Vedrana Vidulin, Mitja Luštrek, Matjaž Gams

Multi-Label Approaches to Web Genre Identification

A web page is a complex document which can share conventions of several genres, or contain several parts, each belonging to a different genre. To properly address the genre interplay, a recent proposal in automatic web genre identification is multi-label classification. The dominant approach to such classification is to transform one multi-label machine learning problem into several sub-problems of learning binary single-label classifiers, one for each genre. In this paper we explore multi-class transformation, where each combination of genres is labeled with a single distinct label. This approach is then compared to the binary approach to determine which one better captures the multi-label aspect of web genres. Experimental results show that both of the approaches failed to properly address multi-genre web pages. Obtained differences were a result of the variations in the recognition of one-genre web pages.

1 Introduction

A web page is a complex document which can share conventions of several genres or contain several parts each of different genre. While this is recognized in the community of automatic web genre identification (AWGI), state-of-the-art implementations of genre classifiers mostly use single-label classification scheme (Karlgren and Cutting, 1994; Lim et al., 2005). In other words, they attribute to a web page one genre label from the set of predefined labels. Recent line of research (Santini, 2007, 2008), however, showed that multi-label classification scheme is more suitable for capturing the web page complexity. In our study we follow this scheme with some modifications.

The need for attributing more that one genre label to a web page is noticed by several authors (Roussinov et al., 2001; Meyer zu Eissen and Stein, 2004; Rosso, 2005), however, the primary goal of this studies was not the implementation of multi-label genre model. In contrast, Santini (2007) implemented the model based on zero-to-multi genre assignment. Her classification scheme was motivated by the two characteristics of genre: hybridism and individualization (Santini, 2008). Since several genres are easily combined in a single web page, she argues that such hybrid forms require attribution of multiple genre labels. In contrast, the absence of the mechanisms which would force the strict application of genre conventions, as in the case of e.g. scientific article, allows the individualization of genres. The individualized web pages have unclear genre conventions, and are consequently marked with zero genres. The model based on the presented scheme was implemented in two steps. First, the combinations of

facets[1] representing text types were hard-coded to obtain a middle-layer model for the recognition of text types. Text types were inferred using the modified form of Bayes' theorem, namely the odds-likelihood or subjective Bayesian method. Second, if-then rules were created to identify genres from the combination of text types and other features (e.g. linguistic, HTML).

Another model of Stubbe et al. (2007) was partially concerned with the issue of multi-label approach to AWGI. They built multiple genre-specific classifiers, one per genre, by combining features provided by the genre expert into rules. They pointed that the classifiers can be combined into a scheme which can attribute several genre labels to a web page. An improved classifier's precision was observed. However, they did not explore this issue in more detail since their main task was to exploit interdependencies between genre specific classifiers to improve the precision of a single label assignment.

In contrast to the presented approaches, which are based on expert knowledge, our goal is to induce a multi-label model from the example web pages with supervised machine learning (ML) methods. Learning a multi-label model can be achieved through the problem transformation or through the algorithm adaptation approach (Tsoumakas and Katakis, 2007). We follow problem transformation approach, and explore two transformations: a transformation to a multi-class problem and a transformation to a set of binary sub-problems. Finally, we use standard ML algorithms to learn the models from the transformed data, and we test their performances to understand which type of the model can better deal with multi-genre web pages.

The binary approach shares several characteristics with the approaches of Santini (2007) and Stubbe et al. (2007). First, we introduce a set of binary classifiers, one per genre. Second, by combining positive answers of multiple classifiers we realize zero-to-multi genre assignment, attributing zero genres when there are no positive answers, a single genre when there is only one positive answer, and multiple genres when there are several positive answers. In contrast to the related work, we do not analyze the performance of separate binary classifiers. Instead, we focus on evaluating the performance of multi-label classifier as a whole.

Our multi-class approach differs from previous research. Its main advantage is the induction of a single classifier, for which we assume that it can better learn the overlaps between different genres, and consequently better recognize web pages containing multiple genres. There are certain disadvantages to this approach. One lies in the inability to capture all genre combinations within a single corpus. The classifier, therefore, cannot properly recognize a web page containing a new genre combination. The best result which the classifier can produce in such a situation is to recognize the dominant genre. Another problem lies in the inability to properly define attribution of zero labels. Even if we introduce the label, e.g. "N/A", the question is what kind

[1] Santini (2007) defines facet as "an 'aspect' in the communicative context that is reflected in the use of language". For example, first person facet is complex feature accounting for appearance of first person pronouns in a web page, and indicates the communication context related to the text producer.

of example web pages can we put in this category to properly learn it. This problems should be explored in more detail, which is beyond the scope of this paper.

