web pages. Simply put, web genre evolution consists in updating old genres and creating new ones. In our approach we focus on the adjustment of old genres. Since automatic genre identification of web pages requires continuous learning, because web pages are often updated, we propose an incremental approach (see Section 3).

Additionally, the Word Wide Web is an open environment, where the user can add a new page, modify the content of actual web page, delete a web page and so on. For this reason, the web is instable and contains many noisy web pages. Taking such web pages into account decreases the accuracy of genre classification (e.g. see Shepherd et al. (2004)). In our approach we propose a refined genre classification of web pages to discard noisy web pages. A web page is considered noisy where its similarities to all genre centroids are below a predetermined threshold.

As mentioned above, a web page is not only a text but contains many HTML tags. The information delimited by these tags is very useful for genre categorization. These

information sources are heterogeneous because they have different representation structures that should be combined to increase the performance of genre classification of web pages.

In summary, the aim of our approach is to provide a flexible, incremental, refined and combined categorization, which is more suitable for automatic web genre identification. Our approach is flexible because it assigns a web page to all predefined genres with a confidence score; it is incremental because it classifies web pages one by one; it is refined because each web page either refines the centroids or is discarded as noisy page; finally, our approach combines three different feature sets, i.e. URL addresses, logical structure and hypertext structure.

This article is organized as follows: in Section 2 we summarize previous work on genre categorization of web pages; in Section 3 we explains our approach; in Section 4 we briefly describe the corpora used in our experiments; Section 5 presents our experimental results; Section 6 presents a comparative study; finally, in Section 7 we present some conclusions as well as our future work.

2 Related Work

Previous work on automatic genre identification is reviewed by focusing on features, classification algorithms and genre corpora.

Features Many types of features have been proposed for automatic genre categorization. In the following paragraphs, the most important features are listed.

Kessler et al. (1997) used four types of features to classify part of the Brown corpus[2] by multiple facets (i.e. brow, narrative and genre). The first type is represented by structural features, which include counts of functional words, sentences, etc. The second type relies on lexical features, which include the presence of specific words or symbols. The third kind of features are character level features, such as punctuation marks. The fourh kind of features is based on derivative features, which are derived from character level and lexical features. These four features sets can be divided into two coarser types: structural features and surface features.

Karlgren (1999) used twenty features including frequencies of functional words and Parts-of-Speech (POSs). He also used text statistics, e.g. counts of characters, words, number of words per sentence, etc.

Stamatatos et al. (2000) identified genre using the most English common words. They used the fifty most frequent words on the BNC corpus[3] and the eight most frequent punctuation marks (period, comma, colon, semicolon, quotes, parenthesis, question mark and hyphen).

Dewdney et al. (2001) adopted two features sets: BOW (Bag Of Words) and presentation features. Presentation features amounted to 89 features including layout features,

[2] http://en.wikipedia.org/wiki/Brown_Corpus

[3] http://www.natcorp.ox.ac.uk

linguistic features, verb tenses, etc. Finn and Kushmerick (2003) used a total of 152 features to differentiate between subjective vs objective news articles and positive vs negative movie reviews. Most of these features were represented by the frequencies of genre-specific words. Meyer Zu Eissen and Stein (2004) used different kinds of features including presentation features (i.e. HTML tag frequencies), classes of words (names, dates, etc.), frequencies of punctuation marks and POS tags. Kennedy and Shepherd (2005) used a feature set including features about content (e.g. common words, met tags), about the form (e.g. number of images) and about the functionality (e.g. number of links, JavaScripts). Boese and Howe (2005) used different kind of features, which can be grouped into three classes, namely stylistic features, form features and content features. More recently, Santini (2007) and Lim et al. (2005) tried to exploit all previously used features. Additionally, Lim et al. (2005) used the URL as new feature and Kanaris and Stamatatos (2007) used character n-grams extracted from both text and structure. Mehler et al. (2007) studied the usefulness of logical document structure in text type classification. They adopted two approaches, which are the Quantitative Structure Analysis (QSA) and the Document Object Model Tree Kernel (DomTK). They conducted experiments to stress the usefulness of structure in document type recognition and compared the QSA approach against the DomTK approach.

Machine Learning Techniques Once a set of features has been obtained, it is necessary to choose a categorization algorithm. Most genre categorization algorithms are based on machine learning (cf. Mitchell (1997)) techniques . Among these techniques, we briefly explain Naïve Bayes, k-Nearest Neighbor, Decision trees and Support Vector Machine techniques because they have been widely used in automatic genre identification.

Naïve Bayes is a simple probability algorithm that determines the probability of a document to belong to a particular genre. Naïve Bayes is a very fast learning algorithm, which is robust to irrelevant features. It needs reduced storage space and can handle missing values. However, since the weights are the same for all features, performance can be degraded by having many irrelevant features. This technique has been implemented by Argamon et al. (1998); Dewdney et al. (2001); Santini (2007).[4]

The k-Nearest Neighbor (k-NN) algorithm groups documents within a vector space. The Term Frequency Inverse Document Frequency (*tfidf*) is usually employed to represent documents. The similarity between documents is computed with Euclidean or cosine measures. New documents are classified with the same genre as the nearest neighbor. The K represents how many neighbors should be analyzed. K-Nearest Neighbor is used only by Lim et al. (2005).

Decision trees are a popular technique used by Argamon et al. (1998), Dewdney et al. (2001) and Finn (2002). Interestingly, Karlgren (1999) applied a combination of decision trees and Nearest Neighbor. He calculated textual features for each document and categorized them into a hierarchy of clusters based on C4.5 *if-then* rules. The

[4]Santini (2007) tried out also Naïve Bayes with different weights.

labels for genres were then decided using Nearest Neighbor assignments and cluster centroids.

