Procesamiento del Lenguaje Natural, Revista nº 54, marzo de 2015, pp 13-20 recibido 20-10-14 revisado 27-01-15 aceptado 11-02-15

A Combination based on OWA Operators for Multi-label Genre Classification of web pages

Una combinación basada en operadores OWA para la Clasificación de Género Multi-etiqueta de páginas web

Chaker Jebari
Colleges of Applied Sciences
P. O. Box 14, P.C. 516, Sultanate of Oman
jebarichaker@yahoo.fr

Resumen: En este trabajo se presenta un nuevo método para la identificación de género que combina clasificadores homogéneos utilizando OWA (promedio ponderado) Pedimos operadores. Nuestro método utiliza caracteres n-gramas extraídos de diferentes fuentes de información, tales como URL, título, encabezados y anclajes. Para hacer frente a la complejidad de las páginas web, se aplicó MLKNN como un clasificador multi-etiqueta, en el que una página web puede verse afectada por más de un género. Los experimentos llevados a cabo usando un conocido corpus multi-etiqueta muestran que nuestro método logra buenos resultados.

Palabras clave: OWA, combinación, multi-etiqueta, clasificadores, género, página web.

Abstract: This paper presents a new method for genre identification that combines homogeneous classifiers using OWA (Ordered Weighted Averaging) operators. Our method uses character n-grams extracted from different information sources such as URL, title, headings and anchors. To deal with the complexity of web pages, we applied MLKNN as a multi-label classifier, in which a web page can be affected by more than one genre. Experiments conducted using a known multi-label corpus show that our method achieves good results.

Keywords: OWA, combination, multi-label, classifier, genre, web page.

1 Introduction

As the World Wide Web continues to grow exponentially, the classification of web pages becomes more and more important in web searching. Web page classification, assigns a web page to one or more predefined classes.

According to the type of the class, the classification can be divided into sub-problems: topic classification, sentiment classification, genre classification, and so on. 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 “Java” will provide a list of web pages containing the word “Java” and belonging to different genres such as “tutorial”, “exam”, “Call for papers”, etc. Therefore, web page genre classification could be used to improve the retrieval quality of search engines (Stein and Meyer, 2008).

Generally speaking, a genre is a category of artistic, musical, or literary composition characterized by a particular style, form, or content, but more specialized characterizations have been proposed (Santini, 2007).

According to Shepherd and Watters (1998), the genres found in web pages (also called cyber-genres) are characterized by the triple <content, form, functionality>. The content and form attributes are common to non-digital genres and refers to the text and the layout of the web page respectively. The functionality attribute concerns exclusively digital genres and describes the interaction between the user and the web page.

A common fact for all defintions is that genre and topic are orthogonal, meaning that documents addressing the same topic can be of different genres and vice versa. Following this

ISSN 1135-5948 © 2015 Sociedad Española para el Procesamiento del Lenguaje Natural

way, we can say that a document genre describes a style of writing and/or presentation rather than the document topic. This style can be captured by exploiting the structure of the document rather than its content.

It is worth noting that a web page is a complex object that is composed of different sections belonging to different genres. For example, a conference web page contain information on the conference, topics covered, important dates, contact information and a list of hyperlinks to related information. This complexity need to be captured by a multi-label classification scheme in which a web page can be assigned to multiple genres.

In this paper we used character n-grams extracted from different sources such as URL, title, headings and hyperlinks. Our constribution is to use OWA (Ordered Weighted Averaging) operators to combine the outputs of three homogenous classifiers: contextual, logical and hyperlink classifiers.

The contextual classifier uses the URL which defines the location of a web page. It is composed of three parts: host name (domain), directory path and file name (Berners-Lee, Fielding, and Masinter, 1998). The URL is not expensive to obtain and it is one of the more informative sources about the genre of the web page. URLs are often meant to be easily recalled by humans, and web sites that follow good design techniques will encode useful words that describe their resources in the web site’s host name (domain). Web sites that present a huge amount of information often break their contents into web pages. This information structuring is also accompanied with URLs structuring. For example, if the file extension is PDF, PS or DOC, then the document is long and it can be a paper, a book, a thesis, a manual, etc. Another example, if the file name contain some genre specific words like faq, cv, how, thesis, etc., we can easily recognize the genre of the web page.