The two approaches were tested on the multi-label 20-Genre-Collection corpus, collected for the purpose of learning the classifier for implementation into a search engine. Since there is no common, widely agreed upon set of web genre categories (Rehm et al., 2008), for that purpose we defined 20 broad categories which combined into the multi-label scheme tend to be robust enough to deal with the diversity and the complexity of web pages on the open web. The corpus is composed of web pages in English, therefore examples in ML tasks are individual web pages.

The rest of the paper is organized as follows. In Section 2 we describe the web genre categories and in Section 3 the multi-label corpus we experimented with. Section 4 lists the features used to describe web pages in terms of web genre. Section 5 presents the methodology behind experiments. Section 6 presents experimental results with discussion, and Section 7 concludes the paper and presents the directions for future work.

2 Web Genre Categories

Although genres form taxonomy, for practical purposes this taxonomy is usually reduced to one level. Santini (2007) selected only basic level genres that can be directly instantiated by formulating text in the proposed genre, e.g. *Personal home page* or *FAQ*. Lim et al. (2005) used broad categories of higher level, composed of one or more basic level genres. For example, the genre *Journalistic materials* includes press reportage, editorial and review, while the genre *Informative materials* includes recipes, lecture notes and encyclopedic information. The advantage of the second approach and the reason while we are following it is that it seems more natural to cope with the diversity of the Internet. However, the disadvantage lies in the difficulty to represent common characteristics of web pages that compose such broad categories. Therefore, the genre classifiers learned on corpora with broad categories showed somewhat lower performance.

We defined our set by reusing and refining existing sets and by adding new categories. The starting point was the work of Lim et al. (2005). In total, Lim et al. (2005) selected 16 categories, 8 non-textual (*Personal homepages, Public homepages, Commercial homepages, Bulletin collections, Link collections, Image collections, Simple table/lists, Input pages*) and 8 textual (*Journalistic materials, Research reports, Official materials, Informative materials, FAQs, Discussions, Product Specifications, Others (informal texts)*). We defined 20 categories presented in the first column of Table 1. The overlaps with the Lim et al.'s (2005) categories are presented in the last column. Partial overlap is marked with an asterisk (*).

Childrens' category includes multiple genres aimed at younger audience. Common characteristics of pages belonging to this category are the use of simple language and colorful formatting. The identification of this category could be useful for children and probably even more for their parents. *Commercial/promotional* pages have the common purpose of promoting organizations and selling one's own products and services. In

contrast, selling products of others is the purpose of *Shopping* pages. *Community* page has the purpose of involving a visitor in the creation of the page, usually by contributing content in a limited way (forums), although there are pages where the users are given even more freedom. *Content delivery* page delivers content that is not part of the page (e.g. download pages). Another purpose is to present embedded non-textual content (e.g. page with flash game). Lim et al.'s (2005) *Image collections* category has a similar purpose. The difference is that we broaden it by taking into consideration other types of media. Furthermore, it is difficult to include genres such as jokes and horoscopes in commonly used genre categories. Their common purpose, which could be interesting to a search engine users, is to entertain. Therefore, we added *Entertainment* category. *Error message* pages are not particularly interesting to a visitor, and are not intended to be offered as a choice in a search engine. Instead, this category is used to filter such uninteresting pages. The purpose of *Gateway* is to transfer the visitor to another page. Some of the gateway pages (e.g. login page) are overlapping with Lim et al.'s (2005) *Input pages*. *Official pages* partially overlap with Lim et al.'s (2005) *Official materials*. For example, they labeled legal info and copyright materials pages as *Official materials*, and we would label them as *Official*. In contrast, they labeled ad page as *Official materials*, while we would label it as *Shopping*. *Personal* pages are home-made (not professionally formatted), written in informal and subjective manner. This category includes Lim et al.'s (2005) *Personal homepages* and opinions, which Lim et al. (2005) classified as *Discussions*. *Poetry* and *Prose fiction* are genres considered by Lim et al. (2005) as informal texts. Because a search engine user can have special interest in those genres (e.g. searching for lyrics of Madonna's song) we separated them in two distinct categories. Similar to *Childrens'* pages, *Pornographic* covers multiple genres. It is, however, targeted at the adult audience.