Support Vector Machine is a powerful learning method introduced by Vapnik (1995) and successfully applied to text categorization by Joachims (1998). SVM is based on Structural Risk Maximization theory, which aims to minimize the generalization error instead of relying on the empirical error on training data alone. The Support Vector Machine technique has been used in genre categorization by many authors (e.g. Kanaris and Stamatatos 2007; Dewdney et al. 2001; Meyer Zu Eissen and Stein 2004; Santini 2007).

Corpora and Evaluation To date, web genre benchmarks built with principled and shared criteria are still missing (cf. Santini and Sharoff in this issue). This means that the performance of a genre categorization system depends on the specific corpora being classified. For instance, Kessler et al. (1997) used a corpus of 499 texts from Brown Corpus belonging to six diverse genres (reportage, scientific and technical, fiction, etc). They report 0.61 and 0.75 accuracies for logistic regression and neural network classifiers, respectively. Dewdney et al. (2001) used a corpus of 9705 texts belonging to seven diverse genres (advertisements, bulletin, boards, radio news, etc.). They achieved 0.83, 0.88 and 0.92 accuracies for Naïve Bayes, C4.5 and SVM, respectively. Meyer Zu Eissen and Stein (2004) compiled the KI-04 corpus. In their first experiment, they used 800 web pages (100 web pages for each of the eight genres included in the corpus) and applied discriminant analysis. They achieved an accuracy of 0.7. Boese and Howe (2005) used the WebKB corpus to study the effect of web genre evolution. Based on logistic regression classifier, they reported an accuracy of 0.8. Kanaris and Stamatatos (2007) used the KI-04 corpus and the SVM classifier. They obtained accuracy between 0.90 and 0.95. Santini (2007) used SVM to classify the KI-04 corpus. She reported an accuracy of about 0.7. Mehler et al. (2007) used SVM classifiers and a German newspaper corpus that contains 31,250 texts distributed over 31 genres or types. Their experiments provided F1=0.78 for QSA and F1=0.57 for DomTK.

3 Proposed Approach

The aim of our approach is to classify web pages by genre based on three different feature sets, namely URL addresses, logical structure and hypertext structure. The proposed approach is based on the construction of genre centroids using a set of genre-labeled web pages. Each new web page is assigned to all genres with different confidence scores, which represent the similarity between the web page and the centroid of each genre.

In the subsection 3.1, we explain our feature extraction process. The representation of features, the construction of centroids, the categorization of new web pages and the combination of classifiers are described in subsections 3.2, 3.3, 3.4 and 3.5, respectively.

3.1 Feature Extraction

In our approach, we used three different types of features, which are the URL addresses, the logical structure and the hypertext structure.

The URL is encoded as a text line, which contains genre-specific words. For example, the presence of "FAQ" and "CV" in the file name is a reliable hint of the membership of a web page to the FAQ and CV genres, respectively.

The logical and hypertext structures of a web page are encoded into the HTML tags used in the web page. The logical structure is represented by the text between $< title >$ and $< /title >$ tags and the text between $< Hn >$ and $< /Hn >$ tags $(n = 1, \ldots, 6)$, while the hypertext structure is represented by the text included in the anchors (between $< A \ldots >$ and $< /A \ldots >$ tags).

To quantify the contextual and structural information, we used the bag-of-words approach – already employed by (Dewdney et al., 2001) for automatic genre identification) – which relies on all words without ordering.

3.2 Representation

Web page representation is performed through three main steps, which are pre-processing, term weighting, and normalization.

Pre-processing Pre-processing is a basic step in document categorization. In our approach, the aim of this step is summarized in the following points:

- Tokenize text into words.
- Remove numbers, non-letter characters and special characters.
- Remove stop words, which are automatically identified using the Luhn Law (Luhn, 1958).
- Use the information gain to reduce the number of obtained terms (Yang and Pedersen, 1997).
- Stem selected terms using the Porter stemmer (Porter, 1980).

Term weighting In our work, web pages are represented using the vector space model. We use three different vectors representing the URLs, the logical structure and the hypertext structure. For each feature set, a web page is represented by a vector p_j of terms. Each term t_i is weighted using the $tfidf$ weighting technique (Salton and Buckley, 1988).

With this technique, the w_{ij} of a term t_i in a web page p_j increases with the number of times that the term t_i occurs in the page p_j and decreases with the number of times the term t_i occurs in the collection. This means that the importance of a term in a page is proportional to the number of times that the term appears in the page, while

the importance of the term is inversely proportional to the number of times that the term appears in the entire collection. Formally, this reasoning is defined as follows:

$$w_{ij} = \frac{tf_{ij}}{max(tf_{1j})} \times \log(\frac{|D|}{n_{t_i}}) \quad (1)$$

where tf_{ij} is the number of times that term t_i appears in web page p_j, $|D|$ is the total number of pages in the collection, and n_{t_i} is the number of pages where term t_i appears.

Normalization The $tfidf$ technique favors large documents and penilizes short documents. To deal with this problem, Lertnattee and Theeramunkong (2004) proposed a normalization technique, called TD, which based on term distribution within a particular class and within a collection of documents.
The term distribution is based on three different factors. These factors depend on the average frequency of the term t_i in all pages of genre g_k. This average, denoted by $\overline{tf_{ik}}$ is defined as follows:

$$\overline{tf_{ik}} = \frac{\sum_{p_j \in g_k} tf_{ijk}}{|D_{g_k}|} \quad (2)$$

where D_{g_k} represents the set of web pages that belongs to genre g_k, and tf_{ijk} is the frequency of term t_i in page p_j of genre g_k.

