A Comprehensive Study of Features and Algorithms for URL-Based Topic Classification

EDA BAYKAN, Izmir University
MONIKA HENZINGER, University of Vienna
LUDMILA MARIAN, CERN
INGMAR WEBER, Yahoo! Research

Given *only* the URL of a Web page, can we identify its topic? We study this problem in detail by exploring a large number of different feature sets and algorithms on several datasets. We also show that the inherent overlap between topics and the sparsity of the information in URLs makes this a very challenging problem. Web page classification without a page's content is desirable when the content is not available at all, when a classification is needed before obtaining the content, or when classification speed is of utmost importance. For our experiments we used five different corpora comprising a total of about 3 million (URL, classification) pairs. We evaluated several techniques for feature generation and classification algorithms. The individual binary classifiers were then combined via boosting into metabinary classifiers. We achieve typical F-measure values between 80 and 85, and a typical precision of around 86. The precision can be pushed further over 90 while maintaining a typical level of recall between 30 and 40.

Categories and Subject Descriptors: H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing

General Terms: Experimentation

Additional Key Words and Phrases: Topic classification, URL, ODP

ACM Reference Format:
Baykan, E., Henzinger, M., Marian, L., and Weber, I. 2011. A comprehensive study of features and algorithms for URL-based topic classification. ACM Trans. Web 5, 3, Article 15 (July 2011), 29 pages.
DOI = 10.1145/1993053.1993057 http://doi.acm.org/10.1145/1993053.1993057

1. INTRODUCTION

Topic classification of Web pages is normally performed based on the *content* of the pages, with additional clues coming from the link structure of the Web graph [Chakrabarti et al. 1998; Qi and Davison 2006]. However, there are several advantages to attempt the classification task using only URLs, and this is the problem studied in this article.

One advantage of such an approach is speed. The length of a URL is a tiny fraction of the typical length of a Web page. This enables a much faster construction of feature vectors and also speeds up the classification itself, due to the reduced number of nonzero features. But there are also scenarios where the content of a Web page

This is an extended version of a poster published at WWW2009 [Baykan et al. 2009].

Authors' addresses: E. Baykan, Izmir University, Izmir, Turkey; M. Henzinger, Department of Computer Science, University of Vienna, Austria; L. Marian, CERN, Geneva, Switzerland; I. Weber (corresponding author), Yahoo! Research, Barcelona, Spain; email: ingmar@yahoo-inc.com.

DOI 10.1145/1993053.1993057 http://doi.acm.org/10.1145/1993053.1993057

might simply not be available. This happens, for example, in the case of information filtering. An institution might want to employ either a blacklist of topics (e.g., pornography), or a whitelist (e.g., science), and only allow its users access to the corresponding pages [Zhang et al. 2006]. The access should ideally be denied before a page is downloaded. For certain pages the content might also be hidden in images. Here any classification algorithm has only the URL to do its job.

Using only the URL for Web page topic classification also has applications in topic-focused crawlers [Chakrabarti et al. 1999]. If such a system can predict the topic of a hyperlink before downloading the page, it can limit the waste of bandwidth caused by irrelevant pages. Even when the topic classification is not 100% accurate, it will still help to increase the harvest rate of such a system. Another application is in a personalized Web browser. Here, hyperlinks on the page currently viewed by the user could be annotated or highlighted according to the user's topic interests.

Finally, a good classification performance on URLs can be used as a building block in a hybrid classifier, which also uses the content. This is particularly relevant for an on-the-fly classification of Web search results, where only limited content is available and speed is of utmost importance [Chen and Dumais 2000].

Article's contribution

Our contributions are the following: (i) we present a comprehensive experimental evaluation of features and algorithms including boosting, (ii) we show that dictionary-based baselines are not good enough to be high-performance URL-based topic classifiers, though they do achieve high precision, (iii) we analyze the reasons for the confusion between certain classes, (iv) we explore the potential of the use of inlinks, (v) we explain why a token-based URL classification method will not work, (vi) we look at the impact on basic properties such as URL length on the relative difficulty of the task, (vii) we discuss why training on both content and URLs hurts the test performance on URLs and, finally, (viii) we explain in which settings training on dataset X but testing on dataset Y can work.

The rest of this article is organized as follows. In Section 2 we describe the methodology, the feature sets, and the algorithms we used. Then in Section 3 we describe the five different datasets used in our classifiers and in Section 4 we present our results for URL-based topic classification on these datasets. In Section 5 we focus on the Open Directory Project (ODP) dataset and dig much deeper into the matter by providing the reader with insights concerning the classification task. Then in Section 6 we discuss related work, including a recent study with preliminary results for the same problem [Baykan et al. 2009]. Finally in Section 7 we give the conclusions.

2. URL-BASED TOPIC CLASSIFICATION

In the following sections, we describe our basic methodology for the problem of topic classification based on URLs. In Section 2.1, we state the problem definition and explain how we mapped the problem to a binary classification problem. In Section 2.2, we describe the evaluation setup used for our study. The mapping to a binary classification problem and the evaluation setup used is the same as in Baykan et al. [2009]. In Section 2.3 we discuss different ways to map URLs to numerical feature vectors. Finally in Section 2.4 we give the details of the algorithms used in our classifiers.

2.1 Problem Definition

Web page topic classification is the task of assigning a Web page to a predefined topic. The classification can be done with binary or multiway outcomes. In binary classification we want to answer whether a Web page belongs to a topic or not. For example, at the end of the classification the answer is Sports or not-Sports. In the multiway classification we want to find the single topic, out of a potentially large set of candidate topics, the Web page belongs to. For example, at the end of the classification the answer is Sports or Arts. In our study we perform binary classification as it is: (i) an easier task than multiway URL-based topic classification, and (ii) it is easier to add/remove topics, without having to retrain an existing classifier.

2.2 Evaluating the Classifiers

How to evaluate the classifiers

Different publications dealing with classification performance use different evaluation setups and we therefore try to make our setup as clear as possible.

Each classifier was evaluated on all of the test URLs in the corresponding dataset. Precision and recall were then computed for a balanced set of positive and negative samples. Concretely, if there are n_+ positive samples and n_- negative ones, and $p(+|+)$ and $p(-|-)$ are the proportion of correctly classified positive and negative samples, then the recall is $R = p(+|+)$ and the precision is $P = (n_+ p(+|+))/(n_+ p(+|+) + n_-(1 - p(-|-)))$. In a setting where there are k different topics we have $n_- = (k-1)n_+$, and this would give too much weight to the negative test samples, as $n_- >> n_+$ when k is large. Therefore, we downweighted the negative test samples proportionally by setting $n_- = n_+$. This is equivalent to choosing an equal number of positive and negative samples for evaluation. All of P, R, and F (defined shortly) are multiplied by 100 to lie between 0 and 100 and rounded to the nearest integer. Averages are computed on the unrounded numbers and rounded afterwards.

Our main evaluation metric is the F-measure, defined as the harmonic mean of precision and recall. It is sometimes in the literature also referred to as balanced F-score or F_1 score, as it is just one representative of the family $F_\beta = (1+\beta^2)(P \cdot R)/(\beta^2 \cdot P + R)$). Note that in our setting an F-measure of 67 is trivially achieved by always outputting "yes", corresponding to $R = 100$ and $P = 50$. To summarize the performance of all classifiers for a single dataset, such as ODP, we use the macroaverage of the F-measure performances for each topic. We use the macroaverage (= average on a per-topic basis) rather than the microaverage (= average on a per-URL basis) as: (i) the importance of the individual topics might depend on the concrete application, and (ii) the distribution of the sizes of the topics might also vary in different settings.

~~Approximate performance results for the multilabel setting can be obtained by modeling the multilabel decision as a sequence of binary decisions where each URL is passed through 15 different classifiers. It is classified correctly if and only if the correct classifier outputs "yes" (which depends on its recall R or rather on its $p(+|+)$) and all others output "no" (which depends on their $p(-|-)$). Ties where multiple (or no) binary classifiers output "yes" could be broken at random.~~

2.3 Mapping URLs to Features

We experimented with four different methods (plus variants) to extract features from URLs: using tokens, *n*-grams derived from tokens, *n*-grams directly derived from the URL, and explicitly encoding positional information into tokens or *n*-grams. These features were chosen as: (i) they were also used before in related work and (ii) they gave better performance compared to a larger set of features which we used in preliminary experiments explained in Section 4.1.

Tokens. Each URL is lower-cased and split into a sequence of strings of letters at any punctuation marks, numbers, or other nonletter characters. Resulting strings of length less than 2 and "http" are removed, but no stemming was done. We refer to a single valid string as a *token*. `http://watchers.com/Info_3922.asp` would be split into the tokens `watchers`, `com`, `info` and `asp` and represented as a bag-of-words that keeps the counts of the tokens as well as the tokens themselves.

n-grams from tokens. This approach starts with the same tokens as the previous method. That is, a URL is first split into tokens. Then letter n-grams, that is, sequences of exactly n letters, are derived from them, and any token shorter than n characters is kept unchanged. For example, the token `watchers` gives rise to the 5-grams "watch", "atche", "tcher", and "chers", whereas `info` is kept intact for $n = 5$. The main advantage of n-grams over tokens is the capability to detect subwords such as "watch", without requiring an explicit list of valid terms.

n-grams from URL. Here we do not parse the URL into tokens before extracting n-grams but we only remove `http`, punctuation, and numbers and map the URL to lower case. The previous example URL is thus mapped to `watcherscominfoasp` and n-grams are then extracted as explained before. As an advantage, this method can learn information related to neighboring n-grams of tokens. For example, when 5-grams are used, it can distinguish between the `news` token in `news.com` with `newsc` and in `sports.com/news` with `mnews`.

Encoding positional information. Although the aforesaid n-gram approach can pick up positional information, we also experimented with explicitly encoding this information. To this end, we duplicated each n-gram (or token) and appended its position in the URL to it. So the 3-grams derived from tokens for `www.epfl.ch` are `www`, `www_1`, `epf`, `epf_2`, `pfl`, `pfl_2`, `ch_`, `ch_3`. The machine learning algorithm will not see the actual number but the derived feature vector will have a different dimension for "epf_1" than for "epf_2". We also tried different variants of this and, for example, used the level of subdirectory as a depth for each n-gram (or token). All these variants gave very similar results.

Note that for all approaches the dimensionality of the feature vectors depends on the training set. Any token or n-gram which is seen only in the test set will be ignored. This happens very frequently for tokens, leading to a poor performance as will be discussed in detail in Section 5.5.

We also experimented with combining different features, such as using both 4-grams and tokens. The best-performing feature set was a combination of 4-, 5-, 6-, 7-, and 8-grams. This combination we call *all-grams*. In Section 4.1 we will explain in detail how we chose the feature set having the best performance.

2.4 Machine Learning Algorithms Used

In our experiments, we used the following four machine learning algorithms because they were also used in related work, and they gave better performance compared to the other algorithms which we used in preliminary experiments and which are described at the end of this section.