The structure of a web page were used to identify the genre (Crowston and Williams, 1997; Jebari and Ounalli, 2004).

Jebari and Ounalli (2004) investigated the usefulness of the internal, also called logical structure to identify the genre of a web page. They used words included in the title and headings to extract the internal structure.

The hyperlink structure has been investigated by Crowston and Williams (1997) to identify the form of the web page and therefore can help to identify its genre.

In our work we have used the hypertext structure in different way than used by the previous researches. In our work we have used the character n-grams and the words contained in hyperlinks contrary to many other researches that use the number of internal and external links, number of images, etc. (Crowston and Williams, 1997; Boese and Howe, 2005; Lim, Lee, and Kim, 2005).

The remainder of the paper is organized as follows. Section 2 reviews previous works on genre classification of web pages. Section 3 describes the multi-label classification. Section 4 presents a brief overview about classifier combination and describes in details OWA operators. Section 5 describes our method. Section 6 evaluates and compares our method with other previous works. Finally, Section 7 concludes our paper and suggests future research directions.

2 *Related works*

A broad number of studies on genre classification of web documents have been proposed in the literature (Santini, 2007). These studies differ with respect to the following three factors: 1) the features used to represent the web document, 2) the classification methods used to identify the genre of a given web document and 3) the list of genres used in the evaluation, called also genre palette.

Many types of features have been proposed for automatic genre classification. These features can be grouped on four groups. The first group refers to surface features, such as function words, genre specific words, punctuation marks, document length, etc. The second group concerns structural features, such as Parts Of Speech (POS), Tense of verbs, etc. The third group is the presentation features, which mainly describe the layout of document. Most of these features concerns HTML documents and cannot be extracted from plain text documents. Among these features we quote the number of specific HTML tags and links. The last group of features is often extracted from metadata elements (URL, description, keywords, etc.) and concerns only structured documents.

Once a set of features has been extracted it is necessary to choose a classification method, which are often based on machine learning techniques such as Naive bayes, SVM, K-nearest neighbor, decision trees, neural networks, centroid-based techniques, etc. (Mitchell, 1997). Broadly speaking, classification methods can be divided into two main categories: single-label and multi-label methods (Tsoumakas, Katakis, and Vlahavas, 2010). In single label methods, a document is associated to only one label, whereas, in multi-label methods, a document is assigned to a set of labels.

The third factor concerns the list of genres used for the evaluation. Many genre corpora[1] (KI-04, KRYS-I, 20-genre, SANTINIS, etc.) have been compiled and used to evaluate genre identification tasks. These corpora differ with respect to the number of genres, the types of genres and the number of documents associated to each genre.

Table 1 presents an overview of features, machine learning techniques and corpora used in web genre classification.

