

## What's in a URL? Genre Classification from URLs

**Myriam Abramson**

Naval Research Laboratory, Code 5584  
Washington, DC 20375  
myriam.abramson@nrl.navy.mil

**David W. Aha**

Naval Research Laboratory, Code 5514  
Washington, DC 20375  
david.aha@nrl.navy.mil

### Abstract

The importance of URLs in the representation of a document cannot be overstated. Shorthand mnemonics such as “wiki” or “blog” are often embedded in a URL to convey its functional purpose or genre. Other mnemonics have evolved from use (e.g., a Wordpress particle is strongly suggestive of blogs). Can we leverage from this predictive power to induce the genre of a document from the representation of a URL? This paper presents a methodology for webpage genre classification from URLs which, to our knowledge, has not been previously attempted. Experiments using machine learning techniques to evaluate this claim show promising results and a novel algorithm for character n-gram decomposition is provided. Such a capability could be useful to improve personalized search results, disambiguate content, efficiently crawl the Web in search of relevant documents, and construct behavioral profiles from clickstream data without parsing the entire document.

### Introduction

In the infancy of artificial intelligence, a paper entitled “What’s in a link”(Woods 1975) addressed the gap between the representation of knowledge in semantic networks and actual meaning. In contrast, no such gap exists with the representation of a URL in the sense that a URL gives immediate access to the object it represents. Consequently, we can be more ambitious and try to induce some properties of the object from the representation itself. This paper investigates whether the representation of a URL can give clues to some of its possible meanings, namely the genre of the webpage it is representing.

URLs are ubiquitous in Web documents. They are the glue and the fabric of the Web linking disparate documents together. Domain names are an intrinsic part of a URL and are, in some instances, highly prized if mnemonic with respect to a certain usage. As the top-level domain, the suffix of a URL can indicate the high-level hierarchy of a document but is not always predictive of genre where genre is defined by the Free Online Dictionary as “A category ... marked by a distinctive style, form, or content.” Recently, additional suffixes have been added to further par-

tion URLs according to purpose<sup>1</sup>. It is well-known that some URLs are highly indicative of genre. For example, most wikis contain the particle “wiki” or URLs from a certain domain might be dedicated to a certain genre (e.g., Wordpress, Tumblr or blogspot host blogs). Spammers have exploited the term relevance of a URL by stringing together several terms into a long URL (Gyongyi and Garcia-Molina 2005). URL features have been used in genre classification to augment other feature sets of a document (Levering, Cutler, and Yu 2008; Boese 2005). Beyond spam recognition, how much can we leverage from this predictive power to identify the genre of a Web page without referring to the document itself? Such a capability would greatly facilitate our ability to personalize search results, disambiguate content, construct efficient Web crawlers, and construct behavioral profiles from clickstream data without parsing the entire document. In addition, a content-based Web page recommender system could propose similar pages matching a user’s genre interest in addition to topic. For example, a student could be interested in tutorial-style documents on a certain topic.

This paper is organized as follows. We first motivate automated genre classification and contrast it with topic classification. We then discuss feature extraction for genre classification both from the text and URL perspectives and the related work in this area in Section. Our genre classification from URLs methodology is then introduced along with our empirical study and analysis of the results in Section. Finally, we conclude with some discussion of the results and future work.

### Genre Classification

The automated genre classification of webpages is important for the personalization aspects of information retrieval, its accuracy in disambiguating content (word-sense disambiguation according to genre) and the construction of language models. It can also be used for the predictive analysis of Web browsing behavior. Genres are functional categories of information presentation. In other words, genres are a mixture of style, form, and content. For example, books have many genres such as poetry, play, novel,