- *Naïve Bayes (NB).* This simple algorithm assumes conditional statistical independence of the individual features given the topic. It then applies the maximum likelihood principle to find the topic which is most likely to generate the observed feature vector.
- *Support Vector Machines (SVM).* The underlying idea is to map the training points to a higher-dimensional vector space and then to find a hyperplane separating most of the positive and negative training points. For efficiency the algorithm avoids ever computing this mapping explicitly by using kernel functions.
- *Maximum Entropy (ME).* The idea behind this approach is to find a distribution over the observed features which explains the observed data but which also tries to maximize the entropy, or uncertainty, in this distribution. This results in a constrained

optimization problem which is then solved using an iterative scaling approach. Details about this scheme can be found in Nigam et al. [1999].

— *Boosting*. Boosting is actually a *meta*-algorithm which tries to combine the output of several (poor) classifiers into a single (good) classifier [Freud and Schapire 1996].

For the Naïve Bayes and Maximum Entropy classifiers we used the Bow Toolkit [McCallum 1996]. For the Support Vector Machine algorithm, we used SVM^{light} [Joachims 1999]. For the implementations of the boosting algorithms, we used both Weka [Witten and Frank 2005] and Matlab[1]. In all cases, we used the default setting of the software packages, but we did preliminary experiments with other settings, such as other kernel functions (SVM) or different numbers of iterations (ME). For SVM, we also tested different ways of weighting feature dimensions. The differences were usually small, but a standard log-log tf-idf weighting gave the best results[2]. The SVM parameters were the default settings of SVM^{light}; see http://svmlight.joachims.org/. As NB and ME are built for discrete, probabilistic models, we used the raw feature counts without any weighting.

We also experimented with the k-nearest neighbor algorithm [Hastie et al. 2001] (trying different values of k between 1 and 30) and with the Rocchio classification algorithm [Manning et al. 2008]. As both of these gave inferior results, we do not report their performance.

3. DATASETS

We used five different datasets: the Open Directory Project[3], the Yahoo! Directory[4], categorized URLs from the Wikipedia[5], pages with topic-specific tags from Delicious[6], and topic-specific search results from Google[7]. The data was downloaded during the second half of 2008. The sizes of these datasets can be found in Table I. In all cases we removed (the small number of) URLs listed in multiple topics of the same dataset, which was between 0.6% and 3% of all unique URLs. The datasets themselves will be explained in detail in the following sections. The differences in the definition of certain topics, for example, what constitutes Web pages listed under "Arts", are discussed in Section 5.8.

3.1 Open Directory Project

The Open Directory Project, or simply ODP, is a human-edited directory of classified Web pages. URLs of Web pages are filed in a topic hierarchy, along with short human-edited summaries, so-called "snippets." When discussing results for the use of snippets we always used ODP, and Section 5.7 refers exclusively to this dataset. Regarding topics, we used Web pages listed under the fifteen main topics "Adult", "Arts", "Business", "Computers", "Games", "Health", "Home", "Kids and Teens", "News", "Recreation", "Reference", "Science", "Shopping", "Society", and "Sports". The additional "World" category, containing only foreign-language Web pages, was not used, but the "International" subtopic of "Kids and Teens" was kept. As the ODP is widely used in the literature to evaluate classification algorithms [Qi and Davison 2009], we focused our in-depth study on this dataset.

[1] http://research.graphicon.ru/machine-learning/gml-adaboost-matlab-toolbox.html

[2] The exact weighting formula was $w_{i,j} = (1 + \ln(f_{i,j})) \cdot \ln(n/n_i)$, where $f_{i,j}$ is the occurrence count of term/n-gram i in URL j and n_i is the number of URLs containing term/n-gram i.

[3] http://www.dmoz.org/

[4] http://dir.yahoo.com/

[5] http://en.wikipedia.org/

[6] http://delicious.com/

[7] http://www.google.com

3.2 Yahoo! Directory

Just as the ODP, the Yahoo! Directory contains human-classified Web pages. However, their chosen topic hierarchy is not perfectly aligned with that in the ODP and, for example, the Yahoo! Directory contains the category “Government”, but lacks the category “Adult”. From the Yahoo! Directory we used the fourteen main categories “Arts and Humanities”, “Business and Economy”, “Computers and Internet”, “Education”, “Entertainment”, “Government”, “Health”, “News and Media”, “Recreation and Sports”, “Reference”, “Science”, “Social Science”, “Society and Culture”, but we ignored the “New Additions” category.

3.3 Wikipedia

A typical article in Wikipedia comes with a list of “external links” at the end of the article. For example, `http://en.wikipedia.org/wiki/Support_vector_machine` lists the URL `http://svmlight.joachims.org/` as an external link. We used such external links as the URLs for our classification task.

The categorization of such URLs was obtained by using the “Categories” at the end of most articles. For example, `http://en.wikipedia.org/wiki/Support_vector_machine` is categorized as “Machine learning” (among others). To find out that “Machine learning” belongs to “Science”, we use Wikipedia’s category hierarchy. For example, this hierarchy contains the path “Machine learning” → “Artificial Intelligence” → “Computer science” → “Formal sciences” → “Science”, pointing from a subcategory to a parent category. This makes it possible to extract a list of categorized URLs for most of the topics listed in the ODP. Here we used Wikipedia’s “Childhood” category as the analog of “Kids and Teens” and similarly paired “Computing” with “Computers” and “Retailers” with “Shopping”. As the Wikipedia is virtually pornography free, the “Adult” class was not used. All other classes (“Arts”, “Business”, etc.) were mapped to the category with the same name in Wikipedia.

However, in doing so one must take a certain amount of care. Apart from existing cycles in the hierarchy [Zesch and Gurevych 2007], there are also several cases where the transitivity of the hierarchy leads to unexpected categorization results, such as “Princeton University”→“Ivy League”→“College athletics conferences”→“Sports leagues”→“Team sports”→“Sports.” This example shows that all external links listed on the article about Princeton University would be classified as “Sports”, which is incorrect for the majority of such links. To avoid this problem, we limited the depth of the hierarchy from the topic-specific root (in this case “Sports”) to four, and would no longer list “Princeton University” under “Sports”. Furthermore, the “Society” class was dropped, as it contained subcategories related to all possible aspects of human culture.

3.4 Google

Combining the names of fourteen main ODP categories (all except “Adult”) with the names of their subcategories, we generated a list of queries such as “sports baseball” or “science physics” to send to Google.[8] The top 1,000 search results for these queries were then treated as belonging to the corresponding topic. Sponsored links and results from `google.com` or variants such as `google.ch`, as well as results from `dir.yahoo.com`, were removed. The sponsored advertisement links were not part of the organic search results and the results returned from the Google Directory, which has identical content to ODP, might have inflated the results, as these results contain the answer (= the topic classification) explicitly in the URL. For this dataset

[8] We queried `http://www.google.ch` with a Firefox user-agent and a list of both Swiss and international proxy servers.

our experiments implicitly address the question: “Given only a URL, which search engine query would it be returned for?”

3.5 Delicious

Delicious is a popular social bookmarking site which allows users to add tags to describe the bookmarked Web pages. Here, we considered a page tagged with one of the 15 ODP category names as belonging to the corresponding topic. However, to avoid noise caused by rare uses of tags, we only used pages where the tag under consideration was used at least 30% as frequently as the most prominent tag for that page. In the literature tag prediction of Web pages is studied a lot. We refer the readers to Alex et al. [2007], Heymann et al. [2008], and Jäschke et al. [2007] for some of the recent work in tag prediction.

Additionally, we used search results on Delicious for pages tagged with both a main and a subcategory name, as described for the Google dataset described earlier. Note that for this dataset our experiments essentially try to answer the question: “Given only a URL, what would be an appropriate tag to apply?”

3.6 Construction of the Training and the Test Datasets

For each topic in each dataset a random subset of URLs was put aside as a test set, and the remaining URLs were used for training. We used a single split of the data into training and test. The sizes of the test sets were the same for each topic, but differed among datasets. The test set size for each topic was 1k for ODP, 1k for Yahoo!, 200 for Wikipedia, 500 for Delicious, and 500 for Google. We used the same number of URLs for each topic so that no topic would dominate the final precision and recall numbers. For example, the smallest topic (News) in ODP had only 8k data available and because of this we picked 1k per topic in ODP as test size. For the other datasets the same logic applies while determining the size of test data. If we did conventional splits it would be possible that topics would not have been represented with the same amount of test data.

The training sizes for all datasets can be found in Table I. For training we combined the same number of positive (e.g., “Sports”) and negative (e.g., “not-Sports”) training URLs, where the negative ones were equally distributed among the not-* classes. So to train an ODP Sports classifier, we used all of the 103k URLs from the Sports training set (see Table I) and 103k/14 random URLs from each of the other fourteen topics. This way, algorithms give an equal weight to positive and negative training samples, and even if one topic among the negative classes is very large, for example, Arts, we still train a Sports versus not-Sports classifier, rather than a Sports versus Arts classifier. If in a dataset the largest of the topic-specific training sets are more than a factor 14 larger than the smallest ones, we had to “fill up” the missing not-* samples. That is, we always kept a balanced split between positive and negative samples for training, but for the very large topics, some small topics were underrepresented in the negative samples. Although it is unlikely that this significantly affects the performance, it would only lead to *lower* numbers, as we used the same number of test URLs for each of the topics. For example, when training a binary classifier for the Arts category of ODP with 268k positive training URLs, the negative subclass News has only 7.5k training URLs and so the gap to 268k/14=19.1k has to be filled up with other non-News URLs.

4. EXPERIMENTAL RESULTS

First in Section 4.1 we show which feature set and machine learning algorithm gives the best performance on the ODP dataset. Then in Section 4.2 and in Section 4.3 we

Table I. The Sizes of Training Set for Each Topic in All Datasets

	ODP	Yahoo!	Wikipedia	Delicious	Google
Topic	Size of training set				
Adult	36k	-	-	525	-
Arts	268k	31k	257k	31k	2.7k
Busin. (& Econ.)	240k	118k	42k	20k	5.3k
Comp. (& Intern.)	119k	8.8k	38k	18k	9.2k
Education	-	1.8k	-	-	-
Entertainment	-	48k	-	-	-
Games	57k	-	16k	6.0k	4.3k
Government	-	9.7k	-	-	-
Health	62k	14k	9.6k	19k	4.7k
Home	29k	-	4.2k	6.9k	5.2k
Kids & Teens	37k	-	1.6k	2.2k	853
News (& Media)	7.5k	4.7k	280	10k	1.6k
Recreation	108k	44k	46k	2.4k	3.9k
Reference	56k	919	24k	14k	2.9k
Science	100k	26k	160k	17k	2.6k
Shopping	100k	-	4.1k	16k	914
Social Science	-	5.5k	-	-	-
Society (& Cult.)	241k	26k	-	18k	4.2k
Sports	103k	-	73k	15k	15.8k
TOTAL	1.6m	338k	676k	196k	64k

nineteen different topics

give the results of the best combination of algorithm and feature set on each dataset. We tried two approaches in our experiments: (1) we trained on each dataset separately and tested on the model which is trained on the same dataset (Section 4.2), (2) we trained on ODP and tested on the other datasets (Section 4.3).

4.1 Which Feature and Algorithm Gives the Best Performance

In this section we explain how the different features and the machine learning algorithms perform. Our target is to identify the combination of feature and machine learning algorithm giving the best performance on one dataset and then test that combination on the other datasets.