3 20-Genre Collection Corpus

We were not able to obtain a publicly available corpus, which adequately represents genre categories presented in Table 1. Therefore, we built our own corpus.

The web pages were collected from the Internet using three methods. Firstly, we used highly-ranked Google hits for popular keywords (e.g. "Britney Spears"). The keywords were chosen according to 2004 Year-End Google Zeitgeist statistics (`http://www.google.com/press/zeitgeist2004.html`). Our purpose was to build a classifier that will not have a problem with recognizing the most popular web pages, which people actually search for. 150 pages were collected by entering the most popular queries from each of the five categories from 2004 Google Zeitgeist. The collected pages ranked from 31^{st} to 60^{th} place. The first 30 hits were skipped as suggested by Lim et al. (2005) to avoid too many *Commercial/promotional* pages. In order to increase the diversity while retaining the popularity criterion, we input the most popular queries of each weekly 2004 Google Zeitgeist into Google and used the hits ranked from 31^{st} to 35^{th} (or 31^{st} to 40^{th} for pages topping the weekly Zeitgeist twice). This gave us 245 pages for a total of 395 pages. Secondly, we gathered 300 random web pages using Mangle

(http://www.mangle.ca/), a random link generator. Finally, we specifically searched for web pages belonging to the genres under-represented to that point by inputting genre-related queries into Google and using relevant hits. The purpose of the last step was to obtain a balanced corpus that represents all genres equally well. Imbalance usually causes difficulties in learning the under-represented classes. In total, 1,539 web pages in English were collected.

The corpus was manually labeled with genres by two independent annotators. Their labels disagreed on about a third of the web pages in the corpus, so those were reassessed by a third and sometimes even a fourth annotator.

The distribution of web pages over the 20 categories in the 20-Genre Collection corpus (http://dis.ijs.si/mitjal/genre/) is presented in Table 2 (the multi-label aspects of corpus are discussed in the section on learning a multi-label genre classifier). The targeted average was 100 pages per genre. We ended up with at least 55 pages of each genre and around 200 pages belonging to the most common genres. Such differences can be attributed to search engine's bias towards certain genres (e.g. *Index, Informative, Journalistic*), or because some genres are simply more common on the Internet.

4 Features

We selected a broad set of features from previous studies and combined them with features obtained from the analysis of 20-Genre Collection corpus to cover different aspects of genre: content, linguistic and visual form, and the context of a web page. In total, 2,491 features were chosen separated in four groups: surface, structural, presentation and context features.

4.1 Surface Features

Surface features pertain to content of a web page. For example, frequent appearance of function word "you" can characterize promotional pages of *Commercial/promotional* genre. They are easily extractable and, hence, commonly used (Stamatatos et al., 2000; Dewdney et al., 2001; Lim et al., 2005). This group includes function words, genre-specific words, punctuation marks, classes of words (such as dates, times, postal addresses and telephone numbers), and word, sentence and document length.

We selected 411 surface features, presented in Table 3. The set of 321 genre-specific words was obtained by combining the list of most frequent content words from the corpus with manually selected genre-describing words. They were stemmed by the Porter stemming algorithm (Porter, 1980).

4.2 Structural Features

Structural features describe syntactic choices. For example, high frequency of nouns can indicate *Informative* pages. They include features like parts of speech (POS), phrases (e.g. noun phrase or verb phrase) and sentence types (e.g. the frequencies of declarative, imperative and question sentences) (Santini, 2007).

We selected 1,908 structural features, presented in Table 4. POS tags were extracted with TreeTagger (Schmid, 1994). Beside single POS, we also extracted POS trigrams to capture pieces of syntactic constructions. To obtain the set of discriminative POS trigrams, we discarded too common and too rare trigrams (Santini, 2004) in two steps. First, we extracted only trigrams that are present more than three times in a web page. Second, we discarded 25% of the most frequent and 25% of the least frequent trigrams in the corpus.

4.3 Presentation Features

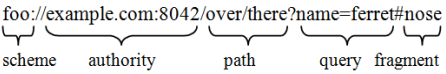
Presentation features describe the formatting of a document. For example, appearances of tags <form> and <input> can indicate *User Input* pages. This group includes token type (e.g. the percentage of a document taken by numbers or whitespaces), text formatting (e.g. amount of bolded text), graphical elements (e.g. the frequencies of images or tables) and similar (Lim et al., 2005).

We selected 93 presentation features, presented in Table 5. Token type can be used to describe formatting of any textual document, while HTML features are specific to web pages. Both single HTML tags and the groups of tags were considered. Following an idea to group tags in macro-features presented in (Santini, 2007), we grouped tags into five categories according to their functionalities.