Copyright © 2012, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

<sup>1</sup><http://www.icann.org/en/announcements/announcement-20jun11-en.htm>

and biography and webpages have also evolved their own genres such as discussion forums, FAQs, blogs, etc. Basically, the genre of a document is tied to its purpose and form. It addresses *how* information is presented rather than *what* information is presented in a document (Rauber and Muller-Kogler 2001). Because of its communication and social aspects, genres rather than topics are more indicative of Web browsing behavior. For example, different professional occupations found different webpage genres useful for their jobs (Crowston, Kwaśnik, and Rubleske 2011). Engineers will access documentation (manual) pages regardless of their respective specialties. Social interaction patterns give rise to a set of different genres accessed together regardless of topics. For example, a researcher might access a submission page to upload a paper and then later a comment page for reviews on the paper (Swales 2004; Tardy 2003). Although genres and content are orthogonal (Eissen and Stein 2004), they do combine in important ways (Dewe, Karlgren, and Bretan 1998; Karlgren and Cutting 1994) and in different proportions depending on the genre itself (Crowston, Kwaśnik, and Rubleske 2011). For example, spam is a combination of content and style. Experiments using only word statistics have shown good results (Kim and Ross 2011) in genre classification and experiments in domain transfer of genre classifiers have shown that genres and topics do overlap (Finn and Kushmerick 2003). It was also shown that just a few words might suffice to categorize the content of a webpage (Koller and Sahami 1997) though this result has not been extended to genre classification.

### Feature Extraction For Genre Classification

Supervised classification tasks rely on the extraction of representative features. Unlike topic classification which is solely concerned with text, genre classification combines different elements. We distinguish below feature extraction from webpages with access to the content of a document and feature extraction from URLs alone.

#### Feature Extraction from Webpages

Stylistic and structural features for genre classification of Web pages can be partitioned according to the following feature sets:

**Syntactic Style features:** number of words (excluding stop words), digit frequency, capitalized word frequencies, number of sentences, average sentence length, average word length,

**Semantic Style features:** frequencies of sentiment words (positive/negative adjectives and adverbs), frequencies of commonly used internet acronyms (e.g., “afaik”, “iirc”).

**Part-of-speech (POS) tags:** frequencies of 36 Penn Treebank part-of-speech tags (Taylor, Marcus, and Santorini 2003).

**Punctuation characters:** frequencies of all 24 punctuation characters.

**Special characters:** frequencies of special characters (e.g., @#\$\$%^&\*+=).

**HTML tags:** frequencies of all 92 HTML 4.01 tags <sup>2</sup>, frequencies of internal links.

**HTML tree features:** average tree width and average tree depth of the HTML structure of a document.

**Function words:** frequencies of 309 function words (e.g. “could”, “because of”) <sup>3</sup>.

These feature sets have been used separately (Finn and Kushmerick 2003) or more frequently in combination (Eissen and Stein 2004; Boese 2005; Santini 2006). A novel contribution in this paper, to our knowledge, are the HTML tree features to represent the layout of a webpage. Other types of features include readability metrics (Boese 2005; Rauber and Muller-Kogler 2001; Kessler, Numberg, and Schutze 1997), visual features (Levering, Cutler, and Yu 2008), word location on a page (Kim and Ross 2011), “erroriness” or noise (Stubbe, Ringlsetter, and Schulz 2007), and character n-grams (Kanaris and Stamatatos 2009; Wu, Markert, and Sharoff 2010; Mason et al. 2010). Character n-grams (sequence of  $n$  characters) are attractive because of their simplicity and because they encapsulate both lexical and stylistic features regardless of language but they were found to be more sensitive to the encoding evolution of webpages (Sharoff, Wu, and Markert 2010).

Feature representativeness is an issue in genre classification because there is no unique characterizing feature or set of features discriminating between genres (Santini 2006; Stubbe, Ringlsetter, and Schulz 2007). Therefore, exporting features to different corpora might be problematic. Moreover, the relevant features depend on the genres to discriminate against (Kim and Ross 2008). For example, the features that distinguish a scientific article from a thesis might be structural (i.e. POS and HTML tags) while the features that distinguish a table of financial statistics from a financial report might be stylistic.

#### Feature Extraction from URLs

The syntactic characteristics of URLs have been fairly stable over the years. URL terms are delimited by punctuation characters and some segmentation is required to determine the implicit words of a domain name. For example, homepage domain names often consist of a concatenation of first name and last name (e.g., “www.barackobama.com”). However, because of the uniqueness requirement of a URL, it is hard to generalize from those terms. A recursive token segmentation approach augmented by stylistic features has produced results comparable to a text approach in a multiclass topic classification task (Kan and Thi 2005). A keyword matching algorithm on common URL lexical terms (e.g., login, search, index) has been used in conjunction with the textual representation of a document in genre classification (Lim, Lee, and Kim 2005). A token-based approach augmented by additional information has achieved high accuracy in identifying suspicious URLs (Ma et al. 2009).

