A Statistical Approach to URL-Based Web Page Clustering *

Inma Hernández, Carlos R. Rivero, David Ruiz, Rafael Corchuelo
University of Seville
{inmahernandez, carlosrivero, druiz, corchu}@us.es

ABSTRACT

Most web page classifiers use features from the page content, which means that it has to be downloaded to be classified. We propose a technique to cluster web pages by means of their URL exclusively. In contrast to other proposals, we analyse features that are outside the page, hence, we do not need to download a page to classify it. Also, it is non-supervised, requiring little intervention from the user. Furthermore, we do not need to crawl extensively a site to build a classifier for that site, but only a small subset of pages. We have performed an experiment over 21 highly visited websites to evaluate the performance of our classifier, obtaining good precision and recall results.

Categories and Subject Descriptors

I.5.2 [**Pattern Recognition**]: Design Methodology—*Classifier design and evaluation*

Keywords

URL Classification, URL Patterns, Web Page Clustering

General Terms

Algorithms, Experimentation

1. INTRODUCTION

Web page classification has been extensively researched, and several techniques have been applied with successful experimental results. In most cases features for classifying web pages are extracted from the page to be classified, which requires downloading it previously. URLs are very useful when used to classify web pages, as they are relatively small strings (therefore, easy to handle), and every web page possess one [4]. Furthermore, they allow to classify pages without downloading them, which results in a faster performance [1].

In this paper, we propose an statistical technique to extract features from URLs in a not supervised way. These features can be used to build URL patterns that represent the different types of URLs in a site. Then, we can use these patterns to classify pages according to their topic, without the need for downloading them first. We focus on pages from a certain site. Furthermore, due to the statistical nature of our proposal, we are able to achieve good classification results parting from a representative non-labeled training set with a relatively small size.

Site	URLs	Class	P	R	F1
TDG Scholar	11055	Authors	1.00	0.84	0.91
		Hosts	1.00	0.90	0.94
		Papers	0.97	0.50	0.63
Ms Academic	9588	Authors	1.00	0.94	0.96
		Papers	1.00	0.80	0.82
Google Scholar	6247	Citations	1.00	1.00	1.00
DBLP	45697	Authors	0.78	0.72	0.74
Arxiv	33748	Authors	0.83	0.76	0.79
		Papers	0.87	0.79	0.83

Table 1: Evaluation results of academical sites.

Our classifier is based not on the words that a URL contains, but in its format. Proposals that use words as features require the URLs to be written a) in human-readable form and b) using meaningful tokens, that is, using words that can be checked against a topic dictionary. In some cases, and more frequently as web sites evolve, URLs are less human-readable. Therefore, these techniques may not work well when no understandable tokens are found. Our proposal overcomes this problem, as it does not require understandable words in URLs; instead, it tries to find a pattern for each site (which is feasible to find [2]), by means of the URL format. Hence, it is always applicable. More details of our technique can be found at [3].

2. PROPOSAL

The whole URL pattern building process is depicted in Figure 1. First, we define the process to create a training set of URLs from a given site. Then, we calculate some features for each token inside each URL of the training set. Finally, we describe the algorithms used to build a set of patterns parting from the training set, using the previously calculated features.

We focus on websites that allow searching their contents by means of forms, i.e., we focus on the Deep Web [5]. To retrieve URLs from Deep Web sites, we select a set of keyword-based queries and issue them using the forms. Each of the queries is usually answered with a hub page, i.e., a list of results to the query, including a brief description of the result, along with a link to another page with the extended information. Once we have obtained a hubset, we extract and tokenise each URL from its hubs, to build the training set.

*Supported by the European Commission (FEDER), the Spanish and the Andalusian R&D&I programmes (grants P08-TIC-4100, TIN2010-21744)

Copyright is held by the author/owner(s).
WWW 2012 Companion, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1230-1/12/04.

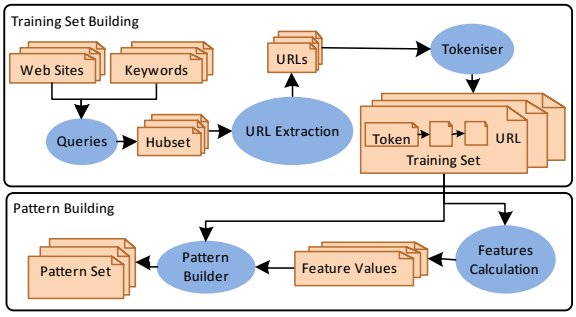

Figure 1: Workflow to build a classifier

Note that we do not need to crawl extensively a website to obtain a training set, but only a small representative subset of pages.