4.4 Context Features

Context features describe the context in which a web page was found. Under context we assume URL and hyperlinks contained within the web page. URL features describe the structure and the content of URL, while hyperlink features describe types of hyperlinks. For example, appearances of words "blog" and "archive" in URL can indicate *Blog* page, while high number of hyperlinks to a different domain can indicate *Commercial/promotional* pages.

We selected 79 features, 76 URL (Table 6) and 3 hyperlink features (Table 7). The choice of URL features describing its structure follows URL syntax defined by Berners-Lee et al. (1998):



URL content was analyzed by marking the appearances of 54 words most commonly present in URL. The words were stemmed with the Porter stemming algorithm.

5 Learning a Multi-Label Genre Classifier

5.1 Data Set

The data set, to which we will refer to as 20-Genre-Collection data set, was obtained by extracting 2,491 features from 1,539 web pages in the 20-Genre-Collection corpus. All features except those pertaining to URL were expressed as ratios. Since it is more probable that a certain feature would appear more frequently in longer pages, expressing features as ratios eliminates the influence of page length.

From 1,539 web pages, 1,059 are labeled with one, 438 with two, 39 with three and 3 with four labels. On average, there are 1.34 labels per web page.

5.2 Problem Transformations

Multi-label classification assumes association of examples with a set of labels $Y \subseteq L$, L representing the set of labels present in a data set. There are two approaches to multi-label classification: problem transformation and algorithm adaptation (Tsoumakas and Katakis, 2007). We have chosen problem transformation because it allows use of the existing tools for single-label classification. Two transformations are explored in this paper: a transformation to a multi-class problem and a transformation to a set of binary problems.

The multi-class transformation assumes treatment of different sets of labels as distinct single labels. Therefore, the goal is to learn a classifier $F : X \rightarrow P(L)$, where X represents examples and $P(L)$ the power set of L. When applied to the 20-Genre-Collection data set, the categories as *Blog, Childrens', Childrens'-Informative, Community-Informative* were obtained. This transformation explicitly captures overlaps between genres, with the negative side-effect of producing high number of categories. In some cases, newly obtained categories were represented with only one example. To properly train and test a classifier, we removed all the examples labeled with the categories not represented with at least one example in the both train and test sets.

The binary transformation assumes learning $|L|$ binary sub-classifiers $F_l : X \rightarrow \{l, \neg l\}$, one for each label $l \in L$. For example, the 20-Genre-Collection data set is transformed into 20 data sets each containing all the examples of the original data set, labeled as positive (e.g. "1") if the labels of the original example contained l and as negative (e.g. "0") otherwise.

5.3 Learning Classifiers

On the transformed data we applied LibSVM (Fan et al., 2005) and AdaBoost (Freund and Schapire, 1996) to learn the classifiers.

LibSVM has built-in problem transformation functionalities, and is good at handling high number of features and sparse data. In the process of tuning the algorithm, we followed the recommendations of Hsu et al. (2008). First recommendation is to scale the data to avoid features in greater numeric ranges to dominate those in smaller

numeric ranges. We scaled the feature values to fall into the [0, 1] interval. Second recommendation is to test the RBF kernel $\left(K\left(x_i, x_j\right) = exp(-\gamma||x_i - x_j||^2), \gamma > 0\right)$ first since it can handle the cases where the relation between class labels and attributes is nonlinear. Besides, they argue that the linear kernel is a special case of RBF kernel, which is a logical second step to test since it is good at handling the problems with higher number of features in comparison to the number of examples. We compared the performances of the two kernels on our data, and the use of linear kernel instead of the RBF did not result in any improvement at all. Third recommendation is to select the parameters C and γ with the grid search in the space of models induced with exponentially growing sequences of parameters C and γ. We used the tool contained within the LibSVM, and evaluated the quality of parameters using the 3-fold cross-validation on the training set. The parameters of a model with the best cross-validation accuracy were picked. In the case of binary transformation the choice of the parameters was separately done for each sub-classifier.

AdaBoost is a meta-learning algorithm. In our previous research (Vidulin et al., 2007) we boosted J48 decision trees (Witten and Frank, 2005), and obtained the best performance among five algorithms (sequential minimal optimization, Naïve Bayes, J48 decision trees, Random Forrest and AdaBoost) tested on the binary transformation of the 20-Genre-Collection data set.