To understand the best feature set we performed preliminary experiments on the ODP dataset with a fixed machine learning algorithm (Naïve Bayes) using only tokens, only n-grams from 3 to 8, and combinations of n-grams. In Table II we present the results for these experiments. Tokens had an F-measure of 76 and 6-grams alone had the best performance among single n-grams with an F-measure of 81. We got the best performance with an F-measure of 82, for the combination of 4-, 5-, 6-, 7-, and 8-grams, which we call as all-grams. However, as explained in Section 2.3 all-grams can be derived either from tokens or from the URL directly. The latter gave slightly better results. Explicitly encoding positional information for *n*-grams derived from tokens performed slightly better than using only such *n*-grams without positional information, but still slightly worse than using *n*-grams from the URL. Generally speaking the differences between using only 4-grams, all-grams, combinations with positional information, and deriving the features from tokens or from the URL directly were very small and typically not more than 1 point in precision, recall, or F-measure for any of the 15 classes. However, using tokens directly resulted in a macroaveraged F-measure

Table II. Macroaveraged F-Measure Values of Various Features with Naïve Bayes

Feature	*F*
Tokens	**76**
3-grams	**75**
4-grams	**80**
6-grams	**81**
8-grams	**79**
4-5grams	**81**
4-5-6grams	**81**
4-5-6-7grams	**81**
4-5-6-7-8grams	**82**

For training and testing the ODP dataset was used.

of merely 76 on ODP, compared to 83 when using all-grams directly derived from the URL. As generally all-grams directly derived from the URL performed best (by a small margin) we chose this feature set as our default one.

However, the choice of features (in particular tokens versus all-grams) had a much bigger influence than the choice of algorithm. When using all-grams directly derived from the URL, the macroaveraged F-measure on the ODP was 82.3, 82.8, and 82.8, for NB, SVM, and ME, respectively, with an absolute difference of 0.5, whereas the difference between the use of tokens and all-grams directly derived from the URL was 7 points in F-measure for ME (75.9 versus 82.8). As ME and SVM had the same macroaveraged performance and as ME was considerably faster to run, we chose ME as our default algorithm.

Although NB did not give the best performance among the algorithms we experimented with, it showed a performance close to ME. Due to the simplicity of Naïve Bayes Algorithm and due to the nonindependent and overlapping features, it is possible to find this result quite surprising. In the literature [McCallum and Nigam 1998; Domingos and Pazzani 1997] it is already shown that NB works well with nonindependent features. The reason of the success of NB is explained with the paradox which states that NB estimates the sign of the classification (in binary cases) with great accuracy while it can estimate the classification function poorly [McCallum and Nigam 1998].

For all-grams, we also tried feature selection using information gain as a selection criterion [Cover and Thomas 1991]. Information gain selects features which reveal the most information about the classes. We experimented with the features having the highest information gain values. Here the goal was to find a subset of actual n-grams such as "por" or "oalke" to use rather than a class of n-gram such as 4-grams versus 8-grams. However, none of the subset of all-grams with the highest information gain values gave an improvement in macroaveraged F-measure over using all the all-grams. Getting no improvement with this approach is not surprising as using a subset is akin to using tokens and reintroduces the problem of "empty" URLs (see Section 5.5). However, a similar approach was tried to construct a dictionary of all-grams for each topic (see Section 5.2) and this led to a high-precision but only moderate-recall classifier.

Furthermore, we experimented briefly with weighting different parts of a URL differently, for example, giving more (or less) weight to *n*-grams derived from the host name. Similarly, we tried to split URLs corresponding to dynamically generated content into key-value pairs and then only work with the keys (or only the

Table III. Performance in Terms of F-Measure for Binary ME Models Trained Using All-Grams Directly Derived from the URLs

	ODP			Yahoo!			Wiki			Delic.			Google		
Topic	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
Adult	90	85	88	-	-	-	-	-	-	73	71	72	-	-	-
Arts	81	83	82	77	80	78	84	90	87	79	82	81	87	88	87
Busin. (& Econ.)	80	84	82	80	84	82	78	87	82	80	77	78	89	85	87
Comp. (& Intern.)	82	80	81	79	78	79	90	93	91	81	82	81	87	87	87
Education	-	-	-	74	77	75	-	-	-	-	-	-	-	-	-
Entertainment	-	-	-	85	81	83	-	-	-	-	-	-	-	-	-
Games	89	88	88	-	-	-	93	90	91	80	76	78	91	86	88
Government	-	-	-	81	79	80	-	-	-	-	-	-	-	-	-
Health	85	79	82	83	81	82	78	81	79	86	85	85	90	86	88
Home	86	80	83	-	-	-	87	85	86	77	76	76	89	86	88
Kids & Teens	82	79	80	-	-	-	78	78	78	69	74	71	86	81	84
News (& Media)	81	82	81	77	78	77	74	81	77	78	78	78	83	83	83
Recreation	80	82	81	83	80	81	86	86	86	78	78	78	89	83	86
Reference	87	83	85	75	80	77	84	83	84	79	78	79	88	84	86
Science	83	81	82	80	81	80	91	89	90	83	83	83	87	83	85
Shopping	79	82	81	-	-	-	86	91	88	83	85	84	88	82	85
Social Science	-	-	-	74	75	74	-	-	-	-	-	-	-	-	-
Society (& Cult.)	81	81	81	76	77	76	-	-	-	80	81	80	87	85	86
Sports	87	84	85	-	-	-	92	88	90	86	84	85	91	91	91
Average	83	82	83	79	79	79	85	86	85	79	80	79	88	85	87

values) [Dasgupta et al. 2008]. None of these approaches led to any performance improvement.

4.2 Performance by Dataset

How well can one classify URLs into topics, using a good feature set (all-grams directly derived from the URL, see Section 2.3) and a good machine learning algorithm (ME, see Section 2.4)? The results for this out-of-the-box approach are presented in Table III. The typical range of F-measure values are between 79 and 87, with Yahoo! and Delicious as the most difficult datasets, and the Google search engine results as the easiest.

The reason that the classification works best for our Google dataset is that the URLs in the top 1,000 search results often contain one or all of the search terms. This situation may be due to the features used in search engines like the text similarity of URL and the search engine query. A recent patent [Poola and Ramanujapuram 2007] presents techniques for tokenizing URLs and extracting keywords from URLs which can be used by search engines. In fact, even a simple dictionary-based method (see Token by-Hand Dictionary in Section 5.2) achieves an F-measure of 74 on the Google dataset, but only 46 on the ODP dataset. For the same reason the “Sports” class is easier than most of the other classes. See Table VI for the per-topic performance of baseline methods on the ODP dataset.

On the ODP dataset “Adult” has the highest F-measure of 88 which is in harmony with the results for human performance (Section 5.1) and dictionary-based techniques (Section 5.2) all having higher F-measure than the macroaveraged F-measure for this topic. “Games”, “Sports”, and “References” also have an F-measure higher than the macroaveraged F-measure. For most of the topics recall is lower than precision. For

Table IV. Performance when Binary ME Classifiers are Trained on the ODP Dataset Using All-Grams, and are Tested on the Other Four Datasets

	Yahoo!			Wiki			Delic.			Google		
Topic	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
Adult	-	-	-	-	-	-	79	60	68	-	-	-
Arts	80	56	66	65	77	70	63	54	58	79	71	74
Busin. (& Econ.)	76	71	73	61	30	40	53	45	49	75	62	68
Comp. (& Intern.)	76	85	80	73	86	79	58	84	68	66	80	73
Games	-	-	-	88	76	81	83	62	71	90	77	83
Health	82	83	82	64	34	44	76	71	73	79	79	79
Home	-	-	-	53	16	25	69	48	56	73	67	70
Kids & Teens	-	-	-	43	31	36	57	47	52	68	68	68
News (& Media)	76	41	53	64	57	60	63	65	64	70	75	72
Recreation	64	36	46	59	28	38	73	61	67	82	60	69
Reference	80	66	73	50	24	32	70	45	55	64	60	62
Science	78	67	72	66	81	73	70	76	73	71	83	77
Shopping	-	-	-	64	27	38	63	71	67	73	59	65
Society (& Cult.)	65	51	57	-	-	-	62	62	62	70	65	67
Sports	-	-	-	85	60	70	83	63	71	90	77	83
Average	75	62	67	64	48	53	68	61	64	75	70	72

The similarities and differences between Yahoo! and ODP are explored in depth in Section 5.8.

the other datasets who have “Sports” as a topic, it is the topic having the highest F-measure.

4.3 Train on Apples, Test on Oranges

In practice, one might want to train a model on one dataset, but then test it on another with very different characteristics. For example, the URLs encountered during a large Web crawl will most likely be very different from the URLs in the ODP, but before starting the crawl only the latter will be available to train a classifier. We investigated this type of scenario by training models on the ODP dataset and then evaluating their performance on the test sets of other datasets. Table IV shows the performance for this setting.

Generally, the performance is a lot worse than when for each topic dataset-specific classifiers are trained. Only for the Google dataset the F-measure values are higher than 66, which could be obtained by a trivial classifier which always outputs “yes.” The main reason for the significantly lower performance is data inconsistency. For example, different directories use different definitions of what constitutes the “Society” topic. This is discussed in detail in Section 5.8.

5. DIGGING DEEPER

In this section: (i) we give the results of an experiment with human judges to get an intuition about the inherent difficulty of the classification task (Section 5.1), (ii) we show that dictionary-based baselines are not good enough to be high-performance URL-based topic classifiers, though they do achieve high precision (Section 5.2), (iii) we analyze the reasons for the confusion between certain classes (Section 5.3), (iv) we explore the potential of the use of inlinks (Section 5.4), (v) we explain why a token-based URL classification method will not work (Section 5.5), (vi) we look at the impact on basic properties such as URL length on the relative difficulty of the task (Section 5.6),

(vii) we discuss why training on both content and URLs *hurts* the test performance on URLs (Section 5.7), (viii) we explain in which settings training on dataset X but testing on dataset Y can work (Section 5.8) and, finally, (ix) we demonstrate the potential of applying boosting to combine several algorithms (Section 5.9). These in-depth experiments (except boosting and dictionaries) are only done for the ODP dataset as in the literature it is more widely used than the others for the classification task [Qi and Davison 2009].

5.1 Human Performance

Humans are very good at isolating the single content-bearing feature such as `books` in `http://www.nowtoronto.com/issues/19/19/Ent/books.html` (which is listed under "Literature" and hence "Arts" in ODP), which could arguably be difficult for a machine. On the other hand, humans will generally not be very familiar with the domains present in ODP. For example, a human might not know that `http://english.pravda.ru/` is a news portal site whereas a machine learning algorithm will probably have seen this domain already in the training set.