Autor	**Features**	**Machine learning**	**Corpora**
(Meyer and stein, 2004)	HTML tag frequencies, classes of words (names, dates, etc.), frequencies of punctuation marks and POS tags	Discriminant Analysis	KI-04
(Lim, Lee, and Kim, 2005)	POS tags, URL, HTML tags, token information, most frequent function words, most frequent punctuation marks, syntactic chunks	K-Nearest Neighbor	The corpus consists of 1224 documents distributed across 15 genres (home page, public, commercial, bulletin, link collection, image collection, FAQ, discussion, product specification, etc.)
(Kennedy and Shepherd, 2005)	Content features (common words, Meta tags), form features (e.g. number of images), and functionality features (e.g., number of links, use of JavaScript).	neural network	The corpus is composed of 321 web pages classified as home pages or as noise pages (not home page). The home pages are classified into three subgenres (corporate home pages, personal home pages and organization home pages.
(Santini, 2007)	Most frequent English words, HTML tags, POS tags, punctuation symbols, genre-specific core vocabulary	SVM	SANTINIS
(Vidulin, Lustrek, and Gams, 2009)	Surface features (unction words, genre-specific words, sentence length). Structural features (POS tags, sentence type). Presentation features describe the formatting of a document through the HTML tags. Context features describe the context in which a web page was found (e.g. URL, hyperlinks, etc.).	AdaBoost	20-genre
(Kim and Ross, 2008)	Image features (extracted from the visual layout of the first page) Style features: genre-prolific words. Textual features are represented by a bag of words extracted from the content of the PDF document.	Naive bayes, SVM, Random Forest	KRYS-I
(Jebari, 2008)	Words extracted from URL, title, headings and anchors	Centroid-based	KI-04 and WebKB
(Kanaris and Stamatatos, 2009)	Character n-grams extracted from text and structure	SVM	20-genre
(Mason, 2009)	Character n-grams extracted from the textual content	SVM	20-genre
(Abramson and Aha, 2012)	Character n-grams extracted from URL	SVM	Syracuse and SANTINIS corpora

Table 1: Overview of previous works

[1] http://www.webgenrewiki.org/index.php5/Genre_Collection_Repository

3 Multi-label classification

In traditional single-label classification, a classifier is built and trained using a set of examples associated with just one single label l of a set of disjoint labels $L=\{l_1, l_2, ...l_i, ...\}$, where $|L|>1$. Moreover, in multi-label classification, the examples can be associated with a set of labels $Y \subseteq L$. In the literature, different methods have been proposed to be applied to multi-label classification problems. These methods are grouped into two main categories: problem transformation and algorithm transformation (Tsoumakas, Katakis, and Vlahavas, 2010).

Problem transformation methods are algorithm independent and transform a multi-label learning problem into one or more single-label learning problems. The most widely used transformation methods are Binary Relevance BR, Label Power Set (LP) and Random k-labelsets method (RA*k*EL). The algorithm transformation methods extend existing learning algorithms to deal with multi-label data directly. Several transformation methods have been proposed in the literature such as BR-SVM, BPMLL and MLKNN.

MLKNN is an instance-based learner (Zhang and Zhou, 2007), it learns a single classifier h_i for each label $l_i \in L$. However, instead of using the standard k-nearest neighbor (KNN) classifier as a base learner, it implements h_i by means of a combination of KNN and Bayesian inference. Given an example x, it finds the k nearest neighbors of x in the training data and counts the number of occurrences of l_i among these neighbors. Considering this number, y, as information in the form of a realization of a random variable Y, the posterior probability of $l_i \in L$ is given by:

$$P(l_i \in L/Y=y)=\frac{P(Y=y/l_i \in L)\cdot P(l_i \in L)}{P(Y=y)} \quad (1)$$

This, leads to the following classification:

$$H_i(x)=\{(l_i, f(l_i), ..., (l_i, f(l_i)), ...\} \quad (2)$$

Where $f(l_i)$ is the posterior probability of $l_i \in L$ defined in the previous equation.

The prior probability $P(l_i \in L)$ as well as the conditional probability $P(Y=y/l_i \in L)$ are estimated from the training data in terms of corresponding relative frequencies.

4 OWA Operators

Based on the assumption that each source of information provides a different view point, a combination has the potential of providing better results than any single method. There are various methods to combine such classifiers (Kuncheva, 2004). These methods can be classified according to the classifier used. Generally, classifiers can be combined at different levels: abstract level, ranking level and measurement level (Kang and Kim, 1995). In abstract level, combination methods combine simple class labels. In ranked level, combination methods combine ranked lists of class labels ordered according to the degree of membership of the input pattern. In the measurement level, combination methods combine values provided by individual classifiers as a measure of the degree of membership of the input pattern to each class. Among the three categories, the combination of classifiers at the measurement level is expected to be the most effective, since it uses all information available.