Unlike n-grams for feature extraction from webpages, using n-grams in feature extraction from URLs is less suscep-

<sup>2</sup><http://www.w3schools.com/tags/>

<sup>3</sup><http://www.sequencepublishing.com/academic.html>

tible to evolutionary encoding changes. An *all-ngram* approach combining n-grams of mixed length (4-8) excluding delimiter characters has produced surprisingly good results for webpage multi-label topic classification with binary classifiers (one vs. all) (Baykan et al. 2009). The superiority of n-grams over tokens arises from their relative frequency even in previously unseen URLs (Baykan et al. 2009). Character n-grams have been successfully used also in language identification where combinations of certain characters (e.g. “th” in English, “oi” in French) are specific to certain languages (Baykan, Henzinger, and Weber 2008). An alternative approach is to use character n-grams that would also encapsulate the delimiters. Four and three-character n-grams seem especially suited for URLs because of the length of common suffixes (e.g., “.edu”, “.com”, “.ca”). Word n-grams are often combined with a naive Bayes (NB) classifier approach to produce probability estimates of word compositions but requires a smoothing method to compensate for low frequency counts and unseen transitions (Chen and Goodman 1999). Backoff models (Katz 1987) and linear interpolation smoothing (Jelinek 1980) automatically adjust the length of an n-gram to capture the most significant transitions. This paper introduces a novel algorithm for character n-gram decomposition rather than composition based on linear interpolation and backoff.

Finally, stylistic features of URLs, including the number of delimiters (e.g., forward slashes and punctuation characters) and the average length of particles, can augment URL-based feature sets.

## Methodology

In this section, we introduce our general methodology for (1) acquiring data from the Web and (2) genre classification from URLs. The acquisition of data is an essential part of successful machine learning approaches. Available genre corpora (Santini et al. 2007) are manually constructed with few examples per genre. Moreover, many corpora do not include the associated URL of the webpage. Consequently, our overall technical approach consists of the following steps.

### Open-set Classification

Open-set classification differs from close-set classification in supervised learning when the set of classes is not assumed to cover all examples. We constructed a cascading classifier that could be trained on different available corpora for genre classification based on the stylistic and structural features from webpages outlined above. Cascading classifiers are sequential ensembles of classifiers ordered in some fashion (Alpaydin and Kaynak 1998; Stubbe, Ringlsetter, and Schulz 2007) with a selection scheme. A cascading classifier enables us to boost our initial corpus with an incomplete genre palette and without computing a threshold of acceptance (Fig. 1). Our cascading classifier is composed of binary classifiers, one for each class, with the option of keeping test examples unclassified if not positively identified by any of the binary classifiers. This latest feature is essential for acquiring data from the Web where new genres emerge

each day. Each binary classifier is customized with a feature selection filter (John, Kohavi, and Pfleger 1994). In addition, resampling of the examples to balance the number of positive and negative examples for the binary classifiers makes our cascading classifier agnostic about the class distribution. Several selection schemes are possible. In (Stubbe, Ringlsetter, and Schulz 2007) the binary classifiers are arranged according to their performance in the training set and the first one to indicate a positive class is selected. We obtained better results by selecting the binary classifier with the highest confidence in the positive class. A multi-label selection scheme is also possible with this classifier.

Random webpages and associated URLs were collected using the random webpage generator from Yahoo<sup>4</sup>. The URLs were then classified based on their corresponding webpage content using our cascading classifier.

### Genre Classification from URLs

Our approach for linear interpolation (LI) smoothing of character n-grams consists of combining n-grams (and their subgrams) from a set of most common n-grams of different length found in a corpus. The probability of an n-gram of length  $n$ ,  $P(ngram_n)$ , is computed as follows:

$$\lambda_n I F_j(ngram_n) + \lambda_{n-1} \sum_i^2 I_i F_j(ngram_{n-1}^i) + \dots + \lambda_2 \sum_i^{n-1} I_i F_j(ngram_2^i) \quad (1)$$

where  $I \leftarrow \begin{cases} 1 & \text{if } ngram^i \in \text{most common ngrams} \\ 0 & \text{otherwise} \end{cases}$