The normalization factors are the interclass standard deviation ($icsd$), the class standard deviation (csd) and the standard deviation (sd).
The inter-class standard deviation promotes a term that exists in almost all genres but its frequencies for those genres are quite different. For a term t_i, this factor is defined as follows:

$$icsd_i = \sqrt{\frac{\sum_k [\overline{tf_{ik}} - \frac{\sum_k \overline{tf_{ik}}}{|G|}]^2}{|G|}} \quad (3)$$

The class standard deviation of a term t_i in a genre g_k depends on the different frequencies of the term in the pages of that genre, and varies from genre to genre. This factor is defined as follows:

$$csd_{ik} = \sqrt{\frac{\sum_{d_j \in g_k} [tf_{ijk} - \overline{tf_{ik}}]^2}{|g_k|}} \tag{4}$$

The standard deviation of a term t_i depends on the frequency of that term in the pages in the collection and is independent of genres. It is defined as follows:

$$sd_i = \sqrt{\frac{\sum_k \sum_{d_j \in g_k} [tf_{ijk} - \frac{\sum_k \sum_{d_j \in g_k} tf_{ijk}}{\sum_k |g_k|}]^2}{\sum_k |g_k|}} \tag{5}$$

Using the $tfidf$ weighting technique and term distributions for normalization, the weight of term t_i for page p_j in genre g_k is defined as follows:

$$wtd_{ijk} = w_{ij} \times sd_i^{\alpha} \times icsd_i^{\beta} \times csd_{ik}^{\gamma} \tag{6}$$

where α, β and γ are the normalization parameters, which were used to adjust the relative weight of each factor and to indicate whether they were used as a multiplier or as a divisor for the term's $tfidf$ weight, w_{ij}. An experimental study is conducted in section 4 to identify the appropriate values of these parameters.

3.3 Construction of genre centroids

The centroid of a particular genre g_j is represented by a vector G_j. This centroid is the combination of the vectors p_j belonging (or not) to that genre. Several ways were proposed to calculate this centroid. The most used one is the normalized sum, defined as follows:

$$G_j = \frac{1}{\|g_j\|} \cdot \sum_{p_i \in g_j} p_i \tag{7}$$

We observed that web pages that are far away from its genre centroid tend to negatively affect the performance of categorization. Our hypothesis is that these web pages increase web search noise and, consequently, they cannot be considered as useful training pages. For this reason, they should be excluded during centroid computation.
Assume that you have obtained a set of genre centroids $G = \{G_1, \ldots, G_j, \ldots, G_{|G|}\}$, where $|G|$ is the number of genres. In our approach, we discarded web pages that have a similarity with the genre centroid below a predefined threshold s_0.
For each genre g_j, we calculate a new set of training or labeled web pages s_j as follows:

$$s_j = \{p_i \in g_j \backslash sim(p_i, G_j) \geq s_0\} \tag{8}$$

where p_i is a web page and sim is the cosine similarity between the page p_i and the genre centroid G_j defined as follows:

$$sim(p_i, G_j) = \frac{p_i \cdot G_j}{\|p_i\| \cdot \|G_j\|} \tag{9}$$

The sets of training pages obtained after refining will be used to recalculate the genre centroids using the normalized sum presented in equation 7 as follows:

$$S_j = \frac{1}{\|s_j\|} \cdot \sum_{p_i \in s_j} p_i \tag{10}$$

Finally, the refined centroids will be applied to classify new web pages. Note that the complexity of centroid construction is linear to the number of labeled web pages m and to the number of predefined genres $|G|$. Hence, learning time depends on $O(m|G|)$.

In order to choose the appropriate threshold, we carried out the experimental study described in the next subsection.

3.4 Categorization of New Web Pages

In our approach, the categorization of new web pages is performed incrementally. For each new web page p, we calculated its cosine similarity with all genre centroids. Then, we refined the centroids that have a similarity with the page p greater or equal than S_0.

The refining process is performed as follows:

$$NS_i = NS_i + p, S_i = \frac{NS_i}{\|NS_i\|} \tag{11}$$

where NS_i is the non-normalized centroid of the genre g_i and represents the norm of the vector. NS_i is calculated as follows:

$$\|NS_i\| = \sqrt{\sum_{k \in NS_i} wtd_{ijk}^2} \tag{12}$$

The complexity of web page classification is linear to the number of genres $|G|$ and to the number of unlabeled web pages. Therefore, the running time for classification depends on $O(n|G|)$.

3.5 Combination

The basic idea behind the combination of different classifier methods is to create a more accurate classifier via some combination of the outputs of the contributing classifiers. In our approach, the idea is based on the intuition that the combination of homogenous classifiers using heterogeneous features might improve the final result.

OWA operators OWA (Ordered Weighting Average) operators were first introduced in (Yager, 1988). Generally speaking, a mapping $F : [0,1]^n \rightarrow [0,1]$ is called an OWA operator of dimension n if it is associated with a weighting vector $W = [w_1, \ldots, w_i, \ldots, w_n]$, such that $w_i \in [0,1]$, $\sum_i w_i = 1$ and $F(a_1, \ldots, a_n) = \sum_i w_i b_i$, where b_i is the ith largest element in the collection $a_1, \ldots, a_n$. Yager (1988) suggested two methods for identifying weights. The first approach uses learning techniques. The second one firstly gives some semantics to the weights, then based on this semantics, the values for weights are provided.
In the experiments described in this article, we used the second method based on fuzzy linguistic quantifiers for the weights. According to Zadeh (1983), there are two types of quantifiers: absolute and relative. Here, we used relative quantifiers typified by terms such as "as most", "as least half", etc. A relative quantifier Q is defined as a mapping $Q : [0,1] \rightarrow [0,1]$, verifying $Q(0) = 0$, there exists $r \in [0,1]$ such that $Q(r) = 1$ and Q is a non-decreasing function. Herrera and Verdegay (1996) defined a quantifier function as follows:

$$Q(r) = \begin{cases} 0, & if r < a; \\ \frac{r-a}{b-a}, & if r \in [a,b]; \\ 1, & if r > b. \end{cases} \quad (13)$$

where a, $b \in [0,1]$ are two parameters. Yager (1988) computed the weight $w_i (i = 1, \ldots, n)$ as follows:

$$w_i = Q(\frac{1}{n}) - Q(\frac{i-1}{n}) \quad (14)$$

where n is set to 3 because we have three classifiers, named URL, logical and hypertext classifiers. Depending on the values of the parameters a and b, we used the following function operators:

- **Minimum:** Represented by the quantifier "For all" and the function:

$$Q(r) = \begin{cases} 0, & r \neq 1; \\ 1, & r = 1. \end{cases} \quad (15)$$

- **Maximum:** Represented by the quantifier "There exists" and the function:

$$Q(r) = \begin{cases} 0, & r < 1/3; \\ 1, & r \geq 1. \end{cases} \quad (16)$$

- **Median:** Represented by the quantifier "At least one" and the function:

$$Q(r) = \begin{cases} 0, & r < 0; \\ r, & 0 \leq r \leq 1; \\ 1, & r > 1. \end{cases} \quad (17)$$

- **Vote1:** Represented by the quantifier "At least half" and the function:

$$Q(r) = \begin{cases} 0, & r < 0; \\ 2r, & 0 \leq r \leq 0.5; \\ 1, & r > 0.5. \end{cases} \quad (18)$$

- **Vote2:** Represented by the quantifier "As possible" and the function:

$$Q(r) = \begin{cases} 0, & r < 0.5; \\ 2r - 1, & 0.5 \leq r \leq 1; \\ 1, & r > 0.5. \end{cases} \quad (19)$$

Decision templates Decision templates were proposed by Kuncheva et al. (2001).

Let E_1, E_2 and E_3 be the URL, the logical and the hypertext classifiers. Each of these classifiers produces the output $E_i(p) = [d_{i1}(p), \ldots, d_{i|G|}(p)]$ where $d_{ij}(p)$ is the membership degree given by the classifier E_i that a web page p belong to the genre j. The outputs of all classifiers can be represented by a decision profile DP matrix as follows:

$$DP(p) = \begin{pmatrix} d_{11}(p) & \ldots & d_{1|G|}(p) \\ d_{21}(p) & \ldots & d_{2|G|}(p) \\ d_{31}(p) & \ldots & d_{3|G|}(p) \end{pmatrix} \quad (20)$$

Using the training set $Z = Z_1, \ldots, Z_N$, we computed the fuzzy template F of each genre i, which is represented by a $3 \times |G|$ matrix $F_i = f_i(k, s)$. The element $f_i(k, s)$ is calculated as follows:

$$f_i(k,s) = \frac{\sum_{j=1}^{N} Ind(Z_j,i) \cdot d_{ks}(Z_j)}{\sum_{j=1}^{N} Ind(Z_j,i)} \quad (21)$$

where $Ind(Z_j,i)$ is an indicator function with value 1 if Z_j comes from genre i and 0 otherwise. At this stage, the ranking of genres 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:

$$CLV(p) = [\mu_1(p), \ldots, \mu_i(p), \ldots, \mu_{|G|}(p)] \quad (22)$$

where $\mu_i(p)$ is the similarity $S(F_i, DP(p))$ between the fuzzy template F_i of the genre i and the decision profile $DP(p)$ of the web page p. This similarity is calculated using the Euclidean measure as follows:

$$\mu_i(p) = S(F_i, DP(p)) = 1 - \frac{1}{3 \times |G|} \cdot \sum_{k=1}^{3} \sum_{s=1}^{|G|} (f_i(k,s) - d_{ks}(p))^2 \quad (23)$$

4 Corpora

In our experiment, we used the KI-04 corpus and WebKB collection[5]. These corpora are composed of English web pages. Each web page is associated with a specific source URL address, and belongs to a single genre class.

- **KI-04** corpus was compiled by Meyer Zu Eissen and Stein (Meyer Zu Eissen and Stein, 2004). It is composed of 1205 HTML web pages, which are divided into eight genres (see Table 1).

- **WebKB** corpus was created at Carnegie-Mellon University during the WebKB project (Craven et al., 1998). This corpus contains 4249 HTML web pages from four different universities. The corpus comprises six genres (see Table 2).

[5]Both these corpora can be reached through the WebGenreWiki `http://http://www.webgenrewiki.org/index.php5/Genre_Collection_Repository/`

Table 1: Composition of the KI-04 corpus

Genre	♯ of web pages
Article	127
Download	151
Link collection	205
Private portrayal	126
Non-private portrayal	163
Discussion	127
FAQ	139
Shop	167

Table 2: Composition of the WebKB corpus

Genre	♯ of web pages
Student	1541
Faculty	1063
Staff	126
Department	170
Project	474
Course	875

5 Evaluation

In this section, we describe our evaluation within the $FRICC$ framework. $FRICC$ is the abbreviation of Flexible, Refined and Incremental Centroid-based Classifier. The aims of the evaluation process can be summarized as follows:

- Identify the best proportions of labeled and unlabeled web pages to achieve the best performance.
- Identify the appropriate number of terms to obtain the best performance.
- Identify the appropriate values of normalization parameters.
- Identify the best thresholds.
- Identify the best combination techniques.