Different types of aggregation operators are found in the literature to combine the information produced by measurement level classifiers (Beliakov, Pradera, and Calvo, 2007). A very common aggregation operator is the Ordered Weighted Averaging (OWA) operator which is first introduced in (Yager, 1988).

Broadly 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 $W=[w_1, …, w_i, …, w_n]$, such that $w_i \in [0, 1]$, $\Sigma_i w_i = 1$ and $F(a_1, …, a_n) = \Sigma_i w_i b_i$ where b_i is the i-th largest element in the collection $a_1, …, a_n$. Yager suggested two methods to identify the weights w_i's. The first approach uses learning techniques and the second one uses fuzzy linguistic quantifiers to gives semantics to the weights. Herrera and Verdegay (Herrera and Verdegay, 1996) defined a quantifier function as follows:

$$Q(r)=\begin{cases} 0 & r \prec a \\ \frac{r-a}{r-b} & r \in [a,b] \\ 1 & r \succ b \end{cases} \quad (3)$$

Where $a, b \in [0, 1]$ are two parameters.

Using this quantifier function, Yager (1988) computes the weight w_i as follows:

$$w_i = Q\left(\frac{i}{n}\right) - Q\left(\frac{i-1}{n}\right), \text{ for } i=1, 2, \ldots, n \qquad (4)$$

Where n is the number of classifiers to combine. According to Yager (Yager, 1988), using the quantifier function defined above, we can identify 5 common OWA operators which are: Minimum, Maximum, Average, Vote1 and Vote2.

5 *Proposed approach*

This section describes how a web page is represented and how a new web page is classified.

5.1 Web page representation

To represent a web page, our approach performs five pre-processing steps:

Step1. This step consists in extracting the content of the elements URL, title, headings and anchors.

Step2. In this step, our method processes the content of each element separately, by removing digits, special characters (., :, /, ?, &, -, _, $, #, etc.) and stop words that differ according to the element. For the URL element we removed the stop words (http, www, etc.), since they are commonly used in all URLs. For the rest of the elements (title, headings and anchor) we removed the known stop words such as: the, of, for, etc.

Step3. This step consists in extracting words and character n-grams from all elements (URL, title, headings and anchors). A character n-grams is a set of n contiguous characters. For example, from the string 'myCV', we can extract 3 different 2-grams (my, yc, cv), 2 different 3-grams (myc, ycv) and one 4-gram (myCV). In our approach we extracted all character n-grams of length between 2 and 5, since they can capture all genre specific words in the URL.

Step4. One of the main challenges of text classification tasks is the high dimenstionality. A typical text will contain a hundreds of features, hence it is extremely difficult to produce an accurate classification without any dimension reduction. Many dimension reduction techniques have been proposed in the literature (Yang and Pedersen, 1997). In this paper we used the Document frequency thresholding technique. Given a term *t*, this technique computes the document frequency *DF* by counting the number of documents in which the term *t* occurs. Then reduce the terms whose document frequency is less than a predefined threshold. In this study, we decided to keep only URL words and character n-grams that appear in at least 100 web pages. For the other elements (title, headings and anchors), we removed words and character n-grams that appears in less than 10 web pages.

Step5. Using the Vector Space Model (VSM) (Salton and Buckley, 1988), a web page is represnetated by a vector where each term is assoictaed with a weight using the *TFIDF* weighting formula (Sebastiani, 2002).

5.2 Classification of a new web page

Given a new webpage p_i, our approach applies the five pre-processing steps described in the previous section to extract character n-grams from different sources (URL, title, headings and anchors). A web page p_i is represented by three vectors. The first vector cp_i, called contextual vector, contains character n-grams extracted from the URL. The second vector lp_i, called logical vector, contains character n-grams extracted from title and headings. The third vector hp_i, called hyperlink vector and contains character n-grams extracted from the anchors. The vectors cp_i, lp_i and hp_i are used to perform contextual, logical and hyperlink classifications named respectively $CC(cp_i)$, $LC(lp_i)$ and $HC(hp_i)$.