$\lambda_n$  are the normalized coefficients of the interpolation and reflect the importance of n-grams of length  $n$  in the prediction of the class.  $F_j(ngram_m^i)$  is the frequency of the  $i$ th n-gram subset of length  $m$  ( $m \leq n \leq 2$ ) for a given class  $j$  in the training set. Finally, the class  $j$  probability is computed as  $(\prod_{i=0}^N P(ngram_i)) Pr(j)$  where  $Pr(j)$  is the class prior probability and  $N$  is the number of top-level n-grams found in a URL string. Our algorithm for linear interpolation and backoff (LIB) in the classification of instances is described in Alg. 1. The backoff procedure stops the decomposition of an n-gram subset. This algorithm is based on breadth-first search and selectively inserts n-grams in a first-in-first-out queue to be decomposed further. The n-grams are extracted on a sliding window of size  $n$  from the URL string and then decomposed when needed. The probabilities of those n-grams are then used with a NB classifier.

The most common n-grams were extracted so that (n-1)-gram subsets were not included unless their counts were at least 5% higher than any subsuming n-grams. Those most common n-grams are the bag-of-words features of our classifiers. The coefficients  $\lambda_n$  were estimated using the information gain attribute selection method (Quinlan 1986) in a

<sup>4</sup><http://random.yahoo.com/bin/ryl>

**Algorithm 1** LIB classification function where  $ngrams$  is a function to parse a string into  $n$ -grams and  $F_j(gram)$  is the frequency of an  $n$ -gram feature for class  $j$  in the training set.

---

```

LIB (instance, n, Classes, priors,  $\lambda$ )=
url  $\leftarrow$  instance.url //string
features  $\leftarrow$  instance.features //ngrams
probs  $\leftarrow$  priors
Q  $\leftarrow$   $\emptyset$ 
FOREACH n-gram  $\in$  ngrams(url, n)
  Q  $\leftarrow$  {n-gram}
  grams  $\leftarrow$   $\emptyset$ 
WHILE Q is not empty
  gram  $\leftarrow$  pop(Q)
  IF gram  $\in$  features //backoff
    grams  $\leftarrow$  grams  $\cup$  {gram}
  ELSE
    m  $\leftarrow$  gram.length - 1
    Q  $\leftarrow$  Q  $\cup$  {ngrams(gram, m)}
  IF grams  $\neq$   $\emptyset$ 
    FOREACH class  $j \in$  Classes
      probs[j]  $\leftarrow$  probs[j]  $\sum_i^{|grams|} \lambda_i F_j(gram_i)$ 
    probs  $\leftarrow$  normalize(probs)
  RETURN argmax $_j$ (probs) //most probable class

```

---

pre-processing step containing the Cartesian product of all  $n$ -grams, their associated sub-grams, and the class.

## Empirical Study

All experiments were conducted in the Weka machine learning workbench (Hall et al. 2009) augmented by our naive Bayes algorithms for Laplace smoothing and linear interpolation. The feature extraction from webpages was done using the open-source Jericho HTML parser (Jericho 2009) and OpenNLP natural language parser (Baldrige and Morton 2004). We compare the LI and LIB approaches with a multinomial NB using Laplace smoothing (with smoothing parameter  $\alpha = 1$ ), a NB with Gaussian smoothing (John and Langley 1995) available in Weka and a support vector machine (SVM) approach (EL-Manzalawy and Honavar 2005) (where  $K=0$  is the linear kernel and  $K=2$  is the radial basis function default kernel), also available in Weka, using the same common  $n$ -grams as bag-of-words features.

Initial experiments were conducted with the 7-genre ‘‘Santini’’ corpus (Santini 2012) consisting of 1400 documents partitioned among 7 genres with 200 examples each. This corpus does not include the associated URL of the document so random pages were classified using a cascading classifier (described above) to acquire URLs of a specific genre. Out of 10000 random webpages, ~25% were unclassified and after validation, 6925 examples were retained. It is worth stressing that unclassified webpages are expected in any random web crawl due to the evolving nature of cyber genres. Other experiments were conducted with the ‘‘Syracuse’’ corpus (Rubleske et al. 2007) consisting of 3025 documents partitioned into 245 ‘‘user-centered’’ genres (e.g, news story, article, how-to page). This corpus includes the asso-