5.4 Evaluation

The performance of multi-label classifiers was evaluated using stratified 3-fold cross--validation. Stratification is a problem which can be approached in different manners in multi-label setting. One approach is to do the problem transformations first and than to separate the data into folds. It results in better balance of classes on the level of individual ML sub-problems. The second approach is to separate the folds before problem transformations and to stratify in a manner to obtain equal distribution of single genre categories over the folds. Since our goal was to obtain the same train-test splits to allow comparisons between induced classifiers we used the second approach. Considering that the number of examples per category decreases after multi-class transformation, we used three instead of ten folds to increase the chance of obtaining more test examples per class.

The performance of classifiers was evaluated with several measures: exact match ratio, micro-averaged precision, recall and F-measure, and macro-averaged precision, recall and F-measure.

Exact match ratio (EX) counts exact matches between the predicted and the actual labels (Eq. 1). This measure is, in a way, similar to accuracy in the case of the single-label classification. However, it does not account for e.g. two out of three correctly predicted labels which is fairly good success in the multi-label setup.

$$EX = \frac{\sum_{i=1}^{M} I[Y_i^{\text{predicted}} = Y_i^{\text{actual}}]}{M} \tag{1}$$

$I[S]$ is 1 if the statement S is true and 0 otherwise, and M represents the number of classified examples.

Besides measuring the error rate, we also measured precision, recall and F-measure. In the case of multi-label classification this measures are obtained as averages over all classifier's decisions – micro-averaging and over all categories – macro-averaging.

Micro-averaged measures weight all the web pages equally, representing the averages over all the (web page, genre category) pairs. They tend to be dominated by the classifier's performance on common categories. Micro-averaged precision (π(micro)) represents the ratio of web pages correctly classified as l (TP = true positives), and all the pages correctly and incorrectly (FP = false positives) classified as l (Eq. 2). Micro-averaged recall (ρ(micro)) represents the ratio of web pages correctly classified as l, and all the pages actually pertaining to the class l (FN = false negatives) (Eq. 3). Micro-averaged F-measure (F(micro)) represents a harmonic mean of π(micro) and ρ(micro) (Eq. 4). $|L|$ represents the number of categories.

$$\pi\,(\text{micro}) = \frac{\sum_{l=1}^{|L|} TP_l}{\sum_{l=1}^{|L|}(TP_l + FP_l)} \tag{2}$$

$$\rho\,(\text{micro}) = \frac{\sum_{l=1}^{|L|} TP_l}{\sum_{l=1}^{|L|}(TP_l + FN_l)} \tag{3}$$

$$F\,(\text{micro}) = \frac{2 \times \pi\,(\text{micro}) \times \rho\,(\text{micro})}{\pi\,(\text{micro}) + \rho\,(\text{micro})} \tag{4}$$

Macro-averaged measures weight equally all the genre categories, regardless of their frequencies. They tend to be dominated by the classifier's performance on rare categories. Macro-averaged precision (π(macro)) is computed firstly by computing the precision for each category separately, and then by averaging over all categories (Eq. 5). The same procedure is used for computing the macro-averaged recall (ρ(macro)) (Eq. 6), and macro-averaged F-measure (F(macro)) (Eq. 7).

$$\pi_l = \frac{TP_l}{TP_l + FP_l}, \quad \pi\,(\text{macro}) = \frac{\sum_{l=1}^{|L|} \pi_l}{|L|} \tag{5}$$

$$\rho_l = \frac{TP_l}{TP_l + FN_l}, \quad \rho\,(\text{macro}) = \frac{\sum_{l=1}^{|L|} \rho_l}{|L|} \tag{6}$$

$$F_l = \frac{2 \times \pi_l \times \rho_l}{\pi_l + \rho_l}, \quad F\,(\text{macro}) = \frac{\sum_{l=1}^{|L|} F_l}{|L|} \tag{7}$$

6 Results and Discussion

The performances of the four classifiers induced with LibSVM and AdaBoost algorithms on the multi-class and binary data sets are presented in Table 8 and Table

9. Both multi-class classifiers correctly classified higher number of examples than the binary classifiers. To understand if this happens due to better recognition of multi-genre web pages, we broke down the correct predictions into categories according the number of actual labels and the number of correctly predicted labels (Table 10). To allow the comparisons between the multi-class and the binary classifiers, we transformed the numbers of correctly predicted labels into ratios. For example, in the case of two-genre pages there were 128 examples. The LibSVM multi-class classifier correctly predicted one of the two genres for 45 examples or the 35% of the two-genre cases, and two of the two genres for 14 examples or the 11% of the two genre cases. As can be seen from the Table 10, the removal of the examples labeled with improperly represented categories in multi-class setting (cf. Evaluation section), resulted in different number of web pages per the number of labels category.