To get an intuition about the inherent difficulty of the classification task two human judges, who are coauthors of this article, classified a set of ODP URLs, 100 per topic presented in random order mixed between topics. Both of the human judges have expertise in information retrieval. The two human judges performed a multiway classification and assigned a unique class per URL. In cases where the topic did not seem obvious to them, they had the option to label the URL as "don't know". Their multiway classification was then mapped to a binary classification in the obvious way. In the "guessing" setting (see Table V) URLs labeled as "don't know" were randomly labeled as "yes" or "no" with a 50-50 probability for each of the 15 topics and a single such URL could accumulate multiple labels. In the "strict" setting such URLs were always labeled as "no" for all topics. Afterwards their classifications were OR-ed by correctly labeling a URL (in the binary setting) if and only if at least one of the judges labeled it correctly, possibly via a random guess. As can be seen by comparing Table III and V, two human judges performed considerably worse in terms of F-measure, though they generally obtained very high precision, at least in the "strict" setting.

Note that both of the topics "Shopping" and "Society" have a low fraction of "don't know" URLs but the overall performance of two human judges is still low for these topics. The reason for this phenomenon appears to be that humans do not have a very clear intuitive notion of these broad topics. They tend to classify an online pharmacy as "Health" instead of "Shopping" or a list of chronological World War II events as "Reference" rather than "Society." To verify the hypothesis that these misinterpretations are not URL dependent, it would be interesting to repeat the task with the actual content of a page. Furthermore, one could try to improve the human performance by "training" them better for the task, explaining the differences between the classes with example URLs/pages. Note that due to the very small scale of this study (two human judges and 2 x 1,500 URLs) one should not overgeneralize the actual numbers. Our motivation was mainly to get a better intuition for the problem.

5.2 Dictionaries and String Matching

How would a human infer topic "Sports" from the URL `http://www.attackvolleyballclub.net/`? Most likely she would recognize the indicator word "volleyball." This simple observation led us to explore the use of a dictionary of indicator words for each topic. For example, a URL is classified as "Sports" if and only if it contains a token from the sports dictionary. We tried both token and substring

Table V. Human Performance on the URL Classification Task

	Guessing			Strict			Don't
	P	R	F	P	R	F	know
Adult	95	78	85	99	73	84	4
Arts	92	66	76	97	46	63	18
Business	91	55	68	95	32	48	21
Computers	93	68	78	97	44	61	20
Games	95	63	76	100	42	59	17
Health	95	70	80	99	52	68	18
Home	93	58	71	99	41	58	6
Kids&Teens	91	51	65	98	28	44	13
News	94	67	78	98	47	63	11
Recreation	90	46	61	96	12	22	22
Reference	87	37	52	94	16	27	14
Science	89	66	76	94	45	61	10
Shopping	92	46	61	98	25	40	9
Society	91	53	67	98	33	49	8
Sports	96	76	85	99	53	69	17
Average	92	60	72	97	39	54	14

"Guessing" refers to the setting where "don't know" labels were randomly assigned to "yes" or "no" decisions for each binary classifier. For "Strict" these cases were always counted as "no" decisions. In both cases the two judges were combined by OR-ing as described in the text. The last column gives the percentage of URLs which both judges labeled as "don't know".

match of a word from the dictionary of indicator words in the URL as definitions of "contains." All dictionaries were constructed from the corresponding training set.

We formed four types of dictionaries which contain a list of representative tokens or all-grams derived from URLs for topics. In the first type of dictionary, *Tokens by-Hand*, we used all words from the first two levels of the ODP hierarchy. For example, the terms "Basketball" and "Football" which are listed one level below "Sports" in ODP hierarchy, are added to the sports dictionary. Some terms, such as "Online" listed under "Games," are not put into the games dictionary if they appeared nontopic-specific to a human. The average dictionary size was 19.8 words per topic. For the second type of dictionary *Tokens by-Statistics* we formed a list of tokens which have length greater than 2, by first merging the "set-of-tokens" from URLs of each Web page listed under ODP for each topic. Set-of-tokens is simply the list of tokens seen in the URL.

Then we obtained representative tokens for a particular topic, by looking at the percentage of "set-of-tokens" containing this token, both for the topic itself and for the other topics. A token was added to the dictionary corresponding to the topic if: (i) it appeared in at least five "set-of-tokens" of the topic, (ii) it had a precision of at least 80 on these "set-of-tokens", and (iii) it had a recall of at least .01. The average dictionary size for this approach was 987 words[9]. We decided on these rules after manually inspecting the list of tokens and we found tokens with a recall above 0.01% and a precision of 80% or more to be tokens that humans might choose as "topically relevant". In Tokens by-Statistics dictionary "basketball" and "mensbasketball" are some example tokens for Sports topic.

[9] We used an equal weight for positive and negative topic samples during dictionary construction.

Table VI. Baseline Results for Using Dictionaries with Token Match on ODP

	Tokens						Domains			All-grams		
	by-Hand			by-Statistics			by-Statistics			by-Statistics		
Topic	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
Adult	99	10	18	94	50	65	99	26	41	67	90	77
Arts	79	3	6	91	37	53	94	27	43	70	75	82
Busin.	65	1	1	91	3	6	92	1	2	78	57	66
Comp.	81	2	4	92	28	43	97	13	23	74	70	72
Games	96	4	7	92	61	73	97	46	63	66	89	76
Health	97	4	8	91	30	45	96	16	28	68	79	73
Home	76	4	7	93	50	65	96	34	50	66	85	74
Kids	82	1	2	89	44	59	94	30	45	69	80	74
News	85	5	9	89	18	30	91	7	14	73	74	73
Recreat.	58	1	2	93	13	23	86	6	12	75	67	71
Refer.	80	4	8	89	60	72	93	44	60	73	85	79
Science	92	5	10	88	54	67	91	39	55	65	80	72
Shopp.	66	1	1	93	4	8	96	1	2	76	65	70
Society	73	3	5	94	24	38	95	16	27	71	65	68
Sports	97	12	21	95	33	49	97	20	34	68	84	75
Average	82	4	7	92	34	46	94	22	33	71	76	73

The third type of token dictionary is referred to as *Domains by-Statistics*. For this the list of typical domains for each topic is formed in the same way as representative tokens are formed for dictionary Tokens by-Statistics[10]. The difference is now "set-of-tokens" contains only domains of URLs. With the Domains by-Statistics dictionary a Web page is classified as "Sports" if and only if it comes from one of the typical sports domains. In Domains by-Statistics dictionary "football.co.uk" and "sportsnetwork.com" are some example domains for Sports topic. Finally, we also trained a dictionary by using all-grams instead of tokens named *All-grams by-Statistics*, which resulted in an average dictionary size of 13k *n*-grams. In All-grams by-Statistics dictionary "sports" and "sportspa" are some example grams for Sports topic.

Table VI shows the performance of the baseline algorithms when the tokens (or all-grams) from the dictionary are checked with token match strategy with the tokens (or all-grams) derived from the test URLs. For all the topics, all the token dictionaries gave fairly high precision values but low recall values. The Tokens by-Hand dictionary gave the lowest recall values as their vocabulary for each topic is limited to the two first levels of ODP hierarchy. The Domains by-Statistics dictionary seems to give higher precision values than the other dictionaries. However, it has a macroaveraged recall of 22. This shows that domains are indeed good indicators but using only domain information will not be enough to achieve an adequate level of recall. When we compare the performances of all dictionaries we see that the statistical dictionaries give higher performances. We had two types of statistical dictionaries, Tokens by-Statistics and All-grams by-Statistics. The Tokens by-Statistics dictionary achieved a higher precision but a much lower recall than the All-grams by-Statistics dictionary. On the ODP dataset All-grams by-Statistics gives the highest performance in terms of F-measure with a macroaveraged F-measure of 73.

In Table VII we present the All-grams by-Statistics dictionary performance on the other datasets, for example, when a dictionary is trained on Yahoo! and is also tested on

[10] For example, the "domain" of `http://ltaa.epfl.ch` is `epfl.ch`.

Table VII. Performance of All-Grams by-Statistics Dictionary with Token Match on Other Datasets

Data set	P	R	F
Yahoo!	71	70	70
Wiki	65	84	73
Delicious	69	74	70
Google	69	84	76

Each dictionary was trained for the respective dataset.

Yahoo!. On all datasets we had a macroaveraged F-measure of around 70, Google having the highest F-measure of 76. For all the datasets with the All-grams by-Statistics dictionary we observed a higher recall than the precision. This is due to all-grams leading to an increase in recall with a decrease in precision.

For the first two token-based dictionaries we also experimented with using substring matches, rather than token matches on the ODP dataset. This increases recall as now the "volleyball" in `http://www.attackvolleyballclub.net/` is also detected. The macroaverages in this case are P = 84, R = 15, F = 24 for the Tokens by-Hand dictionary and P = 70, R = 63 and F = 62 for the Tokens by-Statistics dictionary[11].

From these results we can conclude that baseline algorithms are not enough for a good performance. However, results for baseline algorithms give us an insight about which topics are easier to classify. For example "Adult", "Sports", and "Games" seem to have higher F-measure values than the macroaverage for most of the dictionaries both in cases of token match and substring match. In practice dictionary-based baselines using tokens might be of interest if high precision but not necessarily high recall is required. For example, a topic-focused crawler might want to detect some surely relevant URLs early during a crawl and then use these pages to "bootstrap", for example, using the link information (see Section 5.4).

5.3 Confusion Between Classes

Although it is "obvious" that a page with sexually explicit content should be filed under "Adult" and a page about a football club under "Sports", it is less clear what distinguishes a "Reference" URL from a "Science" URL in ODP. This kind of confusion is inherent to the problem and any classifier will suffer from it. (In Section 5.7 we will argue that this limitation can be partly overcome when the summaries of pages in ODP are used, as they contain the explanation of why a certain page was listed under a certain topic by a human editor.)

In this section we investigate the confusion between topics: (1) by looking at the topic distribution of test URLs whose domains were seen in the training set and (2) by looking at the cases where the URL-based topic classifiers misclassified.

The amount of confusion between topics in ODP can be partly quantified via the diversity of the domains. If the same domain hosts both "Reference" and "Science" pages then this indicates an unclear division. Similarly, it is unlikely that a domain hosting "Adult" content will also host "News" pages and this indicates a clearer topic separation.

[11] Note that the precision, recall, and F-measure are averaged separately so that the average F-measure does not have to lie between the average precision and the average recall.

Table VIII. The Topic Distribution of Test URLs Whose Domains were Seen in the Training Set of the Topics in ODP

	Ad	Ar	Bu	Co	Ga	He	Ho	KT	Ne	Rc	Rf	Sc	Sh	So	Sp
Adult (Ad)	**88**	5	0	0	3	0	0.3	5	6	5	0	0	0	0	1
Arts (Ar)	0	**29**	0	0	6	0.4	2	4	4	4	3	3	0	5	4
Business (Bu)	**6**	8	**86**	1	4	5	2	**17**	2	9	4	5	2	0.4	5
Computers (Co)	1	2	3	**92**	2	1	3	4	3	2	8	7	1	2	3
Games (Ga)	0	5	0	3	**50**	0	3	3	1	2	1	1	0	0	2
Health (He)	0	3	0.3	0	2	**78**	4	3	2	3	6	6	0.2	2	2
Home (Ho)	0	2	0	0	4	1	**69**	2	4	2	3	2	0	2	1
Kids&Teens (KT)	2	3	4	1	6	2	2	**30**	1	2	4	9	0	7	6
News (Ne)	**0.3**	4	1	0	1	2	1	3	**63**	3	5	6	0	4	5
Recreation (Rc)	1	**13**	1	0.4	9	2	2	7	3	**29**	3	3	0	1	6
Reference (Rf)	0	1	0	1	0.3	2	4	1	2	1	**40**	**12**	0	1	1
Science (Sc)	0	1	0.4	1	1	2	3	6	2	1	**15**	**37**	0	3	3
Shopping (Sh)	1	**14**	4	0.3	7	4	0.4	4	0.3	**26**	1	1	**97**	0	**7**
Society (So)	0	6	0.1	1	2	2	3	5	7	5	5	5	0	**72**	4
Sports (Sp)	0	5	0	0	4	1	1	6	3	5	2	4	0	2	**49**

The row corresponds to real topic of the test URL and the column corresponds to the training topic. Entries in one column sum up to 100% and an entry (X,Y) in this matrix gives the answer of "Among the test URLs which had a domain in the training set of topic *Y*, what is the percentage of URLs which were actually labeled as topic *X*?"