Traditional classification algorithms are based on distances between the different elements. However, it has been noticed that edit distance measures applied to URLs do not seem to provide enough support to classify URLs according to the concept contained in their targets [2], i.e., two URLs separated by a very small distance may provide information about two different concepts, while distant URLs may be related to pages with similar concepts. Therefore, our approach is based on probabilistic features instead to build the pattern set. More precisely, That is, we calculate a feature value for each token in a URL as the relative size of the subset of hubs from a hubset which have at least one URL that shares a common prefix with it up to that token (included), in relation with the total size of the hubset.

These feature values range from 0.0 (least frequent tokens) to 1.0 (most frequent tokens). The most frequent tokens are usually part of the URL patterns, whilst the least frequent tokens are usually parameter values or something similar. To create URL patterns, we check the feature value of each token t in each URL, and decide whether to keep the token as a literal, or to replace it with a more abstract representation (i.e., a wildcard $\star$), depending on its feature value. This decision is based on a statistical analysis of the distribution of feature values in other tokens that occupy the same position as t in other URLs. For example, URL `http://www.amazon.com/Head-First-Java/dp?ie=UTF-8&qid=123` gives pattern `http://www.amazon.com/`$\star$`/dp?ie=UTF-8&qid=`$\star$, that represents all links to product information pages in Amazon.

3. EVALUATION

We performed an experiment to assess our technique. We selected the top 16 more visited sites according to Alexa Web Directory plus five academical sites. On each website, we located search forms and issued queries using a dictionary composed of the most frequent words in different domains and languages. We defined some classes for each website, by generating XPath expressions, and we measured the Precision (P), Recall (R) and F1-measure (F1) of our classifier in a 10-fold cross validation.

The main results of the experiment are presented in Tables 1 and 2. For each site in the experiment, we show the number of pages and URLs that compose the training set, the precision, recall and F1-measure, obtaining a mean precision of 0.95 and a mean recall of 0.85.

Site	URLs	Class	P	R	F1
Amazon	34228	Products	0.83	0.78	0.90
		Reviews	0.93	0.90	0.92
		Authors	0.72	0.54	0.54
Ehow	341	Articles	0.89	0.80	0.84
Anwers	13840	Topics	0.98	0.84	0.90
		Questions	0.98	0.98	0.98
Digg	10935	Authors	0.96	0.94	0.94
		Comments	0.51	0.96	0.66
DailyMail	15485	Authors	0.98	0.90	0.94
Squidoo	6192	Articles	1.00	0.98	0.99
		User Profiles	1.00	0.90	0.94
Torrentz	9704	Files	1.00	1.00	1.00
Guardian	19528	Authors	1.00	0.99	0.99
Archive	20442	Articles	1.00	0.94	0.97
Isohunt	12766	Files	1.00	0.94	0.97
		Comments	1.00	1.00	1.00
Yelp	15754	Businesses	0.99	0.62	0.74
Metacafe	13239	Videos	1.00	0.51	0.66
Etsy	19658	Products	1.00	0.76	0.86
		Stores	1.00	0.98	0.99
BBC	8096	News	1.00	0.76	0.86
Alibaba	39630	Products	1.00	0.86	0.92
Target	69757	Products	1.00	0.88	0.93

Table 2: Evaluation results of the most visited sites.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we present a statistical approach to the problem of classifying web pages in a non-supervised way, relying exclusively in features extracted from the page URL. This approach has many advantages, namely: the user intervention is kept to a minimum, which saves user time; pages are classified for features that are outside them, which avoids having to download a page to classify it; it is a general technique, as it is based on the URL format; and finally, we do not need to crawl extensively a site to build a classification model that works properly, but only small subset of pages. Our experiments confirm that it is useful to classify web pages with promising values for precision and recall.

In the future, we plan to perform a more extensive evaluation. Furthermore, we aim at using the URL pattern building technique to support the automatic publication of legacy websites information in RDF, to integrate this information into the Web of Data [6, 7].

5. REFERENCES

[1] E. Baykan, M. R. Henzinger, L. Marian, and I. Weber. Purely URL-based topic classification. In *WWW*.

[2] L. Blanco, N. Dalvi, and A. Machanavajjhala. Highly efficient algorithms for structural clustering of large websites. In *WWW*, 2011.

[3] I. Hernández, C. Rivero, D. Ruiz, and R. Corchuelo. A tool for link-based web page classification. In *CAEPIA*.

[4] M.-Y. Kan and H. O. N. Thi. Fast webpage classification using URL features. In *CIKM*.

[5] J. Madhavan, L. Afanasiev, L. Antova, and A. Y. Halevy. Harnessing the deep web: Present and future. *CoRR*, 2009.

[6] C. R. Rivero, I. Hernández, D. Ruiz, and R. Corchuelo. Generating SPARQL executable mappings to integrate ontologies. In *ER*, 2011.

[7] C. R. Rivero, I. Hernández, D. Ruiz, and R. Corchuelo. On benchmarking data translation systems for semantic-web ontologies. In *CIKM*, 2011.