Fast Webpage Classification Using URL Features

Min-Yen Kan Hoang Oanh Nguyen Thi

Department of Computer Science, School of Computing
3 Science Drive 2, Singapore 117543
{kanmy,nguyent6}@comp.nus.edu.sg

ABSTRACT

We demonstrate the usefulness of the uniform resource locator (URL) alone in performing web page classification. This approach is faster than typical web page classification, as the pages do not have to be fetched and analyzed. Our approach segments the URL into meaningful chunks and adds component, sequential and orthographic features to model salient patterns. The resulting features are used in supervised maximum entropy modeling. We analyze our approach's effectiveness on two standardized domains. Our results show that in certain scenarios, URL-based methods approach the performance of current state-of-the-art full-text and link-based methods.

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

General Terms: Algorithms, Experimentation.

Keywords: Uniform resource locator, webpage classification.

1. INTRODUCTION

Current webpage classification techniques use a variety of information to classify a target page: the text of the page itself, its hyperlink structure, the link structure and anchor text from pages pointing to the target page and its uniform resource locator (URL). Of this information, a web page's URL is the least expensive to obtain and one of the more informative sources with respect to classification. As the URL is short, ubiquitous (all web pages, whether or not they are accessible or even exist, have URLs) and is largely content-bearing, it seems logical to expend more effort in making full use of this resource.

We approach this problem by considering a classifier that is restricted to using the URL as the sole source of input. Such a classifier is of interest as it is magnitudes faster than traditional approaches as it does not require fetching pages or parsing the text. Our implementation uses a two-step approach, in which a URL is first segmented into meaningful tokens, which are then analyzed as features for classification. We use a recursive, entropy reduction based technique to derive tokens from the URL for the first step. We focus here on the second step: deriving useful features for suitable for classification. A more complete report of these experiments and others are discussed in [2]. These features model sequential dependencies between tokens, their orthographic patterns, length, and originating URI component. A key result is that the combination of quality URL segmentation and feature extraction results in a significant improvement in classification accuracy over baseline approaches.

CIKM'05, October 31-November 5, 2005, Bremen, Germany.
ACM 1-59593-140-6/05/0010.

2. FEATURE INVENTORY

In our system, a baseline segmentation using whitespace delimitation and case change [B] is augmented with entropy-based segmentation [S]. We distill the following features from the URL and discuss why they contribute to better accuracy:

- **URL Components [C] + Length [L]**: A token that occurs in different parts of URLs may contribute differently to classification (www.*ibm*.com vs. smb.net/protocols/*ibm*.html). The absence of certain components can influence classification. URLs that underlie an advertising image often have a long query component. URL length may also influence classification. For example, departmental staff listings are usually not deeply nested, while software drivers usually are. As such, token inventories per component and component lengths are added as features (Rows 2 and 3 in Table 1).

- **Orthographic [O]**: Using the surface form of a token also presents challenges for generalization. For example, tokens *2002* and *2003* are distinct tokens. We add orthographic features that generalize tokens with capitalized letters and/or numbers that differentiate these tokens by their length.

- **Sequential *n*-grams [N] and Precedence [P]**: URL trees [5] showed that token sequences are effective in classification. In their work, a tree rooted at the leftmost token (usually *http*) is created, in which successive tokens are inserted as children. A tree structure emerges after many URLs are inserted. The intuition is that subtrees have similar classification. However, this approach does not generalize over multiple websites, as websites appear as separate subtrees. Recurring patterns in different websites are not captured.

  We note it is the sequential order of nodes (from general to specific) in the URL tree that is of import, and not the rooted path. We thus reverse the order the components within the server hostname, as hostnames are written specific-to-general. We use n-gram token sequences on the resulting URL to capture this phenomenon. Furthermore, consider the sequences *states georgia cities atlanta* and *georgia altanta*. Modeling these as token sequences fails to capture their similarity; as the precedence between *georgia* and *atlanta* are missed. We can capture this by introducing features that model left-to-right precedence between tokens (rows 5 and 6).

3. EVALUATION

Here we apply maximum entropy (ME) based learning [1] on our features on two standardized datasets. We show that our approach compares to or outperforms previous methods, even when only using the URL alone. All results here reported in this section are significant at the 95% confidence level, largely due to the size of the datasets (which are noted in the table captions).