Table VIII shows the topic distribution of test URLs whose domains were seen in the training set of the topics in ODP. The main findings from this table are the following: 15% of the test URLs which had a domain in the training set of "Reference" were actually labeled as "Science", while 12% of the test URLs with a domain from the "Science" training set were from "Reference". As expected, a domain hosting "Adult" content will not also host "News" pages, 0.3% of the test URLs with a domain from the "Adult" training set were actually labeled as "News." Among the test URLs which had a domain in the training set of "Adult", the percentage of URLs which were actually labeled as "Adult" was 88%. Only 29% of the test URLs with a domain from the "Recreation" training set were indeed from "Recreation" and 26% of the test URLs with a domain from the same training set were actually labeled as "Shopping." These results indicate that domains hosting topics like Adult, Business, Shopping, and Computers seem to host only these topics in large percentages. On the other hand, domains hosting topics like Reference, Science, and Recreation also include considerable amount of pages from other topics. In other words it is not possible to say that domains can be used to have a clear division between topics.

In addition to the inherent confusion (or even ill-definedness) of the problem, there is another kind of confusion which is due to the fact that the vocabulary of different topics is similar. For example, the term "teens" in a URL is an indicator for both the "Kids and Teens" class but, unfortunately, also for "Adult." Apart from the problem of empty URLs (see Section 5.5), these two kinds of confusion are the main reasons for errors of our classifier. Table IX shows the confusion matrix for our best-performing binary classifiers. As can be seen there is a significant amount of confusion between "Science" and "Reference", between "Shopping" and "Business", but also between "Computers" and "Business". The large entry for "Business" in the last column shows that "Business" URLs are the most confusing ones (as several classifiers trigger for them) whereas the large entry for "Recreation" in the last row shows that "Recreation" has the most confused classifier (which is triggered by several classes). The human judges were also confused with "Recreation" and "Business." The large number of "don't know" choices for these classes indicate this (see Section 5.1).

Table IX. Confusion Matrix for the Default Binary Classifiers Using ME with All-Grams

	Ad	Ar	Bu	Co	Ga	He	Ho	KT	Ne	Rc	Rf	Sc	Sh	So	Sp	$\sum$
Adult (Ad)	85	18	12	13	10	6	8	16	12	15	4	5	17	9	5	235
Arrts (Ar)	18	83	20	17	23	11	13	26	22	31	12	15	24	26	14	355
Business (Bu)	18	24	84	**45**	12	18	15	16	**39**	**34**	17	22	**53**	22	15	434
Computers (Co)	15	21	**42**	80	25	15	14	18	27	22	18	**30**	28	18	11	384
Games (Ga)	8	25	9	17	88	5	8	18	11	23	4	8	12	12	11	259
Health (He)	7	15	19	19	7	79	14	24	22	22	26	25	18	30	11	338
Home (Ho)	8	17	12	12	8	15	80	15	13	20	9	12	19	18	9	267
Kids&Teens (KT)	18	25	11	19	15	13	12	79	14	24	19	**31**	11	19	10	310
News (Ne)	10	29	31	21	7	14	11	13	82	20	17	16	22	**32**	15	340
Recreation (Rc)	14	28	24	19	23	12	18	**30**	22	82	13	16	**32**	25	27	381
Reference (Rf)	5	14	10	20	6	19	8	29	19	13	83	**54**	6	28	17	331
Science (Sc)	7	14	18	**30**	9	20	12	**35**	17	17	**39**	81	11	24	10	344
Shopping (Sh)	20	30	**54**	28	16	13	23	16	**31**	**37**	8	10	82	16	17	401
Society (So)	11	**30**	18	19	14	27	17	28	**30**	28	27	23	16	81	15	384
Sports (Sp)	10	18	16	11	17	10	10	16	21	**34**	17	9	26	18	84	317
$\sum$	254	391	380	370	280	277	263	379	382	418	313	357	377	378	271	

A row corresponds to the true label of a URL and answers the question: "To what percentage of class X URLs, binary classifiers say YES?" A column corresponds to the predicted label of a classifier and answers the question: "Which URLs cause classifier Y to trigger?" A perfect classifier without any classification errors would have a 100 for the entry on the diagonal (= perfect recall) and zeros in the other cells of its column (= perfect precision).

5.4 Using Inlinks

In the setting of a topic-focused crawler at least one inlink will be known for each URL to be classified. As shown, for example, in Qi and Davison [2009] using such link information generally helps with the problem of topic classification. We explored the potential of this approach by obtaining up to 100 inlinks for each of our test URLs from Google using its "link:" feature. Incoming links from the same domain or from `http://www.dmoz.org` were removed, but an estimated 0.1% of obtained inlinks come from ODP mirror sites. For 9,713 out of 15,000 test URLs we managed to obtain at least one inlink with an average number of 17.2. In total, the true category of the linked-to page and the assigned category of the linking page were congruent in 29% of the cases (macroaveraged)[12]. Although far from being perfect, this is significantly higher than guessing uniformly at random between 15 classes. We then tried two different ways to exploit this information. In both cases we used up to 10 inlinks per URL as more seems unreasonable in most scenarios. The performance was only evaluated on the 9,713 URLs which had at least one inlink.

First, we considered the inlinks to simply be an "extension" of the actual URL. That is the URL was lengthened by concatenating it with the URLs of the (up to) 10 inlinks. To avoid the situation that the features of the actual URL were drowned among the inlinks, we combined inlinks with the original URL in a ratio of 1:2. That is, for each inlink we added we took two copies of the original URL. These long URLs were then fed through the same binary classifiers using all-grams with ME, trained on "normal" URLs, and evaluated. This approach led to macroaveraged performance values of P = 79, R = 71, and F = 74, where all per-class performances were evaluated in a balanced binary setting; see Section 2.2. Theoretically, one could use different feature sets for the inlinks and the source URL, but we did not experiment with this approach as it did not appear very promising.

As the preceding approach generates long URLs with arguably abnormal characteristics, we also tried first classifying each inlink separately and then using a voting scheme on the inlinks. As an example, to decide whether a URL with 8 inlinks should be classified as "Sports", we first computed the corresponding binary classification of the inlinks using our default scheme and, for example, 3 URLs were labeled "Sports"

[12] To obtain a *unique* label for each inlink we took the classification result of the binary classifier which had the highest confidence for a "yes" decision.

Table X. Percentage of Test URLs Consisting of Only Stop Words or Features, Not Seen in Any of the URLs of the ODP Training Set

Features	4-gr	6-gr	8-gr	10-gr	12-gr	Tokens
Empty	0.2%	3.5%	11.6%	21.1%	27.3%	32.0%

The numbers are the macroaverages across all the 15 test sets for ODP.

and 5 as “non-Sports”. If at least 30% of the inlinks were classified as “Sports” we counted this as a “yes” decision of the inlinks, which was the case in the example. Then we also classified the URL itself and, let us suppose, obtained a “no” decision. These two decision were then either AND-ed (“yes” if and only if *both* decisions were “yes”) or OR-ed (“yes” if and only if at least one decision was “yes”). The AND setting, which improves precision, had a macroaveraged performance of $P = 91$, $R = 60$, and $F = 72$. The OR setting, which improves recall, had a macroaveraged performance of $P = 75$, $R = 92$, and $F = 82$. As can be seen in Table III the original macroaveraged performance for this experiment was $P = 83$, $R = 82$, and $F = 83$ without the use of inlinks.

None of the two approaches, extending the target URL by inlinks or using voting on the inlinks, seems to improve over the default scheme (see Table III). However, when the baseline performance is computed for the same subset of URLs then one obtains a macroaveraged performance of merely $P = 81$, $R = 77$, and $F = 79$. URLs with inlinks are apparently harder to classify than average URLs. The reason for this seems to be that inlinks tend to point to the main page such as `http://www.bbc.co.uk` rather than to a subpage such as `http://news.bbc.co.uk/1/hi/entertainment/1059124.stm`, which is easier to classify. The macroaveraged number of tokens (including `http`) for the test URLs with inlinks was 5.4, where it was 5.7 for the overall test set[13]. Given this observation, using a voting scheme with inlinks does indeed improve the baseline performance. However, as: (i) inlinks are not always available and (ii) if they are available there is most likely also content for the inlinking pages, we decided not to pursue this approach further within our work on purely URL-based topic classification. In practice it might, however, still be of interest in scenarios where recall is important as inlinks help to avoid the problem of “empty URLs” discussed in the following section.

5.5 Empty URLs

As mentioned in Section 2.3, *n*-grams make it possible to find subwords in long tokens. In fact, if only tokens are used, then a substantial part of test URLs are “empty,” meaning they consist only of stop words or of tokens which are never seen in the training set. A typical example is `http://www.yuzutree.com` or `http://www.xboxgames.com`, and such cases constitute the majority in “Business” and “Shopping” (see the “Empty” column in Table XI).

Table X shows how the percentage of empty URLs drops as shorter and shorter *n*-grams are used. It also shows that any token-based classification algorithm, which uses only URLs, could not do better than to guess for 32.0% of URLs. For Table X we used a stop-word list containing protocols (“http”, ...), generic Web terms (“index”, “www”, ...), and most common top-level domains (“com”, “net”, ...), apart from “edu” and “gov”, as these are related to specific topics.

Note that the chance that two random sequences of eight characters coincide are negligibly small. So the big drop of 20.4% in empty URLs by going from tokens to

[13] Surprisingly, we will see later in Section 5.6 that the number of tokens is not a good indicator for the relative difficulty *between* classes (which also depends on many other factors; see Section 5.3), although it does appear to be a measure of difficulty *within* a given class.

8-grams shows that 8-grams already capture a significant amount of repeatedly appearing letter sequences, most likely subwords, such as “shopping” or “articles”. Such subwords appear in longer tokens such as “bargainshoppingparadise”or “newsarticles”[14]. In our experiments, we used all-grams (see Section 2.3), for which the percentage of empty URLs is the same as for 4-grams.

Of course, even all-grams will not help when a URL such as `http://malx.co.uk/`[15] needs to be classified and the token “malx” has not been seen before. We believe that such URLs are the reason why for all datasets in Table III the recall is lower than desired.