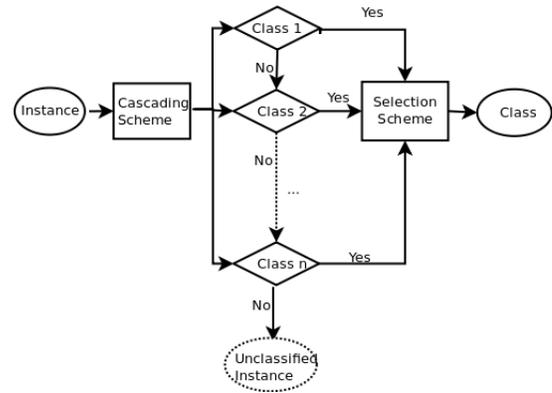


Figure 1: Cascading classifier evaluation framework

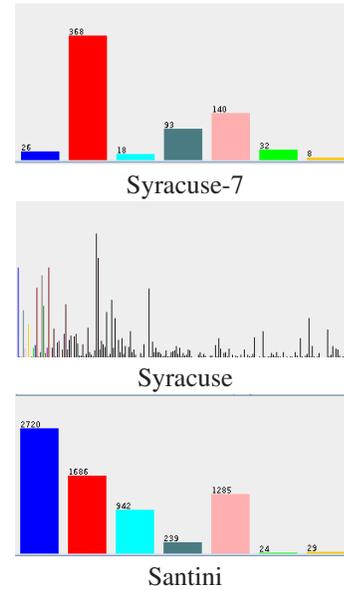


Figure 2: Histogram of the class distribution for the various datasets.

ciated URL of the documents so no data acquisition step was required to obtain them. For comparative purposes, we extracted from this corpus the documents and associated URLs for the genres matching the 7-genre Santini corpus (‘‘Syracuse-7’’) obtaining 685 URLs. Figure 2 illustrates the class distribution of these datasets. Finally, we exported the common  $n$ -grams found in the Syracuse-7 dataset onto the Santini dataset obtaining the Santini/Syr7 dataset.

The coefficients for linear interpolation smoothing using the information gain attribute selection method on character 4-grams, 3-grams, and 2-grams on the different datasets are presented in Table 1. The global coefficients (over all datasets) were used in the experiments. We did not obtain better results for coefficients using the information gain of each  $n$ -gram. The 1000 most common  $n$ -grams from each training dataset, after removing redundant  $n$ -grams (as explained above), were kept as bag-of-words features. The

Table 1: Normalized attribute weights using the information gain attribute selection method for the different datasets.

n-gram length	Syracuse	Syracuse-7	Santini	All
4	0.73	0.55	0.71	0.56
3	0.19	0.34	0.29	0.33
2	0.08	0.11	0.01	0.11

Table 2: The 10 most common unique n-grams per dataset.

Syracuse	Syracuse-7	Santini
22	gle	aa
bus	ards	ser
99	er-	ley
-n	odu	yah
23	erm	.se
er/	on_	m.a
tory	44	ote
id=	55	.wi
m/d	m/20	san
%20	om/2	bc

test sets consisted of those common n-grams found in the training sets. Those 1000 common n-grams were sufficient to populate all URLs in the test sets. The overlap or Jaccard similarity coefficient among common n-grams for the three different datasets was 37% and Tables 2 and 3 illustrate some unique n-grams found.

Table 4 summarizes the results obtained in multi-class classification for the different datasets and the different classifiers using the weighted F1 measure with 10-fold cross-validation. A comparative baseline is provided with a classifier predicting a class at random based on the class distribution during training. Table 5 summarizes results for the multi-class genre classification from webpages for the Syracuse and Syracuse-7 datasets (no ground truth is available for the Santini dataset) using the feature sets described above. Those results show that classification from URLs can give surprisingly better results than classification from webpages for genre classification.

The McNemar’s test (Edwards 1948) was used to evaluate the error rate of the different classifiers. The results do differ depending on the properties of the dataset. LIB does not improve significantly on the performance of Laplace smoothing for the NB classifier validating the independence assumption of a selection of common n-gram features. There is a significant improvement to linear interpolation over all datasets when adding the backoff procedure (LIB). The backoff procedure helps achieve a higher recall by weeding out noisy features and is less prone to overfitting. Backoff provides a lazy feature selection capability on an instance-by-instance basis. NB with Gaussian smoothing does significantly better in the Syracuse dataset maybe because the extrapolation to a normal distribution overcomes the small example-to-class ratios in this dataset. The results of augmented NB with Laplace smoothing and LIB are