For multiclass corpora, it is suitable to use the break-even-point (BEP), which is defined in terms of the standard measures of precision and recall (Joachims, 1997). Precision P is the proportion of true document-category assignments among all assignments predicted by the classifier. Recall R is the proportion of true document-category assignments that were also predicted by the classifier. Formally, the BEP statistic finds the point where precision and recall are equal. Since this is hard to achieve in

practice, a common approach is to use the arithmetic mean of recall and precision as an approximation, i.e. $BEP = (P + R)/2$. Since our corpora are unbalanced, we used the micro-averaged BEP computed by first summing the elements of all binary contingency tables (one for each genre). Then, the micro-averaged BEP is computed from these accumulated statistics.
Note that the noisy web pages are not considered in evaluation process.

To measure the performance, we used the $10 \times k$ cross-validation. This means that we randomly split each corpus into k equal parts. Then we used one part for testing and the remaining parts for training. This process was performed 10 times and the final performance is the average of the 10 individual performances. The number k is identified experimentally according to the used features and corpora.

5.1 Results

In the following paragraphs, we describe a number of experiments and show the results.

Effect of Incremental Aspect In this experiment, we varied the proportion of unlabeled web pages between 10% and 90% by step of 10%. For each proportion, we measured the micro-averaged BEP for each feature set and corpus. The results are illustrated in Figure 1.

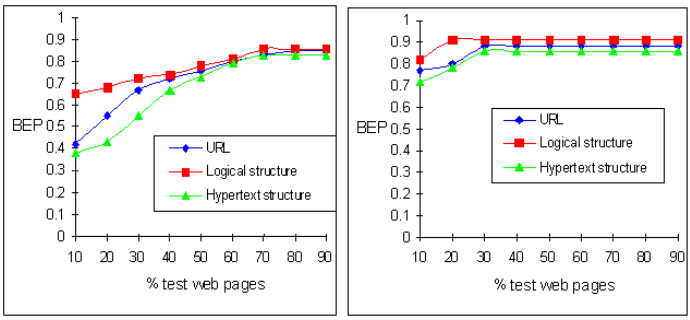

Figure 1: Micro-averaged BEP for each feature and for both KI-04 (Left) and WebKB (Right) corpora when the proportion of test pages is varied between 10% and 90%

The curves presented in Figure 1 shows that micro-averaged BEP depends on the proportion of labeled and unlabeled web pages. These curves also show that it is the logical structure classifier that achieves the best performance for both KI-04 and WebKB corpora. The proportions of unlabeled and labeled web pages to achieve the best performance are presented in the Table 3. These proportions are used in the next experiments.

Table 3: Best proportions of training and test web pages (Test%-Train%)

	URL	Logical Structure	Hypertext Structure
KI-04	80%-20%	70%-30%	70%-30%
WebKB	30%-70%	20%-80%	30%-70%

Effect of Vocabulary Size The aim of this experiment is to identify the ideal number of terms to achieve the best performance. For this purpose, we calculated the micro-averaged BEP by varying the number of terms between 5 and 3000. The number of terms complies to the information gain measure. Note that in this experiment, we used the $tfidf$ weighting technique without normalization. The obtained results are illustrated in Figure 2.

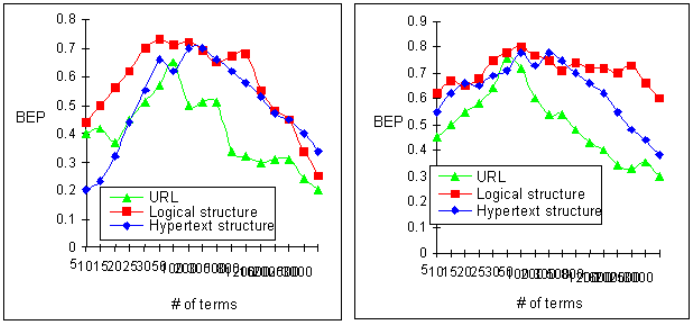

Figure 2: Micro-averaged BEP for each feature and for both KI-04 (Left) and WebKB (Right) corpora when the number of terms is varied between 5 and 3000

The ideal number of terms to achieve the best performance are summarized in Table 4. These values will be used in the next experiment.

Table 4: Best values of number of terms

	URL	Logical Structure	Hypertext Structure
KI-04	100	50	200
WebKB	50	100	300

Effect of Term Weighting In order to evaluate the effect of each normalization factor alone ($icsd$, csd and sd), we conducted an experiment whose results are showed in Table 5.

Table 5: The effect of each normalization factor on genre categorization performance

KI-04					
α	β	γ	**URL**	**Logical**	**Hypertext**
1	0	0	0.63	0.74	0.66
-1	0	0	0.66	0.75	0.68
0	1	0	**0.68**	**0.76**	0.72
0	-1	0	0.67	0.68	0.68
0	0	1	0.64	0.63	0.7
0	0	-1	0.66	0.68	**0.73**
WebKB					
1	0	0	**0.78**	**0.83**	**0.81**
-1	0	0	0.75	0.81	0.76
0	1	0	0.72	0.78	0.70
0	-1	0	0.70	0.75	0.65
0	0	1	0.65	0.80	0.64
0	0	-1	0.71	0.72	0.58

We observed that the *icsd* factor is very suitable for the KI-04 corpus because it contains heterogeneous genres. On the other hand, the *sd* factor achieves the best performance for the WebKB corpus because it contains homogenous genres.