5.6 Properties of Different ODP Topics

We tried to find characteristics that could explain to us why some topics seemed harder than others to classify. In other words we searched for characteristics which predict the relative difficulty of classifying URLs for a topic. As empty URLs pose a challenge for URL-based topic classification, we tried to form some notion for emptiness with the following explained characteristics.

First we looked at the percentage of empty URLs (see Section 5.5), which contain only unseen tokens and which are the main reason to use *n*-grams. Then we also investigated a related measure, namely, the ratio of URLs-per-domain (UpD) in the test set. If for a particular dataset and a particular topic the ratio of URLs-per-domain is high, then there is good chance that one has already seen the domain of a test URL in the training set, which is often a strong signal. In addition, we also considered the length of a URL as an indicator of the difficulty (or easiness) for classification. As “length,” we used both the number of tokens (including http) and the number of characters (including http). Table XI gives these characteristics for each topic, as well as again reporting the F-measure, when ME is used in combination with all-grams, for making correlations between classification performance and characteristics.

None of these four characteristics turned out to have a strong predictive power. Concerning length, only one out of the three topics with the longest URLs had an above average F-measure. Similarly, “Business” and “Shopping” have short URLs, but both performed better than most other topics. As for UpD, the topic with the highest UpD (= “Games”) also has a high F-measure, but the two topics with the lowest UpD (“Business” and “Shopping”) also perform above average. The relatively good performances for “Business” and “Shopping,” despite the fact that the percentage of empty URLs is high for both classes and that the UpD is very low, show that the classification algorithms do more than simply memorize domain names. For hosting sites such as `http://www.blogger.com` there is also a high topic diversity, illustrating that UpD is not a good predictor for classification performance.

5.7 Using Summaries

To have an upper bound for the performance of any classification algorithm, which does not use hyperlink information, we experimented with classifying Web pages, using short summaries. Short summaries were reported in Shen et al. [2004] to be preferable to the content of Web pages, and for the ODP dataset these summaries were also easy to obtain. We refer to these human-edited summaries as “snippets.” Training and testing ME models with all-grams on the combination of snippets and URLs gave an F-measure averaged, across all 15 topics, of 94 (see Table XII). Here, the combination of snippets and URLs was done by a simple concatenation from which the n-gram

[14] `http://www.wanttoknow.info/newsarticles`

[15] A URL in the ODP “News” category.

Table XI. The Table Shows Different Properties of the Different Test Sets of URLs for ODP

Topic	Empty	UpD	#tokens	#chars	F
Adult	24.7%	1.1	5.8	37.0	88
Arts	21.9%	1.4	6.1	39.8	82
Business	64.5%	1.0	4.4	27.7	82
Computers	34.6%	1.1	5.2	32.1	81
Games	15.1%	1.9	6.5	42.2	88
Health	38.6%	1.2	5.6	36.6	82
Home	19.7%	1.5	6.5	44.3	83
Kids & T.	12.1%	1.3	6.2	39.2	80
News	45.4%	1.1	4.9	31.5	81
Recreation	40.2%	1.2	5.1	33.1	81
Reference	14.9%	1.4	5.7	33.5	85
Science	14.2%	1.3	6.9	42.2	82
Shopping	69.9%	1.0	4.3	29.0	81
Society	31.1%	1.2	5.5	35.5	81
Sports	34.0%	1.3	5.8	37.8	85
Average	32.1%	1.3	5.7	36.1	83

An "empty" URL consists uniquely of unseen tokens (see Section 5.5). "UpD" stands for URLs-per-domain and gives the ratio between the size of the ODP test set and the number of distinct domain names seen in the test set. The next two columns give the average "length" of a test URL, once counting the number of tokens and once counting the number of characters. The F-measure is for binary ME models with all-grams, using only URLs. Note that the properties are *not* features used by our algorithms and are only given for explanatory purposes.

Table XII. Macroaveraged Performance for the Three Basic Train-Test Configurations on the ODP Dataset

Train: Snippets + URLs						Train: URLs		
Test: Snipp+U			Test: URLs			Test: URLs		
P	R	F	P	R	F	P	R	F
94	94	94	83	76	79	83	82	83

Note that training on snippets and URLs when testing only on URLs, *hurts* the performance, compared to training only on URLs.

features were then extracted. The accuracy of the corresponding multiway classifier, which simply asks all binary classifiers and trusts the one with the highest reported confidence value, was 81% when all 15 categories were used, and 83% when only the same set of 12 as used in Qi and Davison [2006] were used. Using tokens rather than all-grams gave a macroaveraged F-measure of 94. The fact that both features performed virtually the same shows that all-grams are not needed when some kind of content is available.

However, when we tested the same model only on URLs which had been trained on both snippets and URLs, the performance dropped to an F-measure of 79. How can it be that using more data to train the model does not only not improve but actually hurt the performance? The reason is that the vocabulary used to describe a Web page is different from the one used to construct its URL. Table XIII shows a few representative

Table XIII. Examples of Words which are Topic-Specific when Snippets are Used, but Which are No Longer Specific when Only URLs are Used

Business	compan*, firm, manufactur*, product*, custom
Health	cause*, center, effect*, families, providing, side
News	classified*, covering, issues, stories, weekly
Sports	club, fixtures, results, official, statistics

words which are strong indicators for a topic, when snippets are used, but which lose their indicative power when only URLs are available.

Using only Tokens by-Statistics dictionary (see Section 5.2), trained from both URLs and snippets, together with token matching rather than substring matching, gives a macroaveraged F-measure of 85 when testing on both URLs and snippets. So using only a simple dictionary to classify Web pages using their summaries already gives a respectable performance.

Although we were certainly expecting an improved performance when snippets were used for the task, we were still surprised to obtain an F-measure of 94. How is such an high performance possible, given the inherent fuzziness in ODP (see Section 5.3)? Does an article about the health benefits of swimming fall under "Health" or under "Sports"? Does a page with news about technical gadgets fall under "News" or under "Computers"? We believe that the main reason that the snippets circumvent these problems is that they contain the definition of the topic, essentially explaining why the editor chose to put the URL in the specific category.

For example, the URL `http://codeinepub.tripod.com/` is listed under "Recreation" → "Drugs" and not under either of "Health" → "Pharmacy" → "Drugs and Medications" or "Health" → "Addictions" → "Substance Abuse." Even though these alternative classifications would seem equally appropriate, the following snippet explains the chosen classification. "Codeine Information - A nonprofit source of information on the recreational use of codeine, dihydrocodeine, hydrocodone, oxycodone, and pholcodine. Includes facts, extraction guide, list of products containing codeine, information on side-effects, addiction, and tolerance."

5.8 Data Consistency

We saw in Section 4.3 that training models on the ODP dataset and then evaluating them on the other datasets generally leads to a poor performance, even when the topic has the same name in both datasets. The main reason for this is "data consistency" which refers to the problem that in different Web directories different definitions of topics are used. For example, in the ODP all movie-related pages such as `http://www.imdb.com` are listed under "Arts," but in the Yahoo! directory these pages are listed under "Entertainment." Similarly, the tag "adult" is also used for "adult swimming classes" in Delicious, but refers exclusively to pornographic pages in the ODP. Such inconsistencies make cross-training (train on ODP, test on Yahoo!) very difficult and also make it difficult to merge training sets in a meaningful manner. These kinds of problems need to be addressed when different taxonomies are merged [Avesani et al. 2005].

We looked specifically at the problem of data consistency between ODP and Yahoo! by asking: Assuming a page is listed both in ODP and in the Yahoo! directory, then in which ODP categories are pages from a given Yahoo! directory category found? Table XIV looks at the problem of data consistency between ODP and Yahoo! by looking at the URLs listed in both directories.

Table XIV. In which Topics are Web Pages from the Yahoo! Directory, which are also Listed in the ODP, Found in which Topics of ODP?

		Yahoo! Topics												
		Art	Bus	Com	Edu	Ent	Gov	Hea	New	Rec	Ref	Sci	SoS	S&C
ODP Topics	Adult	6	**15**	2	0	6	0	1	0	0	0	0	1	11
	Arts	27	2	1	1	60	1	0	7	0	1	0	3	1
	Busin.	1	20	1	1	0	1	0	0	1	1	1	1	1
	Comp.	2	11	**84**	1	2	1	0	2	1	4	4	1	1
	Games	1	3	2	0	5	0	0	0	**34**	0	0	1	1
	Health	0	5	0	2	0	2	**83**	0	0	0	3	1	2
	Home	2	3	3	2	3	5	2	1	1	1	7	0	**40**
	K&T	19	1	2	9	6	6	2	1	6	10	21	9	9
	News	4	4	1	1	4	22	0	**84**	1	0	1	6	2
	Recr.	2	6	1	1	6	1	1	2	17	1	12	2	4
	Refer.	17	3	2	74	2	10	5	1	1	79	12	28	4
	Science	2	4	1	2	0	10	2	1	1	1	35	41	3
	Shopp.	2	20	0	0	1	0	0	0	0	0	0	0	0
	Soc.	16	3	1	8	3	40	4	1	1	2	3	8	23
	Sports	0	2	0	0	2	0	1	0	36	0	0	0	0

Each column sums up to 100 (modulo rounding) and shows the distribution of Web pages from the Yahoo! Directory in the ODP.

It immediately shows that the *only* topics where one might expect cross-training to work are “Computers”, “Health”, and “News” because among the Yahoo! pages which appeared in these topics and also appeared somewhere in ODP, 80% of them also appeared in ODP under these topics. Indeed, the first two are the best-performing classes in the setting where the training is done on ODP and testing is done on Yahoo!. As can be seen from Table IV “Computers” and “Health” have an F-measure of 80 and 82, respectively. The low F-measure value 53 for “News” in Table IV is mainly caused by low recall, which in turn is caused by a much broader definition of “News” in Yahoo!. Yahoo! lists many pages about news presenters (such as Wikipedia articles or IMDB pages) or home pages of small radio stations as “News,” whereas these pages are not listed at all in ODP. Table XIV also shows that, for example, many “Business” Yahoo! URLs end up in “Adult” under the ODP hierarchy (as “Adult Internet Services” are listed under “Business and Economy” in Yahoo!), many “Recreation” URLs end up in “Games” (as Yahoo! lists games under “Recreation”) and many “Society and Culture” URLs end up in “Home” (as “Cooking” is listed under “Society and Culture” in Yahoo!, but under “Home” in ODP). In constructing the table, the different ODP topics were normalized such that the largest class (“Arts”) does not a priori attract more URLs from Yahoo!, than the smallest class.

5.9 Combining Different Classifiers

How much can we improve the performance by combining different algorithms by using boosting algorithms on top of the outcome of our baseline algorithms (NB, SVM, and ME) in combination with all-grams (derived from tokens and directly derived from the URL)? To answer this, the test set for each topic was further split into halves. One half was used to train the boosting algorithm and the other half was used to evaluate the performance of the boosting algorithm.