Table 3: Top 10 unique common n-grams per genre in the Syracuse-7 dataset

Top 10 unique common n-grams	Genres
.as, boo, k., Ch, lt, gr, 7C, pro, tb, %7	s-page
ss, rs, ho, ks, ph, un, ml	e-shop
sta, ww, qs, ow, dm, mv, ny, /, aq, the	faq
all, ce, l., el, .g, e.c, up, ie, eb, ba	home-page
/0, /1, /2, /200, chi, log, blo, 05, 06, e-	blog
ls, org/, .ed, w.m, edu, .org, /h, u., so, du/	front-page
ruc, con, ru, k/, ty, uct, /w, cl, yc, uc	list

Table 5: Comparative evaluation of genre classification from webpages using the weighted F1-measure metric with 10-fold CV and averaged over 10 iterations.

Dataset	NB	SVM
	Gaussian Smoothing	K=2
Syracuse	0.16±0.002	0.20±0.002
Syracuse-7	0.46±0.005	0.62±0.001

very competitive with SVM (K=2) in the Syracuse-7 and Syracuse datasets possibly because of the relative small example-to-class ratios in those datasets. Table 6 illustrates the differences in precision/recall between the two different classifiers for the Syracuse-7 dataset. Our observations for the other datasets are similar. The SVM classifier achieves its high degree of accuracy by discarding all outlier classes. We also note that SVM is robust with respect to noisy class labels when enough data is provided as in the Santini dataset. The SVM linear kernel performance in the Syracuse and Syracuse-7 dataset indicates that the classes are linearly separable from URL character ngrams. The learning curves for augmented NB with LIB show that additional data will help tame the variance of the classifier (Fig. 3) since the error rate on the test set decreases as the error rate in the training set increases albeit at a slower rate and further work will consist in improving the bias of this classifier. The results in the Santini/Syr7 dataset indicate that the n-gram features of URLs are exportable across corpora with the NB classifier resulting in higher performance for LaPlace and LIB smoothing. Finally, the stylistic features of URLs did not improve to the overall results of n-gram classification of URLs maybe because the punctuation characters were included in the n-grams. In comparison (Table 5), we note that classification from URLs makes obvious mistakes, for example misclassifying a blog with URL “www.questioncopyright.org” as e-shop (because e-shop has a high recall) while classification from webpages misclassified the blog at <http://radio.weblogs.com/0100544/2003/03/22.html> as a homepage maybe because of the presence of an image tag and that the two genres sometimes overlap. Knowing the publisher of a book often helps disambiguate its content (e.g., Tor publishes science-fiction books). Similarly, a document is often ambiguous but when making the document

Table 4: Comparative evaluation of genre classification from URLs using the weighted F1-measure metric with 10-fold CV.

Dataset	Random Classifier	NB Laplace	NB LI	NB LIB	NB Gaussian Smoothing	SVM K=2	SVM K=0
Syracuse	0.02±0.01	0.24±0.02	0.22±0.02	0.24±0.03	0.26±0.03	0.22±0.01	0.28±0.02
Syracuse-7	0.34±0.02	0.66±0.03	0.65±0.06	0.66±0.04	0.64±0.04	0.66±0.03	0.70±0.05
Santini	0.27±0.02	0.36±0.02	0.35±0.01	0.36±0.01	0.32±0.02	0.47±0.02	0.46±0.05
Santini/Syr7	0.27±0.02	0.37±0.01	0.35±0.001	0.37±0.002	0.32±0.02	0.43±0.01	n/a

Table 6: Precision/Recall comparison on the Syracuse-7 dataset with 10-fold CV and averaged over 10 iterations.

Genres	Examples (%)	NB LIB		SVM K=2	
		Precision	Recall	Precision	Recall
s-page	0.04	0.29 ± 0.06	0.22±0.03	0	0
e-shop	0.54	0.76±0.01	0.80±0.01	0.69±0.00	0.96±0.00
faq	0.03	0.23±0.03	0.18±0.02	0	0
home-page	0.13	0.37±0.02	0.4±0.02	0.57±0.01	0.47±0.01
blog	0.20	0.86±0.01	0.79±0.00	0.96±0.01	0.70±0.00
front-page	0.05	0.26±0.02	0.29±0.03	0.1±0.31	0±0.01
list	0.01	0	0	0	0

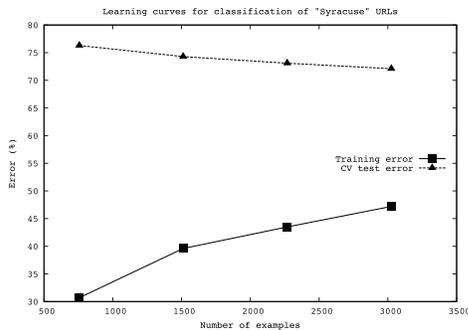


Figure 3: Learning Curves for Naive Bayes with linear interpolation and backoff for the Syracuse dataset.

available online, a categorization, reflected by the URL, is imposed to meet conventions and expectations that help disambiguate its genre in multi-class classification.