Table 6: The effect of normalization on genre categorization performance

KI-04					
α	β	γ	**URL**	**Logical**	**Hypertext**
0.5	1	0.5	0.66	0.72	0.7
-0.5	0.5	1	0.45	0.78	0.75
0.5	-1	0	0.72	0.8	0.67
0.5	-0.5	-0.5	0.6	0.81	0.63
0.5	0	-0.5	0.68	0.73	0.55
-1	0.5	-0.5	0.7	0.65	0.77
0.5	**-1**	**-0.5**	**0.7**	**0.81**	**0.82**
1	0	-0.5	0.66	0.55	0.8
-1	-0.5	1	0.7	0.52	0.81
WebKB					
0.5	1	0.5	0.76	0.68	0.43
-0.5	0.5	1	0.77	0.66	0.71
0.5	-1	0	0.83	0.73	0.74
0.5	-0.5	-0.5	0.85	0.45	0.55
0.5	0	-0.5	0.81	0.65	0.45
-1	0.5	-0.5	0.56	0.76	0.82
0.5	**-1**	**-0.5**	**0.86**	**0.86**	**0.84**
1	0	-0.5	0.83	0.77	0.68
-1	-0.5	1	0.65	0.58	0.52

To choose the appropriate value of normalization parameters, we varied the values of α, β and γ between -1 and 1 by a step of 0.5. The best results are presented in the

Table 6. The best performance is reported by setting the normalization parameters α, β and γ to 0.5, -1 and -0.5 respectively. These values will be used in the next experiment to choose the appropriate threshold.

Effect of Refining Aspects To measure the effect of refining on genre categorization, we varied the refining threshold between 0 and 1 by step of 0.1. Zero value means that is no refining. As illustrated in Figure 3, the value of threshold affects the micro-averaged BEP of genre categorization.

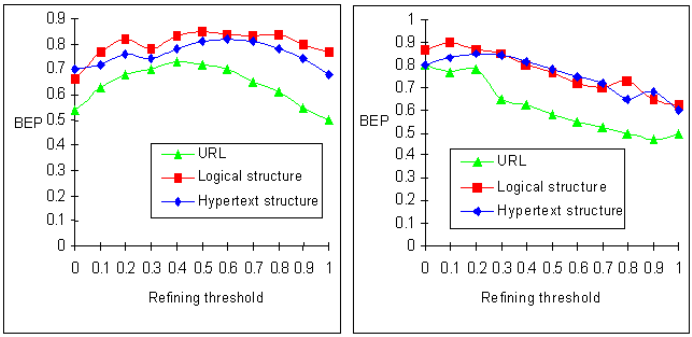

Figure 3: Micro-averaged BEP for each feature and for both KI-04 (Left) and WebKB (Right) corpora when the refining threshold is varied between 0 and 1

We noticed that in the case of noisy web pages like those contained in KI-04 corpus, the refining is very useful. On the other hand, for noiseless corpus like WebKB corpus, the refining is useless. The best refining thresholds will be used in the next experiments. The number of noisy web pages and the thresholds to achieve the best micro-averaged BEP are shown in Table 7.

Table 7: Best number of noisy web pages and refining thresholds for each feature and corpus

	URL	Logical Structure	Hypertext Structure
KI-04	7(0.4)	6(0.5)	4(0.6)
WebKB	5(0)	4(0.1)	7(0.2)

Effect of Combination Here we conducted many experiments to choose the appropriate operator for combination. The obtained results are shown in the Table 8. These results show that the decision template technique provides the best micro-averaged BEP (0.96 for KI-04 corpus and 0.98 for WebKB corpus).

Table 8: Micro-averaged BEP for each combination technique and for both KI-04 and WebKB corpora

Combination technique	KI-04	WebKB
Minimum	0.88	0.93
Maximum	0.96	0.97
Median	0.91	0.94
Vote1	0.90	0.94
Vote2	0.90	0.93
Decision templates	**0.96**	**0.98**

6 Comparisons

Accuracy The majority of the previous studies do not provide a reliable comparison with other approaches. The main reason for this is that, until recently, there were no publicly available and standard corpora for this task. Another reason is that there is not a commonly perceived sense of specific web page genres. For example, in two recent studies, user agreement was only 57In this article, we propose a comparison with other experiments, namely Meyer Zu Eissen and Stein (2004), Kanaris and Stamatatos (2007) and Santini (2007), where the KI-04 corpus is employed. The WebKB corpus is used only by Boese and Howe (2005), so we will compare our results with this experiment.

These experiments are evaluated using the accuracy measure. The micro-averaged accuracy for both KI-04 and WebKB corpora for each author is presented in Table 9. According to the results shown in this table, our approach outperforms other methods.

Table 9: Micro-averaged accuracy for both KI-04 and WebKB corpora

Author	KI-04	WebKB
Meyer Zu Eissen and Stein (800 web pages)	0.70	-
Boese and Howe	0.75	0.80
Kanaris and Stamatatos (1205 web pages)	0.84	-
Santini (1205 web pages)	0.70	-
Our approach	**0.96**	**0.98**

Machine Learning Techniques Since our approach is based on new learning aspects, we conducted experiments to compare it against other known machine learning methods used in genre categorization. Among these techniques, we used the Rocchio algorithm (Rocchio with α=14 and β=4 as control parameters), K-Nearest Neighbor (KNN with k=5), Support Vector Machines (SVM with Fisher Kernel), Naïve Bayes (NB) and decision trees (TreeNode). These techniques are implemented within the Rainbow toolkit. Micro-averaged BEP for each feature set and for both KI-04 and WebKB corpora are presented in Tables 10 and 11.