From Table 10 it can be seen that the higher EX of the multi-class classifiers in comparison to the binary classifiers is due to better recognition of single-genre web pages (on average 10 percentage points improvement). In the case of two-genre web pages the quality of recognition was on average the same – around 10%. Because of the small number of the three-genre and four-genre web pages, we cannot make proper conclusions for more than two genres per page.

Considering other qualities of the classifier, the binary classifiers showed considerably higher precision in the comparison to the multi-class classifiers. We consider high precision as a good property of genre classifier since on the open web it is of higher importance to get precise top ten hits than to retrieve all possible hits.

7 Conclusion and Future Work

In this paper we compared two approaches to multi-label web genre classification – multi-class and binary – to understand which one can better capture the relations between genres that appear together in multi-genre web pages. To this end four multi-label classifiers were induced, two multi-class and two binary. Overall performances of both multi-class and binary classifiers are relatively low. For example, the exact match ratio is around 38% for multi-class and around 29% for binary classifiers. A potential reason is the high number of features (2,491) in comparison to the number of examples (1,539), an aspect which we intend to address in further experiments through feature selection.

Binary classifiers considerably outperformed the multi-class classifiers in precision (by around 62% when micro-averaged and around 61% when macro-averaged). However, this criterion alone is not enough to make the choice between the approaches. Further evidence showed that the differences between the two approaches were largely in different ability to correctly classify single-genre web pages. Therefore, we can conclude that under presented circumstances both approaches fail to address the issue of correct recognition of multi-genre web pages.

As part of the future work, the approaches could be tested on another data set, preferably larger and with more multi-label examples. Several other ML algorithms

could be applied. This would rule out the influences of the specific corpus and the specific ML algorithms.

References

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

Dewdney, N., VanEss-Dykema, C., and MacMillan, R. (2001). The form is the substance: Classification of genres in text. In *Proceedings of the workshop on Human Language Technology and Knowledge Management*, pages 1–8. Association for Computational Linguistics.

Fan, R., Chen, P., and Lin, C. (2005). Working Set Selection Using Second Order Information for Training Support Vector Machines. *The Journal of Machine Learning Research*, 6:1889–1918.

Freund, Y. and Schapire, R. (1996). Experiments with a New Boosting Algorithm. In *Proceedings of the 13th International Conference on Machine Learning*, pages 148–156. Morgan Kaufmann.

Hsu, C., Chang, C., and Lin, C. (2008). A Practical Guide to Support Vector Classification. `http://www.csie.ntu.edu.tw/~cjlin`.

Karlgren, J. and Cutting, D. (1994). Recognizing Text Genres with Simple Metrics Using Discriminant Analysis. In *Proceedings of the 15th International Conference on Computational Linguistics*, pages 1071–1075. Association for Computational Linguistics.

Lim, C., Lee, K., and Kim, G. (2005). Multiple Sets of Features for Automatic Genre Classification of Web Documents. *Information Processing and Management*, 41(5):1263–1276.

Meyer zu Eissen, S. and Stein, B. (2004). Genre Classification of Web Pages: User Study and Feasibility Analysis. In *KI 2004: Advances in Artificial Intelligence*, pages 256–269. Springer.

Porter, M. (1980). An Algorithm for Suffix Stripping. *Program*, 14(3):130–137.

Rehm, G., Santini, M., Mehler, A., Braslavski, P., Gleim, R., Stubbe, A., Symonenko, S., Tavosanis, M., and Vidulin, V. (2008). Towards a Reference Corpus of Web Genres for the Evaluation of Genre Identification Systems. In *Proceedings of the 6th Language Resources and Evaluation Conference (LREC 2008)*.

Rosso, M. (2005). *Using Genre to Improve Web Search*. PhD thesis, University of North Carolina.

Roussinov, D., Crowston, K., Nilan, M., Kwasnik, B., Cai, J., and Liu, X. (2001). Genre Based Navigation on the Web. In *Proceedings of the 34th Hawaii International Conference on System Sciences*.

Santini, M. (2004). A Shallow Approach to Syntactic Feature Extraction for Genre Classification. In *Proceedings of 7th Annual Colloquium for the UK Special Interest Group for Computational Linguistics*.