## Conclusion And Future Work

The experiments have shown that it is possible to estimate the genre of a document from the URL alone although the task is more difficult for URLs obtained through a random walk with noisy class labels as in the case of the Santini dataset or with a small example-to-genre ratio as in the case of the Syracuse dataset. This prompts questions on the prototypicality of a Web document with respect to its perceived genre and the degree to which this prototypicality also transfers to URLs. Learning from prototypical examples produces more accurate classification models with linear decision boundaries. We have provided a novel algorithm to combine linear interpolation smoothing with backoff for the

classification of URLs in a naive Bayes classifier. This approach compares well with SVM on small-size corpora and with respect to computational performance during training and we will investigate other *all-ngram* models of mixed length covering the entire URL string for genre classification.

In follow-up experiments we will boost our corpora using our cascading classifier to increase the accuracy of our genre classification from URLs approach. We will combine classification from URLs with classification from webpages in a multimodal approach to leverage the strength of both perspectives in order to identify prototypical pages. Finally, we will postulate emerging genres to reduce the number of unclassified webpages obtained from the random walk of a Web crawler.

## References

- Alpaydin, E., and Kaynak, C. 1998. Cascading classifiers. *Kybernetika* 34:369–374.
- Baldrige, J., and Morton, T. 2004. OpenNLP. <http://opennlp.sourceforge.net>.
- Baykan, E.; Henzinger, M.; Marian, L.; and Weber, I. 2009. Purely url-based topic classification. In *Proceedings of the 18th international conference on World wide web, WWW '09*, 1109–1110. New York, NY, USA: ACM.
- Baykan, E.; Henzinger, M.; and Weber, I. 2008. Web page language identification based on urls. *Proc. VLDB Endow.* 1:176–187.
- Boese, E. S. 2005. Stereotyping the web: Genre classification of web documents. Master’s thesis, Colorado State University.
- Chen, S., and Goodman, J. 1999. An empirical study of