For a predefined set of genres G={g_1, …, g_i, …, g_m}, the contextual, logical and hyperlink classifications are defined as follows:

$$\begin{aligned} &CC(cp_i) = \{(g_1, \alpha_1), \ldots, (g_i, \alpha_i), \ldots, (g_m, \alpha_m)\} \\ &LC(lp_i) = \{(g_1, \beta_1), \ldots, (g_i, \beta_i), \ldots, (g_m, \beta_m)\} \qquad (5) \\ &HC(hp_i) = \{(g_1, \lambda_1), \ldots, (g_i, \lambda_i), \ldots, (g_m, \lambda_m)\} \end{aligned}$$

Where α_i, β_i and λ_i are the similarities between the web page p_i and the genre g_i, for the contextual, logical and hyperlink classification respectively. This similarity is calculated using the cosine formula.

In order to provide a final classification, our approach combines the contextual, logical and hyperlink classifications using the different OWA operators.

For a given web page p_i, the final classification $C(p_i)$ is defined as follows:

$$\begin{aligned} C(p_i) &= OWA_j(CC(cp_i), LC(lp_i), HC(hp_i)) \qquad (6) \\ &= \{(g_1, OWA_j(\alpha_1, \beta_1, \lambda_1)), \ldots, \\ &\qquad (g_i, OWA_j(\alpha_i, \beta_i, \lambda_i)), \ldots, \\ &\qquad (g_m, OWA_j(\alpha_m, \beta_m, \lambda_m)) \} \end{aligned}$$

Where OWA_j is one of the five OWA operators introduced in Section 4.

6 *Experimentation*

Our experimentation methodology is to experiment contextual, logical, hyperlink and combined separately. In our experimentation we used MLKNN classifier. This classifier is already implemented in the Mulan toolkit[2]. In our experimentation, we followed the k-cross-validation procedure which consists of randomly splitting the corpus into k equal parts. Then we used k-1 parts for testing and the remaining one part for training. This process is performed k times and the final performance is the average of the k individual performances. Due to the small number of web pages in each genre, we decided to use 3-cross-validation.

6.1 Corpus

In this paper we used the corpus 20-genre (Vidulin, Lusterk, and Gams, 2007). For the best of my knowledge, 20-genre is the only multi-label genre corpus available at the moment. This corpus consists of 1539 English web pages classified into 20 genres as shown in Table 2.

Genre	#pages	Genre	#pages
Blog	83	Index	308
Adult	79	Informative	318
Children's	113	Journalistic	206
Commercial/ Promotional	193	Official	85
Community	82	Personal	133
Content Delivery	207	Poetry	76
Entertainment	126	Prose Fiction	75
Error Message	90	Scientific	98
FAQ	71	Shopping	81
Gateway	119	User Input	96

Table 2: Composition of 20-genre corpus

6.2 Evaluation metrics

The evaluation of multi-label classifiers requires different evaluation metrics from those used in single-label classifiers. In a single-label classification, conventional metrics such as accuracy, precision, and recall are used to verify that an example is correctly or incorrectly classified. However, performance evaluation in multi-label classification is much complicated than traditional single-label setting, as each example can be associated with multiple labels simultaneously. Several multi-label evaluation metrics have been proposed in the literature (Tsoumakas, Katakis, and Vlahavas, 2010).

In this study, we used the following metrics: Hamming Loss, Micro-averaged precision, One-Error, Coverage and Ranking Loss.

Hamming Loss (HL) evaluates how many times an example-label pair is misclassified. The smaller the value of HL, the better the performance. The performance is perfect when the value of HL is 0.

Micro-averaged precision (MP) is the precision averaged over all the example/label pairs. The higher the value of the MP, the better the performance.

One-Error (OE) evaluates how many times the top-ranked label is not in the set of relevant labels of the example. The smaller the value of OE, the better the performance.

Coverage (CV) evaluates how far, on average, we need to go down the list of ranked labels in order to cover all the relevant labels of the example.The smaller the value of CV, the better the performance.