Santini, M. (2007). *Automatic Identification of Genre in Web Pages*. PhD thesis, University of Brighton.

Santini, M. (2008). Zero, Single, or Multi? Genre of Web Pages Through the Users' Perspective. *Information Processing and Management*, 44(2):702–737.

Schmid, H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of International Conference on New Methods in Language Processing*, volume 12.

Stamatatos, E., Fakotakis, N., and Kokkinakis, G. (2000). Automatic Text Categorization in Terms of Genre and Author. *Computational Linguistics*, 26(4):471–495.

Stubbe, A., Ringlstetter, C., and Schulz, K. (2007). Genre as Noise: Noise in Genre. *International Journal on Document Analysis and Recognition*, 10(3):199–209.

Tsoumakas, G. and Katakis, I. (2007). Multi-Label Classification: An Overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13.

Vidulin, V., Luštrek, M., and Gams, M. (2007). Using Genres to Improve Search Engines. In *Proceedings of International Workshop: Towards Genre-Enabled Search Engines: The Impact of Natural Language Processing*, pages 45–51.

Witten, I. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier.

Table 1: Web Genre Categories used in this paper.

Genres	Examples	Genres in (Lim et al., 2005)
Blog	blogs, diaries, time-stamped updates	N/A
Childrens'	encyclopedia for children, lyrics for children	N/A
Commercial/ promotional	homepages of institutions, organizations, political parties, institutionalized individuals; product descriptions; service descriptions; press releases	*Public homepages, Commercial homepages, Product specifications*
Community	forums, news group pages, portals with user-generated content	*Discussions, Bulletin collections*
Content delivery	download pages, image and movie galleries, games	*Image collections*
Entertainment	jokes, puzzles, horoscopes, games	N/A
Error message	custom HTTP error pages, non-HTTP errors	N/A
FAQ	faq	*FAQs*
Gateway	introductory pages, redirection pages, login pages	*Input pages**
Index	link collections, table of contents	*Link collections*
Informative	encyclopedic materials, recipes, user manuals, how-tos, lecture notes for a wide audience, informative books, biographies, discographies, filmographies	*Informative materials*
Journalistic	news, reportages, editorials, interviews, reviews	*Journalistic materials*
Official	legal materials, official reports, rules	*Official materials**
Personal	personal homepages, pages with opinions, descriptions of interests and activities	*Personal homepages, Discussions**
Poetry	poems, lyrics	*Other (informal texts)**
Pornographic	pictures and videos, stories	N/A
Prose fiction	fanfiction story, short story, novel	*Other (informal texts)**
Scientific	papers, theses, lecture notes for a specialized audience, scientific books	*Research reports*
Shopping	online stores, classified ads, price comparators, price lists	*Official materials**
User input	forms, surveys	*Input pages*

Table 2: A composition of 20-Genre Collection corpus.

Genre	No. of pages
Blog	77
Childrens'	105
Commercial/ promotional	121
Community	82
Content delivery	138
Entertainment	76
Error message	79
FAQ	70
Gateway	77
Index	227
Informative	225
Journalistic	186
Official	55
Personal	113
Poetry	72
Pornographic	68
Prose fiction	67
Scientific	76
Shopping	66
User input	84

Table 3: A set of surface features.

Features
Function words: number of occurrences of 50 most common function words in the corpus / total number of function words
Genre-specific words: number of occurrences of 321 selected content words / total number of content words
Punctuation marks: number of occurrences of 34 selected punctuation symbols / total number of punctuation symbols
Classes of words: number of named entities of the classes date, location and person / total number of words
Text statistics: average number of characters per word; average number of words per sentence; number of characters in hyperlink text / total number of characters

Table 4: A set of structural features.

Features
POS tags: number of occurrences of 36 available POS tags / total number of words
POS trigrams: number of occurrences of 1,868 selected POS trigrams / total number of POS trigrams
Sentence types: number of declarative sentences, interrogative sentences, exclamatory sentences and other sentences (in most cases list items) / total number of sentences

Table 5: A set of presentation features.