The six-dimensional feature input vectors for the boosting algorithms were the “yes” probabilities output by NB, SVM, and ME for all-grams derived from tokens and all-grams derived directly from the URL and spanning across tokens. We also tried boosting on the six-dimensional binary output. However, using the explicit probabilities performed better as in this setting the booster can learn to, say, not trust SVM if the

Table XV. By Allowing Different Algorithms (SVM, ME, or NB) and Combinations of Algorithms via Boosting for Different Topics, the Macroaveraged F-Measure is Improved from 83 to 84 for the ODP Dataset Compared to Using ME with All-Grams for all Topics (see Table III)

Topic	Method	P	ΔP	R	ΔR	F	ΔF
Adult	GentleAdaB	90	0	87	+2	89	+1
Arts	ModestAdaB	82	+1	84	+1	83	+1
Busin.	SVM	79	-1	90	+6	84	+2
Comp.	SVM	83	+1	85	+5	84	+3
Games	ModestAdaB	90	+1	89	+1	90	+2
Health	ModestAdaB	84	-1	84	+5	84	+2
Home	ModestAdaB	87	+1	81	+1	84	+1
Kids	RealAdaB	82	0	85	+6	83	+3
News	ModestAdaB	86	+5	79	-3	82	+1
Recreat.	RealAdaB	77	-3	89	+7	83	+2
Refer.	SVM	90	+3	82	-1	86	+1
Science	ModestAdaB	83	0	81	0	82	0
Shopp.	RealAdaB	77	-2	91	+9	83	+2
Society	ModestAdaB	82	+1	82	+1	82	+1
Sports	ModestAdaB	87	0	86	+2	87	+2
Average	-	84	0.4	85	2.8	84	1.6

For “Computers” and “Kids and Teens” the improvement is 3 points in F-measure. The best results among boosted and atomic methods, chosen on a per-topic basis, for the other datasets were 81 (+2) for Yahoo!, 87 (+2) for Wikipedia, 80 (+1) for Delicious, and 88 (+1) for Google in terms of macroaveraged F-measure.

point is very close to the decision boundary if at the same time NB is really sure of its decision.

We experimented with RealAdaBoost [Freud and Schapire 1996], ModestAdaBoost [Vezhnevets and Vezhnevets 2005], and GentleAdaBoost [Friedman et al. 2000]. ModestAdaBoost generally gave the best results though not all topics could be improved via boosting. After boosting, for each topic we chose the better of the best boosted performance and the best “atomic” method (SVM, ME, or NB). Table XV shows that this choice of locally best (boosted) method gives a performance increase of 1–2 points in F-measure across all the datasets.

6. RELATED WORK

For a recent survey of work related to topic classification of Web pages in general, we refer the reader to Qi and Davison [2009].

A first step towards purely URL-based topic classification was recently presented in Baykan et al. [2009]. There the authors present experimental results for a study with three feature sets and three algorithms for the Open Directory Project (ODP) dataset (see Section 3). The present work is a significant extension in terms of the number of datasets used, the number of algorithms and feature sets experimented with, and in terms of completely new approaches (human evaluation, use of inlinks, use of snippets, training on dataset A while testing on dataset B, combining different algorithms by boosting). Apart from the substantial difference in comprehensiveness this article also presents the reader with much more insights, well beyond providing the results of algorithm X on dataset Y.

6.1 Content Classification of ODP

When only the content of a Web page (and no link information) is used, the best performance reported in the literature for the ODP dataset is between 73 and 77 in terms of accuracy [Qi and Davison 2006, 2008]. This is for a setting in which only 12 out of our 15 ODP categories are used. They use Support Vector Machine (SVM) for the classification algorithm and they use word counts for the feature set. When we map our corresponding 12 binary SVM models to their multiway setting, we obtain an accuracy of 58 using only URLs[16]. Note, however, that each of our "one-against-many" models was trained with 14, rather than 11, negative classes. Although this comparison shows that the content of a Web page helps with the classification task this is, of course, only possible if the content is available. This is not always the case (see the discussion in the Introduction). Additionally, a classifier using the content of a Web page will be slower than one using only its URL.

The best reported accuracy for ODP is for a classification model that does not only use the content of the page to be classified, but also content about neighboring pages. Here, the content is "fielded" and titles, full text, and anchor text are treated separately. "Neighboring" pages do not only include pages that the page under consideration is linked from or that it links to, but also colinking pages which share a common link source page or a common link target page. In this setting, a classification accuracy of up to 90 can be achieved [Qi and Davison 2008].

Using only the summaries and URLs of Web pages, we obtained an accuracy of 83, compared to the aforementioned 73–77 for the use of the content of a page. The observation that short summaries are, for the task of classification, preferable to the content of Web pages was also made in Shen et al. [2004]. Performance results for our binary classifiers trained on summaries are given in Section 5.7. Using additional information like URLs with the content or the summary of a Web page as a hybrid approach can complement the weaknesses of both URL-based and content-based classification. This hybrid approach can be useful for applications like the on-the-fly classification of Web search results [Chen and Dumais 2000], where only limited content is available and speed is of utmost importance.

In a recent work [Power et al. 2009] Web pages are classified into focused topics which are not as frequently seen on the Web as general topics like Sports. Furthermore, the overlap between focused topics is little according to the definition of focused topics in Power et al. [2009]. For features, they use a list of frequent words and a list of rare words related to each topic. These features are obtained by using thresholds which are determined manually. On a test set which is acquired after querying Google with the topic as query, they achieved a recall and precision of 99 points with SVM algorithm. On the WebKB dataset, crawled Web pages of 4 universities (see Section 6.2), where as the topics are student, faculty, course, and project pages are used, they achieved 91 points of accuracy with Naïve Bayes.

6.2 URL Classification of "Four Universities"

The idea of classifying Web pages into topics using only URLs is not a new one. Apart from the work in Baykan et al. [2009], different research groups tried such an approach using the "4 Universities" dataset, containing 4,167 URLs from the universities of Cornell, Texas, Washington, and Wisconsin, as well as other universities grouped as "misc". These pages were collected in January 1997 and are classified into student,

[16] For this mapping, each test URL was given to all 12 classifiers, and the classification outcome of the classifier with the highest reported confidence was used as the outcome of the corresponding multiway classifier.

faculty, course, and project pages [Chaker and Habib 2007; Devi et al. 2007; Kan 2004; Kan and Nguyen 2005].

The first work in this series is Kan [2004]. As the authors do not use character *n*-grams (see Section 2.3), they explore two different ways to detect subwords. In one approach, they try all possible ways to split a token into parts, and then use the partitioning which has the highest probability, where the probabilities are estimated using token frequencies from the WebBase project.[17] In their second approach, which gave better results, they used titles of Web pages in a training phase to detect that certain abbreviations, such as "cs", often appear in URLs where the title often contains the corresponding complete terms such as "computer science." This then allowed them to expand such letter sequences in the test set. Using this method with SVM as a machine learning algorithm, they obtain a macroaveraged F-measure of 34 when only URLs are used. They also show that the information from URLs is more valuable than the information from anchor text of incoming hyperlinks.

The authors extend their work in Kan and Nguyen [2005] by experimenting with different sets of features derived from URLs, such as treating tokens seen at the last directory of the URL and at the first directory of the URL differently and also taking token sequences into account. With these new features, and by using a maximum entropy classifier, they improve their results to 53 in macroaveraged F-measure. In the same setting when they used the content of Web pages for classification with the ME algorithm they achieved a macroaveraged F-measure of 60.

To understand how our study in this article relates to works in Kan [2004], and Kan and Nguyen [2005] in terms of performance, we used Support Vector Machine (SVM), Naïve Bayes, and Maximum Entropy algorithms in binary classifiers for each of the four categories. As feature set we used grams (4-, 5-, 6-, 7-, 8-grams) derived from URLs. We used the same leave-one-university-out setup as in Devi et al. [2007], Kan [2004], and Kan and Nguyen [2005], where for each category four different models are trained, each with three universities and the "misc" in the training set and the fourth university in the test set. For each category, we took the average of the four F-measures, one for each university. Unlike our main experiments no balancing between positive and negative samples was done for the test set, to make performance numbers comparable. For this setting, the Naïve Bayes models obtained F-measures of 63, 75, 78, and 50, for the categories student, faculty, course, and project, respectively. This gives a macroaveraged F-measure of 66, which improves the best previously reported F-measure [Kan and Nguyen 2005] for the "URL only" setting by 13 points. Our performance is 6 points better and is comparable to their results [Kan and Nguyen 2005] when they used the content of the Web pages. The macroaveraged F-measure for SVM and maximum entropy was 59. Though the performance gain is encouraging, we did not try to improve these numbers further, as our main focus is on topic classification of arbitrary Web pages, rather than on function classification of university pages.

In Devi et al. [2007] different algorithms for the same problem are compared. The authors write that they "parse the URL into meaningful words", but how this is done remains unclear. Their best-performing algorithm is a radial basis function network which leads to a success rate of 52%. However, on the test set this algorithm classifies everything as positive.

The authors of Chaker and Habib [2007] report a microaveraged break-even-point (BEP) of 74 for the same dataset with two added categories ("staff" and "department"), when using only URLs. How this compares to other results is unclear, as the BEP is the arithmetic mean of precision and recall, which is always higher than the F-measure, which is the harmonic mean. Similarly, the difference between the

[17] http://dbpubs.stanford.edu:8091/ testbed/doc2/WebBase/

macro- and microaveraged results are not reported. Furthermore, they do not use the leave-one-university-out evaluation methodology, which makes it easier for their method to discover the typical URL structure for each university-category pair from the training set. They use tokens (see Section 2.3) as features with additional stemming and a feature selection step. In this step they use "only words that appear more than 3 times in the document and appear in more than 3 documents", which seems to imply they use the content of a document to prune the set of tokens in its URL.

The 4-universities dataset was also used, along with other small datasets, for *clustering* rather than classification experiments in P and Khemani [2006]. Clustering can potentially make the problem easier as URLs, or more generally data points, are no longer "labeled" independently of each other. For example, one difficult URL might be similar to another URL which is easier to label. If these two URLs end up in the same cluster and if a label for the whole cluster can be found, then this label can also be applied to the difficult URL. On the other hand a *classification* algorithm with a training set and test set split, labels a test URL one at a time independent of the labels of the other test URLs. This is also the scenario that is required for most applications where a single decision is required instantly. The evaluation in P and Khemani [2006] also differs substantially from ours. They use agglomerative hierarchical clustering and they focus on the "purity" of the clusters when only 50% of merges have been done (i.e., when there are 150 URLs to cluster there are still 75 disjoint clusters). This way, they avoid having to make any statement about difficult URLs or other outliers for which no similar URL exists. These cases will be kept in their own 1-item clusters with perfect purity and will only be merged at the very end. In our evaluation setup we always output an answer. When we remove test URLs with unseen domains, the performance for a topic such as Reference increases to 90 in F-measure. The problem of "empty" URLs with only unseen tokens is discussed in detail in Section 5.5.

6.3 Other Related Work

In Zhang et al. [2006] the use of URLs for detecting pornographic Web pages is analyzed. The motivation of their work is content filtering in public institutions, where teenagers try to access such pages. As these Web pages often contain only images, a content-based classification is difficult. They use a slight variant of our "all-grams" (see Section 2.3) as features, with a range of 3–10 rather than 4–8 characters, together with their own custom-made linear classifier. In their experiments they "achieved 80–90% true positive rate with less than 10% false positive rate", which is comparable to our results for the "Adult" class for the Open Directory Project dataset (see Table XV).