Table 10: Micro-averaged BEP for the KI-04 corpus

	URL	Logical structure	Hypertext structure
FRICC	**81.12**	**85.44**	83.17
SVM	80.76	84.75	**84.20**
Rocchio	77.55	82.10	82.15
NB	71.95	79.65	81.45
KNN	67.35	65.85	80.56
TreeNode	62.77	61.89	65.55

Table 11: Micro-averaged BEP for the WebKB corpus

	URL	Logical structure	Hypertext structure
FRICC	**85.73**	**87.32**	84.24
SVM	84.33	86.88	82.29
Rocchio	82.18	87.24	**88.78**
NB	80.88	86.76	80.85
KNN	70.22	74.11	76.16
TreeNode	61.89	64.40	62.33

Statistical Significance To determine the statistical significance of the results, we used 5×2 cross validation $t-test$ (Dietterich, 1998). The results are presented in Table 12. The symbols used in this table are defined as follows:

- $\approx$ Indicates no significant differences.
- $<$ Indicates that the machine learning method achieves a significantly lower measurement than $FRICC$ with 0.05 as a significance level.
- $<<$ Indicates that the machine learning method achieves a significantly lower measurement than $FRICC$ with 0.01 as a significance level.
- $<<<$ Indicates that the machine learning method achieves a significantly lower measurement than $FRICC$ with 0.005 as a significance level.

Table 12 shows that the $FRICC$ approach outperforms all other machine learning methods in 27 cases. Only SVM has similar performance to $FRICC$.

Training and Test Times Here we consider another important aspect, namely execution speed. Time is a very important aspect, especially when genre classification has to be integrated in a search engine. Figures 4, 5 and 6 show a comparison of the execution speeds for each classification method, in both training and test phases, for the KI-04 and WebKB corpora.

Results show that our approach is the fastest, but also Rocchio and SVM have a good performance. These results indicate that the required time is proportional to the number of categories instead of the number of web pages. Decision tree is, indeed, the slowest machine learning technique for all feature sets and for both corpora.

Table 12: Statistical Significance of our approach $FRICC$ against other machine learning techniques

KI-04			
	URL	**Logical Structure**	**Hypertext Structure**
SVM	$\approx$	$<<$	$<<$
Rocchio	$<<$	$<$	$<<$
NB	$<<$	$<<$	$<<<$
KNN	$<<$	$<<$	$<<<$
TreeNode	$<<<$	$<<<$	$<<<$
WebKB			
SVM	$\approx$	$<<$	$\approx$
Rocchio	$<<$	$<$	$<<$
NB	$<<$	$<<$	$<<<$
KNN	$<<<$	$<<$	$<<<$
TreeNode	$<<<$	$<<<$	$<<<$

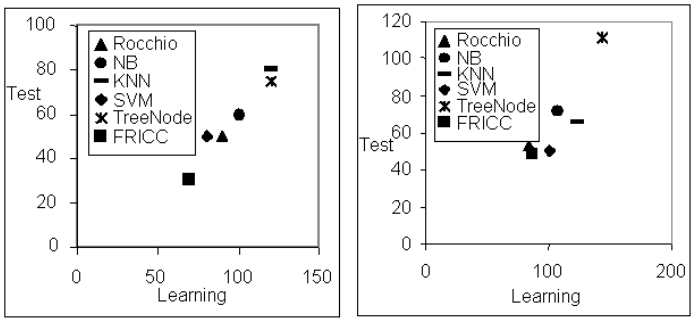

Figure 4: Training and test times for URL and for both KI-04 (left) and WebKB (right) corpora

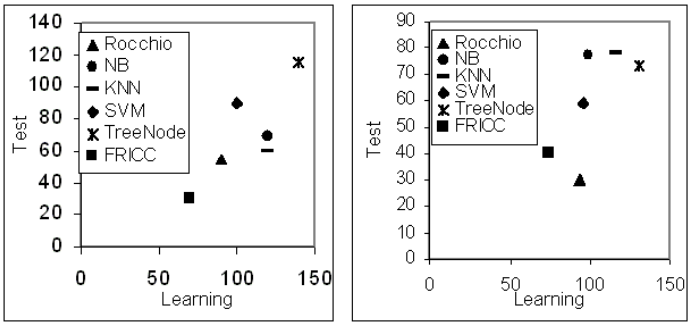

Figure 5: Training and test times for logical structure and for both KI-04 (left) and WebKB (right) corpora

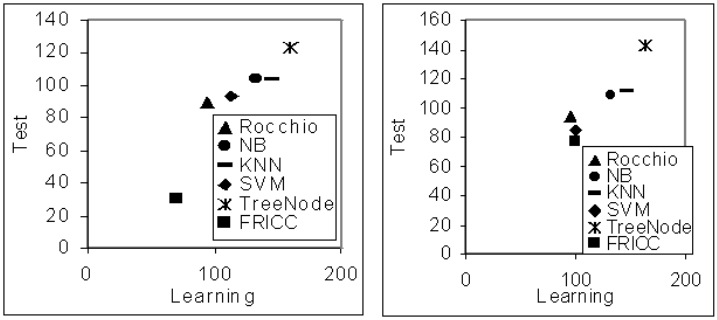

Figure 6: Train and test times for hypertext structure and for both KI-04 (left) and WebKB (right) corpora

7 Conclusions and Future work

In this article, we proposed a new approach for genre categorization of web pages. Our approach implements four new aspects that were not explored in previous studies on genre categorization. These aspects are flexibility, refining, incrementing and combination. Additionally, we conducted many experiments to measure the effectiveness, efficiency and speed of these aspects. Comparisons with previous approaches shows that our method is very fast and outperforms results documented in previous work.