Features
Token type: number of alphabetical tokens (sequences of letters), numerical tokens (sequence of digits), separating tokens (sequences of separator characters, such as spaces and returns) and symbolic tokens (sequences of characters excluding alphanumeric and separator characters) / total number of tokens

HTML tags: number of single tags / total number of tags; number of tags belonging to a class of tags / total number of tags for 5 classes:
1. **Text Formatting:** <abbr>, <acronym>, <address>, <b>, <basefont>, <bdo>, <big>, <blockquote>, , <cite>, <code>, <del>, <dfn>, , <font>, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, <i>, <ins>, <kbd>, <pre>, <q>, <s>, <samp>, <small>, <strike>, , <style>, <sub>, <sup>, <tt>, <u>, <var>
2. **Document Structure:**
, <caption>, <col>, <colgroup>, <dd>, <dir>, , <dl>, <dt>, <frame>, <hr>, <iframe>,   , <menu>, <noframes>,   , <p>, , <table>, <tbody>, <td>, <tfoot>, <th>, <thead>, <tr>,
3. **Inclusion of external objects:** <applet>, <img>, <object>, <param>, <script>, <noscript>
4. **Interaction:** <button>, <fieldset>, <form>, <input>, <isindex>, <label>, <legend>, <optgroup>, <option>, <select>, <textarea>
5. **Navigation:** Counting href attribute of tags <a>, <area>, <link> and <base>

Table 6: A set of context features – URL features

Features	Description
Https	Indicates whether the scheme is https.
URL depth	Number of directories included in the path.
Document type	Described by four Boolean features, each indicating whether the document type is static HTML (document extensions html and htm), script (document extensions asp, aspx, php, jsp, cfm, cgi, shtml, jhtml and pl), doc (document extensions pdf, doc, ppt and txt) or other (the other document extensions).
Tilde	Appearance of "/ " in the URL.
Top-level domain	Described by ten Boolean features, each indicating whether the top-level domain is com, org, edu, net, gov, biz, info, name, mil or int.
National domain	Indicates whether the top level domain is a national one.
WWW	Indicates if the authority starts with www.
Year	Indicates the appearance of year in the URL.
Query	Indicates the appearance of query (?foo) in the URL.
Fragment	Indicates the appearance of fragment (#foo) in the URL.
Appearance of 54 most commonly used words in URL	Indicates the appearance of common content words in URL: *about, abstract, adult, archiv, articl, blog, book, content, default, detail, download, ebai, english, error, fanfic, faq, forum, free, fun, funni, galleri, game, help, home, index, joke, kid, legal, librari, link, list, lyric, main, member, music, new, paper, person, poem, poetri, product, project, prose, pub, public, quiz, rule, search, sport, stori, topic, tripod, user, wallpap*

Table 7: A set of context features – links.

Features
Links: number of hyperlinks to the same domain, to a different domain and containing "mailto" / total number of hyperlinks

Table 8: The performances of the two classifiers induced with the LibSVM on the multi-class and binary data sets.

	LibSVM						
Data set	EX	$\pi\,(micro)$	$\rho\,(micro)$	$F\,(micro)$	$\pi\,(macro)$	$\rho\,(macro)$	$F\,(macro)$
Multi-class	38%	0.37	0.37	0.37	0.20	0.16	0.16
Binary	29%	0.55	0.29	0.38	0.50	0.34	0.34

Table 9: The performances of the two classifiers induced with the AdaBoost on the multi-class and binary data sets.

	AdaBoost						
Data set	EX	$\pi\,(micro)$	$\rho\,(micro)$	$F\,(micro)$	$\pi\,(macro)$	$\rho\,(macro)$	$F\,(macro)$
Multi-class	38%	0.35	0.35	0.35	0.15	0.18	0.16
Binary	29%	0.68	0.32	0.43	0.71	0.34	0.44

Table 10: Correctly classified examples having one, two, three and four labels (Act. = actual labels, Pred. = predicted labels)

No-Labels		Multi-class				Binary			
Act.	Pred.	LibSVM		AdaBoost		LibSVM		AdaBoost	
1	1	169/352	48%	165/352	47%	133/353	38%	132/353	37%
2	1	45/128	35%	44/128	34%	54/146	37%	54/146	37%
2	2	14/128	11%	11/128	9%	14/146	10%	11/146	8%
3	1	2/4	50%	1/4	25%	5/13	38%	5/13	38%
3	2	0/4	0%	1/4	25%	2/13	15%	2/13	15%
3	3	0/4	0%	0/4	0%	0/13	0%	0/13	0%
4	1	N/A	N/A	N/A	N/A	0/1	0%	0/1	0%
4	2	N/A	N/A	N/A	N/A	0/1	0%	0/1	0%
4	3	N/A	N/A	N/A	N/A	0/1	0%	0/1	0%
4	4	N/A	N/A	N/A	N/A	0/1	0%	0/1	0%