# Purely URL-based Topic Classification

Eda Baykan
Ecole Polytechnique
Fédérale de Lausanne
Lausanne, Switzerland

Monika Henzinger
Ecole Polytechnique
Fédérale de Lausanne &
Google
Lausanne, Switzerland

Ludmila Marian
Ecole Polytechnique
Fédérale de Lausanne
Lausanne, Switzerland

Ingmar Weber
Ecole Polytechnique
Fédérale de Lausanne
Lausanne, Switzerland

{eda.baykan, monika.henzinger, ludmila.marian, ingmar.weber}@epfl.ch

## ABSTRACT

Given *only* the URL of a web page, can we identify its topic? This is the question that we examine in this paper.

Usually, web pages are classified using their *content* [7], but a URL-only classifier is preferable, (i) when speed is crucial, (ii) to enable content filtering before an (objectionable) web page is downloaded, (iii) when a page's content is hidden in images, (iv) to annotate hyperlinks in a personalized web browser, without fetching the target page, and (v) when a focused crawler wants to infer the topic of a target page *before* devoting bandwidth to download it.

We apply a machine learning approach to the topic identification task and evaluate its performance in extensive experiments on categorized web pages from the Open Directory Project (ODP). When training separate binary classifiers for each topic, we achieve typical F-measure values between 80 and 85, and a typical precision of around 85.

We also ran experiments on a small data set of university web pages. For the task of classifying these pages into faculty, student, course and project pages, our methods improve over previous approaches by 13.8 points of F-measure.

## Categories and Subject Descriptors

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

## General Terms

Experimentation

## Keywords

topic classification, URL, ODP

## 1. INTRODUCTION AND RELATED WORK

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 [4, 6]. However, there are several advantages to do the classification task using *only* URLs, such as speed, classification before downloading a page, classification when the content is hidden in images or the annotation of hyperlinks by topics in a personalized browser. In this paper, we report our findings for web page topic classification only from URL on a large collection of 1.5 million categorized web pages from the Open Directory Project [2].

The idea of classifying web pages into topics using *only* URLs is not new. Most notably, different research groups tried such an approach using the "4 Universities" data set [1], containing 4,167 URLs from the universities of Cornell, Texas, Washington and Wisconsin, as well as other universities grouped as "misc". These pages are classified into student, faculty, course and project page [5]. With the same leave-one-university-out set-up as in [5], we were able to obtain a macro-averaged F-measure of 62.9 using Naive Bayes classifiers with our all-grams features (see below). This is an improvement of 13.8 over the best previously reported F-measure of 52.5 [5]. The problem of *language*, rather than topic classification was investigated in [3].

## 2. TOPIC CLASSIFICATION USING URLS

To apply a machine learning algorithm URLs need to be mapped to numerical feature vectors. We experimented with two methods to extract such features vectors: using whole tokens as features and using letter $n$-grams of the tokens.

*Tokens as features.* Each URL is lower-cased and split into a sequence of strings of letters at any punctuation mark, number or other non-letter character. 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.* For example, `http://www.allwatchers.com/Topics/Info_3922.asp` would be split into the tokens `allwatchers`, `com`, `topics`, `info` and `asp` and represented as a bag of words.

*$n$-grams as features.* This approach splits the URL in the same tokens as the method above. Then letter $n$-grams, i.e., sequences of exactly $n$ letters, are derived from them, and any token shorter than $n$ characters is kept unchanged. For example, the token `allwatchers` gives rise to the 5-grams "allwa", "llwat", "lwatc", "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. We experimented with $n = 4$, 5, 6, 7, and 8.

We also tried an *all-grams* feature set, which was a combination of 4-, 5-, 6-, 7- and 8-grams. For each feature set we experimented with the following machine learning algorithms: Support Vector Machines (SVM), Naive Bayes (NB) and Maximum Entropy (ME).

As a baseline method we tried to imitate the human approach. Given `http://www.attackvolleyballclub.net/` a human would classify it as "Sports" based on the indicator

*WWW 2009,* April 20–24, 2009, Madrid, Spain.
ACM 978-1-60558-487-4/09/04.

word "volleyball". So we compiled a list of words indicative for a certain topic and this list was then used with substring search. To compile such dictionaries, we used all words from the first two levels of the ODP hierarchy. For example, the terms "Basketball" and "Football" are listed one level below "Sports" and were hence included. Some terms, such as "Online" listed under "Games", were not used if they appeared non-topic-specific. The average dictionary size was 19.8 words per topic. The performance of this simple baseline is shown in Table 1.