Ranking Loss (RL) evaluates the average fraction of label pairs that are reversely ordered for the particular example. The smaller the value of RL, the better the performance, so the performance is perfect when RL=0.

6.3 Results and discussion

6.3.1 Experiment1

In this experiment, we evaluate the contextual (CC), logical (LC) and hypertext (HC) classifiers using character n-grams and bag of words (BOW) representations. The results are reported in Table 3.

		HL	OE	RL	CV	MP
CC	Grams	0.082	0.700	0.312	7.126	0.602
	BOW	0.085	0.712	0.344	7.110	0.550
LC	Grams	0.081	0.412	0.215	8.774	0.901
	BOW	0.080	0.415	0.300	9.005	0.805
HC	Grams	0.081	0.560	0.280	8.123	0.720
	BOW	0.084	0.670	0.320	8.250	0.680

Table 3: Results achieved by contextual, logical and hypertext classifiers

By considering each classifier seperatly, we can conclude that using character n-grams achieves better results in comparison with BOW representation. Overall, the logical

[2] http://mulan.sourceforge.net/index.html

classfier (LC), reported the best results with respect to all all metrics except the Coverage metric which is better for contextual and logical classifiers. This is because, the majority of the significant genre words or grams are found in the title and heading sections. Moreoever, the contextual classifier achieves the lowest results due to the lack of genre specific words in the URL.

6.3.2 Experiment2

To evaluate the combined classifier, we used different OWA operators described in Section 4. The results achieved are presented in Table 4. Overall, the best results are achieved using Avg operator with respect to all metrics except the Coverage metric where the highest value is reported by the Vote1 operator. Moreover, we observe that the results obtained using character n-grams are much better in comparison with BOW representation.

		HL	**OE**	**RL**	**CV**	**MP**
Min	Grams	0.101	0.098	0.088	9.100	0.760
	BOW	0.201	0.102	0.090	8.550	0.720
Max	Grams	0.116	0.085	0.094	8.885	0.815
	BOW	0.186	0.082	0.090	8.900	0.770
Avg	Grams	0.065	0.054	0.082	9.118	0.941
	BOW	0.070	0.066	0.090	9.002	0.935
Vote1	Grams	0.095	0.088	0.092	7.778	0.885
	BOW	0.092	0.090	0.096	7.320	0.820
Vote2	Grams	0.058	0.055	0.082	8.226	0.920
	BOW	0.060	0.066	0.099	8.100	0.905

Table 4: Results achieved using different OWA operators

6.4 Comparison with similar works

In this section we compare our proposed method with three previous studies (See Table 5). This studies uses the multi-label corpus 20-genre.

Study	**Classifier**	**MP**
Our work	MLKNN	0.94
(Vidulin, Lustrek, and Gams, 2009)	AdaBoost	0.35
(Mason, 2009)	SVM	0.70
(Kanaris and Stamatatos, 2009)	SVM	0.74

Table 5: Classifier used and performance achieved by some previous works

As shown in the above table, our method achieves the best results. We should mention that the other studies are based on single-label classifiers such as SVM and AdaBoost, whereas in our study we used MLKNN classifier which is a multi-label classification method. It is worth noting also that all the studies used character n-grams except (Vidulin et al., 2009). So, we can confirm that using character n-grams we obtain better results rather than using other kind of features. Morover, using a multi-label classifier we can achieve better classification performance in comparison with single-label classifiers such as SVM and AdaBoost.

7 *Conclusion and future work*

In this paper, we proposed a combination of multi-label genre classifications using OWA operators. Our method exploits the character n-grams extracted from different sources such as URL, title, headings and links. The experiments conducted using a known multi-labeled corpus show that using character n-grams achieves better results than using bag-of-words. As part of the future work, we plan to evaluate our approach using other data sets, preferably with more examples. Morover, we plan to test other combination methods such as Dempester-shafer theory of evidence and Behavior Knowledge Space.