In the future we hope to investigate the following points:

- As the pdf format is a very useful format on the web, we propose to classify pdf documents.
- In this work, we used only English web page, in the future we wish to focus on Arabic web documents.
- As our approach is very fast and outperforms many other machine learning techniques, we hope to include it in a search engine (i.e. Google, FireFox), in a similar way as the WEGA add-on (Stein et al.) .

Remark

The work described in this article summarizes the PhD thesis "Catégorisation Flexible et Incrémentale avec Raffinage de Pages web par Genre", completed by the author, Chaker Jebari, in October 2008, *Tunis El Manar University*, College of Science, Computer Science Department, Tunisia.

Acknowledgements

The author would like to thank the anonymous reviewers, the proofreaders and the journal's editors for useful comments, for the stylistic improvement of the article, and for the considerable editorial effort invested in the publication of this work.

References

Argamon, S., Koppel, M., and Avneri, G. (1998). Routing documents according to style. In *Proc. International Workshop on Innovative Internet Information Systems (IIIS-98)*, Pisa.

Boese, E. S. (2005). *Stereotyping the Web: Genre Classification of Web Documents (M.S. Thesis)*. Computer Science Department, Colorado State University, USA.

Boese, E. S. and Howe, A. E. (2005). Effect of web document evolution on genre classification. In *Proceedings of the 14th ACM International conference on Information and knowledge management*.

Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., and Slattery, S. (1998). Learning to extract symbolic knowledge from the word wide web. In *Proceedings of the 10th conference on artificial Intelligence*.

Dewdney, N., VanEss-Dykema, C., and MacMillan, R. (2001). The form is the substance: Classification of genres in text. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics and 10th Conference of the European Chapter of the Association for Computational Linguistics*.

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Comput*, 10(7):1895–1923.

Finn, A. (2002). *Machine learning for genre classification (M.S. Thesis)*. Computer Science Department, University College of Dublin, UK.

Finn, A. and Kushmerick, N. (2003). Learning to classify documents according to genre. In *Proceedings of the Workshop "DOING IT WITH STYLE: Computational Approaches to Style Analysis and Synthesis*, Mexico.

Herrera, F. and Verdegay, J. L. (1996). *Genetic algorithms and soft computing*. PhysicaVerlag, Heidelberg, Germany.

Joachims, T. (1997). A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In *Proceedings of ICML-97*, pages 143–151.

Joachims, T. (1998). Text categorization with support vector machines: learning with many relevant features. In *Proceedings of 10th European Conference on Machine Learning*.

Kanaris, I. and Stamatatos, E. (2007). Webpage genre identification using variable length character n-grams. In *Proceeding of the 19th IEEE International Conference on Tools with Artificial Intelligence*.

Karlgren, J. (1999). Stylistic experiments in information retrieval. In *Natural Language Information Retrieval*.

Kennedy, A. and Shepherd, M. (2005). Automatic identification of home pages on the web. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*.

Kessler, B., Nunberg, G., and Schütze, H. (1997). Automatic detection of text genre. In *Proceedings of the 35^{th} ACL/8^{th} EACL*, pages 32–38.

Kuncheva, L. I., Bezdek, J. C., and Duin, R. P. (2001). Decision templates for multiple classifier fusion. *Pattern Recognition*, 34(2):299–314.

Lertnattee, V. and Theeramunkong, T. (2004). Effect of term distributions on centroid-based text categorization. *Journal of Information Sciences*, 158(1):89–115.

Lim, C. S., Lee, K. J., and Kim, G. C. (2005). Multiple sets of features for automatic genre classification of web documents. *Journal of Information processing and management*, 41(5):1263–1276.

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165.

Mehler, A., Geibel, P., and Pustylnikov, O. (2007). Structural classifiers of text types: Towards a novel model of text representation. *LDV Forum*, 22(2):51–65.

Mehler, A., Sharoff, S., and Santini, M., editors (2009). *Genres on the Web: Computational Models and Empirical Studies*. Springer, Berlin/New York.

Meyer Zu Eissen, S. (2007). *On Information Need and Categorizing Search (PhD. Thesis)*. University of Paderborn.

Meyer Zu Eissen, S. and Stein, B. (2004). Genre classification of web pages: User study and feasibility analysis. In *Proceedings KI 2004: Advances in Artificial Intelligence*, pages 256–269.

Mitchell, T. (1997). *Machine learning*. McGraw-Hill.

Porter, M. (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137.

Rauber, A. and Müller-Kögler, A. (2001). Integrating automatic genre analysis into digital libraries. In *JCDL '01: Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries*, pages 1–10.

Salton, G. and Buckley, C. (1988). Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523.

Santini, M. (2007). *Automatic identification of genre in web pages (PhD. Thesis)*. University of Brighton, UK.

Shepherd, M., Watters, C., and Kennedy, A. (2004). Cybergenre: automatic identification of home pages on the web. *Journal of Web Engineering*, 3(3):236–251.

Shepherd, M. A. and Watters, C. (1998). Evolution of cybergenre. In *Proceedings of the 31nd Hawaiian International Conference on System Sciences*.

Stamatatos, E., Fokatakis, N., and Kokkinakis, G. (2000). Text genre detection using common word frequencies. In *Proceedings of the 18th International Conference on Computational Linguistics*.

Stein, B., Meyer zu Eissen, S., and Lipka, N. Web genre analysis: Use cases, retrieval models, and implementation issues.

Vapnik, V. (1995). *The Nature of Statistical Learning*. Springer-Verlag.

Yager, R. (1988). On ordered weighted averaging aggregation operators in multi-criteria decision making. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1):183–190.

Yang, Y. and Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *Proceedings of ICML-97*.

Zadeh, L. (1983). The role of fuzzy logic in the management of uncertainty in expert systems. *Fuzzy Sets and Systems*, 11(3):199–227.