- smoothing techniques for language modeling. *Computer Speech & Language* 13(4):359–393.
- Crowston, K.; Kwaśnik, B.; and Rubleske, J. 2011. Problems in the use-centered development of a taxonomy of web genres. In *Genres on the Web, Computational Models and Empirical Studies*. Springer. 69–84.
- Dewe, J.; Karlgren, J.; and Bretan, I. 1998. Assembling a balanced corpus from the internet. In *Proceedings of the 11th Nordic Conference on Computational Linguistics*.
- Edwards, A. 1948. Note on the correction for continuity in testing the significance of the difference between correlated proportions. *Psychometrika* 13(3):185–187.
- Eissen, S. M. Z., and Stein, B. 2004. Genre classification of web pages. In *Proceedings of the 27th German Conference on Artificial Intelligence*, 256–269.
- EL-Manzalawy, Y., and Honavar, V. 2005. *WLSVM: Integrating LibSVM into Weka Environment*. Software available at <http://www.cs.iastate.edu/~yasser/wlsvm>.
- Finn, A., and Kushmerick, N. 2003. Learning to classify documents according to genre. In *International Joint Conference on Artificial Intelligence (IJCAI), Workshop on Computational Approaches to Style Analysis and Synthesis*.
- Gyongyi, Z., and Garcia-Molina, H. 2005. Web spam taxonomy. In *First international workshop on adversarial information retrieval on the web (AIRWeb 2005)*.
- Hall, M.; Frank, E.; Holmes, G.; and Bernhard Pfahringer, Peter Reutemann, I. H. W. 2009. The WEKA data mining software: an update. In *SIGKDD Explorations*, volume 11.
- Jelinek, F. 1980. Interpolated estimation of markov source parameters from sparse data. *Pattern recognition in practice* 381–397.
- Jericho, M. 2009. Jericho html parser. <http://jerichohtml.sourceforge.net>.
- John, G. H., and Langley, P. 1995. Estimating continuous distributions in bayesian classifiers. In *Eleventh Conference on Uncertainty in Artificial Intelligence*, 338–345. San Mateo: Morgan Kaufmann.
- John, G.; Kohavi, R.; and Pfleger, K. 1994. Irrelevant features and the subset selection problem. In *Proceedings of the eleventh international conference on machine learning*, volume 129, 121–129. San Francisco.
- Kan, M.-Y., and Thi, H. O. N. 2005. Fast webpage classification using url features. In *Proceedings of the 14th ACM international conference on Information and knowledge management, CIKM '05*, 325–326. New York, NY, USA: ACM.
- Kanaris, I., and Stamatatos, E. 2009. Learning to recognize webpage genres. *Information Processing & Management* 45(5):499 – 512.
- 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*, 1071–1075.
- Katz, S. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *Acoustics, Speech and Signal Processing, IEEE Transactions on* 35(3):400–401.
- Kessler, B.; Numberg, G.; and Schutze, H. 1997. Automatic detection of text genre. In *Proceedings of the Association of Computational Linguistics*.
- Kim, Y., and Ross, S. 2008. Examining variations of prominent features in genre classification. In *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*, 132.
- Kim, Y., and Ross, S. 2011. Formulating representative features with respect to genre classification. In *Genres on the Web, Computational Models and Empirical Studies*, volume 42. Springer. 129–147.
- Koller, D., and Sahami, M. 1997. Hierarchically classifying documents using very few words. In *Proceedings of the Fourteenth International Conference on Machine Learning ICML*, 170–178. Morgan Kaufmann.
- Levering, R.; Cutler, M.; and Yu, L. 2008. Using visual features for fine-grained genre classification of web pages. In *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*, 131–131. IEEE.
- Lim, C.; Lee, K.; and Kim, G. 2005. Multiple sets of features for automatic genre classification of web documents. *Information processing & management* 41(5):1263–1276.
- Ma, J.; Saul, L.; Savage, S.; and Voelker, G. 2009. Identifying suspicious urls: an application of large-scale online learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 681–688. ACM.
- Mason, J.; Shepherd, M.; Duffy, J.; Keselj, V.; and Watters, C. 2010. An n-gram based approach to multi-labeled web page genre classification. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, 1 –10.
- Quinlan, J. 1986. Induction of decision trees. *Machine learning* 1(1):81–106.
- Rauber, A., and Muller-Kogler, A. 2001. Integrating automatic genre analysis into digital libraries. In *First ACM-IEEE Joint Conference on Digital Libraries*.
- Rubleske, J.; Crowston, K.; Kwaśnik, B. H.; and Chun, Y.-L. 2007. Building a corpus of genre-tagged web pages for an information-access experiment. In *Colloquium on Web Genres, Corpus Linguistics*.
- Santini, M.; Sharoff, S.; Rehm, G.; and Mehler, A. 2007. Web genre wiki. Retrieved from <http://www.webgenrewiki.org>.
- Santini, M. 2006. Some issues in automatic genre classification of web pages. In *JADT Journee Internationales d'Analyse statistique des donnees textuelles*.
- Santini, M. 2012. 7-genre corpus. Retrieved from <http://www.webgenrewiki.org>.
- Sharoff, S.; Wu, Z.; and Markert, K. 2010. The web library of babel: evaluating genre collections. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation*, 3063–3070.
- Stubbe, A.; Ringlstetter, C.; and Schulz, K. U. 2007. Genre

as noise: noise in genre. *Int. J. Doc. Anal. Recognit.* 10:199–209.

Swales, J. M. 2004. *Research Genres: Exploration and Applications*. Cambridge University Press.

Tardy, C. 2003. A genre system view of the funding of academic research. *Written Communication* 20(1):7–36.

Taylor, A.; Marcus, M.; and Santorini, B. 2003. The penn treebank: an overview. *Treebanks: Building and using parsed corpora* 5–22.

Woods, W. A. 1975. What's in a link? In *Representation and Understanding: Studies in Cognitive Science*. 35–82.

Wu, Z.; Markert, K.; and Sharoff, S. 2010. Fine-grained genre classification using structural learning algorithms. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 749–759. Association for Computational Linguistics.