References

Abramson, M., and D. W. Aha. 2012. What's in a URL? Genre Classification from URLs. *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Beliakov, G., A. Pradera and T. Calvo. 2007. *Aggregation Functions: A Guide for Practitioners*. Springer-Verlag.

Berners-Lee, T., R. Fielding, and L. Masinter. 1998. RFC2396: *Uniform Resource Identifiers (URI): Generic Syntax*, RFC editor, USA.

Boese, E., and A. Howe. 2005. Genre Classification of Web Documents. *In Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-05)*, USA.

Crowston, K., and Williams, M. 1997. Reproduced and Emergent Genres of Communication on the World Wide Web. *In Proceedimgs of the 30th Hawaii International Conference on System Sciences*, USA.

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

Jebari, C. 2008. Catégorisation Flexible et Incrémentale avec raffinage de pages web par genre. PhD thesis, Tunis University, Tunisia.

Jebari, C., and H. Ounalli. 2004. The Usefulness of Logical Structure in Flexible Document Categorization. *In Proceeding of the International Conference on Computational Intelligence*, Turkey.

Kanaris, I., and E. Stamatatos. 2009. Learning to Recognize Webpage Genres. *Information Processing and Management journal*, 45(5): 499-512.

Kang, H. J. and J.H. Kim. 1995. Dependency relationship based decision combination in multiple classifier systems. *In Proceedings of the 14th International Joint Conf on Artificial Intelligence*.

Kennedy, A., and M. Shepherd. 2005. Automatic Identification of Home Pages on the Web. *In the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05)*, USA.

Kessler, B., G. Nunberg, and H. Schutze. 1997. Automatic detection of text genre. *In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, Morristown, NJ, USA.

Kim, Y. and S. Ross. 2008. Examining Variations of Prominent Features in Genre Classification. *In the Proceedings of the 41th Annual Hawaii International Conference on System Sciences (HICSS'08)*, USA.

Kuncheva, LI. 2004. Combining Pattern Classifiers Methods and Algorithms. John Wiley & Sons.

Lim, C. S., K. J. Lee, and G. C. Kim. 2005. Multiple sets of features for automatic genre classification of web documents. *Information Processing and Management Journal*, 41(5): 11263-1276.

Mason, J. 2009. An n-gram Based Approach to the Automatic Classification of Web Pages by Genre. PhD thesis, Dalhousie University, Canada.

Meyer, S. E., and B. Stein. 2004. Genre Classification of Web Pages. *In Proceedings of the 27th German Conference on Artificial Intelligence.*

Mitchell, T. 1997. *Machine Learning*, McGraw Hill.

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.

Sebastiani, F. 2002. Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 34(1), pp. 1-47.

Shepherd, M. and C. Watters. 1998. Evolution of Cybergenre. *In proceedings of the 31th Hawaiian International Conference on System Sciences*, USA.

Stein, B., and S. E. Meyer. 2008. Retrieval Models for Genre Classification. *Scandanivian Journal of Information Systems*. 20(1):93-119.

Tsoumakas, G., I. Katakis, I. Vlahavas. 2010. Mining Multi-Label Data. *Data Mining and Knowledge Discovery Handbook*, Springer.

Vidulin, V., M. Luštrek, and M. Gams. 2007. Using Genres to Improve Search Engines, *1st International Workshop: Towards Genre-Enabled Search Engines: The Impact of Natural Language Processing, RANLP'2007*, Borovest, Bulgaria, pp. 45-51.

Vidulin, V., M. Lustrek and M. Gams. 2009. Multi-Label Approaches to Web Genre Identification. *Journal of Language and Computational Linguistics*, 24(1): 97-114.

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

Yang, Y., and J. O. Pedersen. 1997. A Comparative Study on Feature Selection in Text Categorization, *In proceedings of ICML'1997.*

Zhang, M. L., and Z. H. Zhou. 2007. Ml-KNN: A lazy learningapproach to multi-label learning. *Pattern Recognition*, 40(1):2038–2048.