The observation that for a single site all pages with a URL prefix such as `http://edition.cnn.com/TECH/` will most likely be about the same topic ("Technology"), led the authors in Shih and Karger [2004] to construct a hierarchical URL tree for classification, which is then used as one of two features for classification. Their use of such information is, however, mostly useful for classifying pages on the same host. When arbitrary Web pages need to be classified and only one page per host is available, this approach seems of limited use.

Related to our problem of combining various datasets (see Sections 4.3 and 5.8) is the general problem of matching taxonomies, which is studied, for example, in Avesani et al. [2005], McGuinness et al. [2000], Noy [2004], Stumme and Maedche [2001], and Zhang and Lee [2004].

In the literature there are papers which use information extracted from URLs in different contexts. In the same spirit of "how much information is hiding in a URL," there is work on detecting the language of a Web page using *only* its URL [Baykan et al. 2008]. The authors show that, surprisingly, the language of a Web page can be deduced from the URL alone with an accuracy rivaling the best content-based algorithms. Silvestri [2007] proposed a method which uses the lexicographical ordering of URLs to enhance the performance of the compression of inverted file indexes in Web Search Engines. Umbrich et al. [2009] presented a focused crawler for media type targeted search engines. In their crawler they only used URLs. Koppula et al. [2010] present techniques to find rules from URLs to identify duplicate Web pages. Freudiger et al. [2009] use URLs of Web pages to decide whether they contain potentially sensitive information from which online advertisers can benefit by putting third-party cookies during browsing.

7. CONCLUSIONS

In this article we studied Web page topic classification from URL. Even though content-based topic classifiers gave better results than URL-based ones, topic classification from URL is preferable when the content is not available, or when classification speed has the highest importance.

We can summarize our main findings for URL-based Web page topic classification as follows.

(1) We showed that dictionary-based baseline algorithms are not enough for high-performance URL-based topic classification. For the dictionary-based methods even the best-performing variant using all-grams (a combination of 4-, 5-, 6-, 7-, and 8-grams) achieved a macroaveraged F-measure of only 73. On the other hand for topic classifiers where precision is important and some recall can be sacrificed, token-based statistical dictionaries can be used as they achieved a macroaveraged precision of 92 with a macroaveraged recall of 34 on the ODP dataset. (2) We showed that the features have more impact on the classifier performance than the classification algorithms. All-grams derived from URLs was the best feature set, considerably better than tokens. Explicitly encoding positional information for all-grams derived from tokens performed slightly better than using only such all-grams without positional information, but still slightly worse than using all-grams from the URL. Given the same feature set, the ME and SVM algorithms showed the same performance, and the other algorithms also performed similarly. (3) We obtained a macroaveraged F-measure of 83 when ME algorithm is used with all-grams derived directly from URLs. (4) We reported a performance which improves the best previously reported URL-only performance for a small dataset of university pages by 13 points in F-measure. (5) Using summaries of Web pages for training and testing led to a large improvement over using only URLs, with a macroaveraged F-measure of 94. On the other hand the performance of URL-based topic classification decreased when the summaries of Web pages are used in training phase in addition to URLs. The reason for this is the vocabulary difference between URLs and the summaries of Web pages. (6) We achieved an additional small improvement with using inlink information. (7) Applying boosting to combine different algorithms gave a small performance improvement of 1 or 2 points in F-measure. (8) The challenges for URL-based topic classification are: (i) data consistency as the definitions of topics differ from one dataset to another, (ii) overlap between different topics in one dataset, (iii) empty URLs consisting of only stop tokens or previously unseen tokens.

REFERENCES

ALEX, P., CHIRITA, R., COSTACHE, S., NEJDL, W., AND HANDSCHUH, S. 2007. P-tag: Large scale automatic generation of personalized annotation tags for the Web. In *Proceedings of the International Conference on World Wide Web (WWW)*. 8–12.

AVESANI, P., GIUNCHIGLIA, F., AND YATSKEVICH, M. 2005. A large scale taxonomy mapping evaluation. In *Proceedings of the International Semantic Web Conference (ISWC)*. 67–81.

Baykan, E., Henzinger, M., and Weber, I. 2008. Web page language identification based on URLs. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*. 176–187.

Baykan, E., Henzinger, M., Marian, L., and Weber, I. 2009. Purely URL-based topic classification. In *Proceedings of the International Conference on World Wide Web (WWW)*. 1109–1110.

Chaker, J. and Habib, O. 2007. Genre categorization of Web pages. In *Proceedings of the International Conference on Data Mining Workshops (ICDMW)*. 455–464.

Chakrabarti, S., Dom, B., and Indyk, P. 1998. Enhanced hypertext categorization using hyperlinks. In *Proceedings of the International Conference on Management of Data (SIGMOD)*. 307–318.

Chakrabarti, S., van den Berg, M., and Dom, B. 1999. Focused crawling: A new approach to topic-specific Web resource discovery. *Comput. Netw. 31,* 11–16, 1623–1640.

Chen, H. and Dumais, S. 2000. Bringing order to the Web: Automatically categorizing search results. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*. 145–152.

Cover, T. and Thomas, J. 1991. *Elements of Information Theory*. Wiley & Sons.

Dasgupta, A., Kumar, R., and Sasturkar, A. 2008. De-Duping URLs via rewrite rules. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*. 186–194.

Devi, M. I., Rajaram, R., and Selvakuberan, K. 2007. Machine learning techniques for automated Web page classification using URL features. In *Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA)*. 116–120.

Domingos, P. and Pazzani, M. 1997. On the optimality of the simple bayesian classifier under zero-one loss. *Mach. Learn. 29*, 103–130.

Freud, Y. and Schapire, R. 1996. Experiments with a new boosting algorithm. In *Proceedings of the International Conference on Machine Learning (ICML)*. 148–156.

Freudiger, J., Vratonjic, N., and Hubaux, J.-P. 2009. Towards privacy-friendly online advertising. In *Proceedings of the IEEE Web 2.0 Security and Privacy Conference (W2SP)*.

Friedman, J., Hastie, T., and Tibshirani, R. 2000. Additive logistic regression: A statistical view of boosting. *Ann. Statist. 38,* 2, 337–374.

Hastie, T., Tibshirani, R., and Friedman, J. H. 2001. *The Elements of Statistical Learning*. Springer.

Heymann, P., Ramage, D., and Garcia-Molina, H. 2008. Social tag prediction. In *Proceedings of the International Conference on Research and Development in Information Retrieval (SIGIR)*. 531–538.

Jäschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., and Stumme, G. 2007. Tag recommendations in folksonomies. In *Proceedings of the European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*. 506–514.

Joachims, T. 1999. *Making Large-Scale Support Vector Machine Learning Practical*. MIT Press, 169–184. `http://svmlight.joachims.org/`.

Kan, M.-Y. 2004. Web page classification without the Web page. In *Proceedings of the International World Wide Web Conference on Alternate Track Papers and Posters*. 262–263.

Kan, M.-Y. and Nguyen, H. O. T. 2005. Fast Webpage classification using URL features. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*. 325–326.

Koppula, H. S., Leela, K., Agarwal, A., Chitrapura, K. P., Garg, S., and Sasturkar, A. 2010. Learning URL patterns for Webpage de-duplication. In *Proceedings of the International Conference on Web Search and Data Mining*. 381–390.

Manning, C. D., Raghavan, P., and Schütze, H. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

McCallum, A. and Nigam, K. 1998. A comparison of event models for naive Bayes text classification. In *Proceedings of the AAAI Workshop on Learning for Text Categorization*. 41–48.

McCallum, A. K. 1996. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. `http://www.cs.cmu.edu/ mccallum/bow`.

McGuinness, D., Fikes, R., Rice, J., and Wilder, S. 2000. An environment for merging and testing large ontologies. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*. 483–493.

Nigam, K., Lafferty, J., and McCallum, A. 1999. Using maximum entropy for text classification. In *Proceedings of the Workshop on Machine Learning for Information Filtering*. 61–67.

Noy, N. 2004. Tools for mapping and merging ontologies. In *Handbook on Ontologies*, S. Staab and R. Studer Eds., Springer, 365–384.

P, D. and Khemani, D. 2006. Unsupervised learning from URL corpora. In *Proceedings of the International Conference on Management of Data (COMAD'06)*.

POOLA, K. L. AND RAMANUJAPURAM, A. 2007. Techniques for keyword extraction from URLs using statistical analysis. http://www.faqs.org/patents/app/20090089278. US Patent application.

POWER, R., CHEN, J., KARTHIK, T., AND SUBRAMANIAN, L. 2009. Document classification for focused topics. In *Proceedings of the AAAI Spring Symposium on AI for Development*.

QI, X. AND DAVISON, B. D. 2006. Knowing a web page by the company it keeps. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*. 228–237.

QI, X. AND DAVISON, B. D. 2008. Classifiers without borders: Incorporating fielded text from neighboring Web pages. In *Proceedings of the Conference on Research and Development in Information Retrieval (SIGIR)*. 643–650.

QI, X. AND DAVISON, B. D. 2009. Web page classification: Features and algorithms. *ACM Comput. Surv. 41,* 2, 1–31.

SHEN, D., CHEN, Z., YANG, Q., ZENG, H., ZHANG, B., LU, Y., AND MA, W. 2004. Web-Page classification through summarization. In *Proceedings of the Conference on Research and Development in Information Retrieval (SIGIR)*. 242–249.

SHIH, L. K. AND KARGER, D. R. 2004. Using URLs and table layout for Web classification tasks. In *Proceedings of the International Conference on World Wide Web (WWW)*. 193–202.

SILVESTRI, F. 2007. Sorting out the document identifier assignment problem. In *Proceedings of the European Conference on IR Research (ECIR)*. 101–112.

STUMME, G. AND MAEDCHE, A. 2001. FCA-MERGE: Bottom-Up merging of ontologies. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 225–230.

UMBRICH, J., KARNSTEDT, M., AND HARTH, A. 2009. Fast and scalable pattern mining for media-type focused crawling. In *Proceedings of the Knowledge Discovery, Data Mining, and Machine Learning Workshop*.

VEZHNEVETS, A. AND VEZHNEVETS, V. 2005. Modest AdaBoost - Teaching AdaBoost to generalize better. In *Proceedings of the Computer Graphics and Applications Conference (GraphiCon)*. 322–325.

WITTEN, I. H. AND FRANK, E. 2005. *Data Mining: Practical Machine Learning Tools and Techniques* 2nd Ed. Morgan Kaufmann.

ZESCH, T. AND GUREVYCH, I. 2007. Analysis of the Wikipedia category graph for NLP applications. In *Proceedings of the Workshop on TextGraphs: Graph-Based Algorithms for Natural Language Processing (NAACL)*. 1–8.

ZHANG, D. AND LEE, W. S. 2004. Web taxonomy integration using support vector machines. In *Proceedings of the International Conference on World Wide Web (WWW)*. 472–481.

ZHANG, J., QIN, J., AND YAN, Q. 2006. The role of URLs in objectionable Web content categorization. In *Proceedings of the International Conference on Web Intelligence (WI)*. 277–283.

Received June 2009; revised September 2010; accepted December 2010