## 3. EXPERIMENTAL SETUP

For our experiments we used categorized web pages from the Open Directory Project (`http://www.dmoz.org/`). All 15 of its English topics were used (see Table 1) and only the "World" class, which contains pages in non-English languages, was dropped. For each of the 15 topics we put aside a random subset of 1,000 URLs to be used as a test set. The remaining URLs were used to train 15 separate binary classifiers, e.g., one classifier outputs either "Arts" or "non-Arts". In each case we chose an equal number of positive and negative training samples. The negative training samples were further balanced between the 14 "negative" classes, unless one of the 14 classes was too small, in which case we maintained the highest possible degree of balance. Each classifier was evaluated on *all* of the test URLs in the corresponding data set, a total of 15k. 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 $P = (n_+p(+|+))/(n_+p(+|+) + n_-(1 - p(-|-)))$. By default, this would give too much weight to the negative test samples, as $n_- = 14n_+$. Therefore, we downweighted the performance on the negative test samples proportionally by setting $n_- = n_+$, which is equivalent to choosing an equal number of positive and negative samples for evaluation. See [3] for details. All of $P$, $R$ and $F = 2/(1/P + 1/R)$ are multiplied by 100 to lie between 0 and 100. Note that a value of $F = 66$ is trivially obtained by a classifier which always outputs "yes" ($R = 100$, $P = 50$).

## 4. RESULTS

For all machine-learning algorithms all-grams gave the best results and tokens the worst. Using SVM, the macro-averaged F-measure was 75.6, 81.8 and 82.4 for tokens, 4-grams and all-grams respectively. When using all-grams, the macro-averaged F-measure for NB and ME was 81.8 and 82.3 respectively. The reason for the better performance for $n$-grams is that for both tokens and $n$-grams 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. Using tokens the fraction of "empty" URLs containing only previously unknown tokens or general tokens such as "http" or "www" is 32%, while for all-grams it is close to 0%. This means that a token-based classifier cannot do more than randomly guessing in roughly 32% of the cases as it has *no* information to use for the classification. This problem is responsible in large part for the improvement in performance when all-grams are used as they manage to recognize also previously seen *parts* of tokens.

As the differences between the algorithms NB, SVM and ME, were small (0.6 in F-measure), we only report detailed results for SVM, as this performed best on ODP. Table 1 shows the performance breakdown across the 15 ODP categories. Both for the SVM classifier and for the dictionary-based baseline there is usually a problem of recall and not precision. This is mostly caused by "empty" URLs such as `http://www.yuzutree.com`. Such URLs are impossible to classify correctly (in this case as "Business"), unless another URL from the same domain is in the training set. Generally, the high level of performance (80-85 in F-measure and around 85 for precision), though certainly not perfect, came as a surprise. Our experiments also showed that using human-edited short summaries (snippets) *and* the URL leads to an improvement of the F-measure to about 94.

| | | SVM + all-grams | Dictionary |
|---|---|---|---|
| Topic | Size | F / P / R | F / P / R |
| Adult | 36k | 87.6 / 92.5 / 83.3 | 36.7 / 98.1 / 22.6 |
| Arts | 268k | 81.9 / 84.2 / 79.8 | 27.3 / 76.7 / 16.6 |
| Business | 240k | 82.9 / 77.7 / 88.8 | 9.7 / 74.9 / 5.2 |
| Computers | 119k | 82.5 / 78.7 / 86.7 | 16.0 / 80.6 / 8.9 |
| Games | 57k | 86.7 / 82.6 / 91.3 | 47.2 / 96.6 / 31.2 |
| Health | 62k | 82.4 / 87.5 / 77.9 | 21.0 / 94.7 / 11.8 |
| Home | 29k | 81.0 / 92.3 / 72.1 | 19.6 / 78.8 / 11.2 |
| Kids | 37k | 80.0 / 86.4 / 74.4 | 16.9 / 90.6 / 9.3 |
| News | 7.5k | 80.1 / 86.1 / 74.9 | 38.8 / 92.7 / 24.6 |
| Recreation | 108k | 79.7 / 80.8 / 78.6 | 7.5 / 72.5 / 3.9 |
| Reference | 56k | 84.4 / 89.1 / 80.2 | 15.1 / 72.6 / 8.4 |
| Science | 100k | 80.1 / 85.0 / 75.7 | 16.7 / 89.6 / 9.2 |
| Shopping | 100k | 83.1 / 76.9 / 90.4 | 14.7 / 80.3 / 8.1 |
| Society | 241k | 80.2 / 81.8 / 78.6 | 14.4 / 74.3 / 8.0 |
| Sports | 103k | 84.0 / 90.9 / 78.1 | 54.6 / 92.3 / 38.8 |
| Average | 107k | 82.4 / 85.4 / 80.1 | 23.8 / 84.3 / 14.5 |

**Table 1: Size of and performance on the ODP categories when SVM is used with all-gram features. For each topic 1,000 URLs were used as a test set.**

## 5. REFERENCES

[1] The 4 universities data set. `http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/`.

[2] Open directory project. `http://www.dmoz.org/`.

[3] E. Baykan, M. Henzinger, and I. Weber. Web page language identification based on urls. In *International conference on Very Large Data Bases (VLDB)*, pages 176–187, 2008.

[4] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *International conference on Management of data (SIGMOD)*, pages 307–318, 1998.

[5] M. Kan and H. O. N. Thi. Fast webpage classification using url features. In *International conference on Information and knowledge management (CIKM)*, pages 325–326, 2005.

[6] X. Qi and B. D. Davison. Knowing a web page by the company it keeps. In *International conference on Information and knowledge management (CIKM)*, pages 228–237, 2006.

[7] X. Qi and B. D. Davison. Web page classification: Features and algorithms. *ACM Computing Surveys*, 41, 2009. To appear.