Feature Class (class tag)	http://audience.cnn.com/services/activatealert.jsp?source=cnn&id=203&value=hurricane+isabel
0. Baseline (B)	http audience cnn com services activatealert jsp source cnn id 203 value hurricane Isabel
1. Entropy Reduction (S)	http audience cnn com services activate alert jsp source cnn id 203 value hurricane isabel
2. URI Components (C)	scheme:http extHost:audience dn:cnn tld:com ABSENT:port path:services … ABSENT:fragment
3. Length (L)	chars:total:42 segs:total:8 chars:scheme:4 segs:scheme:1 chars:extHost:8 segs:extHost:1... segs:extn:1
4. Orthographic (O)	Numeric:3 Numeric:queryVal:3
5. Sequential n-grams (N)	com|cnn cnn|audience audience|services services|activate cnn|audience|services ... services|activate|alert|jsp
6. Precedence Bigram (P)	com>services com>activate com>alert com>jsp cnn>services cnn>activate cnn>alert cnn>jsp ... activate>jsp

Table 1: URL feature classes and examples.

In **link recommendation**, the goal is to build a classifier to recommend useful links given a current webpage in a browser. In [5][5], such a dataset was created from 176 users that examined five news web pages. We follow their published experimental procedure to extend their experiments.

Table 2 shows the results of the experiment in which the classifier recommends the best 1, 3, 5, or 10 links on a page with the highest probability of similarity to user clicks on the training pages. The specialized tree learning algorithm (row TL-URL) using their URL features performs best at recommending the single most probable link, but is outperformed on the top 3, 5 and 10 metrics. Better classification is achieved by better URL feature extraction, and outweighs the gains made by using a specialized learner. This is exemplified in rows 2 and 3, where the same learner is used (Support Vector Machines (SVM), with a linear kernel) but using different features.

Table 2: Number of correct recommendations (2 classes, 182K data points). Bolded numbers indicate top performers.

Learner Configuration	Top 1	Top 3	Top 5	Top 10
TL-URL (from [5])	**385**	979	1388	2149
SVM-URL (from [5])	308	839	1268	1953
SVM (our features)	363	996	1456	2412
ME (our features)	365	**1100**	**1682**	**2775**

For **multi-class classification**, we employ the standardized subset of the WebKB corpus (ILP 98 [6]), in which each page is also associated with its anchor text. The task is identical to earlier published experiments: pages are classified as *student*, *faculty*, *course* and *project* pages, and *leave-one-university-out* cross-validation is done. Previous work using the full text have employed SVMs [7], maximum entropy [4], and inductive logic programming [6]. Our results are shown alongside past results.

Performance is measured both by instance accuracy and macro F_1, as both metrics are used previously. Our new URL features perform very well, boosting performance over URL-only previous work by over 30% in the best case, resulting in 76% accuracy. This is impressive as our URL-only method achieve about 95% of the performance of full text methods. Also, our URL features can supplement full text methods, as a small performance gain is observed when the two methods are combined.

Note that our experiment show a best performance of ~78% accuracy using full text in contrast with [4] which showed 92% accuracy. The difference may be due to our use of *leave-one-university-out* cross validation, which we feel is more fair.

Table 3: WebKB performance (4 classes, 4.1K data points). Past results re-printed on top half. 'NR' = not reported.

Learner Configurations [Cite]	Accuracy	Macro F_1
SVM w/ URL (Kan, [3])	--	.338
SVM w/ Full Text (Sun *et al.*[7])	--	.492
SVM w/ Anchor Text (also [7])	--	.582
ME w/ Full Text (Nigam *et al.* [4])	92.08%	--
ME w/ our URL features	76.18%	.525
ME w/ Full Text	78.39%	.603
ME w/ Full Text + our URL features	80.98%	.627

4. CONCLUSIONS AND FUTURE WORK

Given that the URL is a ubiquitous feature of web pages, we study how they can be maximally leveraged for classification tasks. In this report, we concentrate on URL feature extraction and have added features to model URL component length, content, orthography, token sequence and precedence. A full disclosure of these experiments and per-feature class analyses are reported in [2]. Results indicate that these extra features significantly improve over existing URL features and suggest that they may also improve full text methods.

5. REFERENCES

[1] A. L. Berger, S. D. Pietra, and V. J. D. Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39-71, 1996.

[2] M.-Y. Kan and H. O. Nguyen Thi. Fast Webpage Classification Using URL Features. NUS Tech. Rpt. TRC 8/05.

[3] M.-Y. Kan. Web page classification without the web page. In *Proc. of WWW '04*, 2004. Poster paper.

[4] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *IJCAI-99 Workshop on Machine Learning for Information Filtering*, 1999.

[5] L. K. Shih and D. Karger. Using URLs and table layout for web classification tasks. In *Proc. of WWW '04*, 2004.

[6] S. Slattery and M. Craven. Combining statistical and relational methods for learning in hypertext domains. In *8th Int'l Conf. on Inductive Logic Programming*, 1998.

[7] A. Sun, E.-P. Lim, and W.-K. Ng. Web classification using support vector machine. In *4th Int'l Workshop on Web Information and Data Management (WIDM 2002)*, 2002.