An Experiment to Test URL Features for Web Page Classification

Inma Hernández, Carlos R. Rivero, David Ruiz, and José Luis Arjona

Abstract. Web page classification has been extensively researched, using different types of features that are extracted either from the page content, the page structure or from other pages that link to that page. Using features from the page itself implies having to download it before its classification. We present an experiment to proof that URL tokens contain information enough to extract features to classify web pages. A classifier based on these features is able to classify a web page without having to download it previously, avoiding unnecessary downloads.

1 Introduction

Web page classification has been extensively researched, using different types of features. In this context, features can be extracted from different sources, including: the web page content [11], [13], [14], like the bag of words the page contains or the number of images in it; the page structure [1], [2], [5], [15], like the distance between its different components; or from other pages that link to the page [7], [8], like the words in the anchor text.

Our hypothesis is that, in many web sites (excluding URL-friendly sites), the set of tokens extracted from a page URL contains enough information to classify that page. That is, we can calculate some features based exclusively on the information contained in those tokens, and those features are usable to build a web page classifier.

Inma Hernández · Carlos R. Rivero · David Ruiz
University of Seville
e-mail: {inmahernandez,carlosrivero,druiz}@us.es

José Luis Arjona
University of Huelva
e-mail: jlarjona@gmail.com

*Supported by the European Commission (FEDER), the Spanish and the Andalusian R&D&I programmes (grants TIN2007-64119, P07-TIC-2602, P08-TIC-4100, TIN2008-04718-E, TIN2010-21744, TIN2010-09809-E, TIN2010-10811-E, and TIN2010-09988-E).

We have observed that in those sites, for all URLs there are mainly two classes of tokens: on one hand, we have tokens whose appearance in a URL depends on the request made by the user to obtain that page. For example, URL `http://scholar.google.com/scholar?q=java` is obtained after issuing a query to Google Scholar site, and token "java" depends on the particular query. On the other hand, there are tokens that do not depend on the user request; instead, they are always part of the URL for each kind of URLs. In the former example, the token "scholar" is always included in any URL obtained after issuing a query in this site, in the same position.

Considering exclusively the latter type of tokens, we are able to build URL prototypes, that is, strings that represent a collection of URLs with a similar format. Each URL prototype has a different format, and usually, points to a different type of page, hence if we build a collection of prototypes for the site, each one representing a different type of pages, we are able classify any given page, by comparing its URL with the different prototypes, and finding the best match.

In this paper, we propose an experiment to calculate features for tokens in URLs to test our previous hypothesis, and confirm our observation. We define features for tokens that distinguish between these two different classes of tokens, and we perform an statistical analysis to test whether these two classes of tokens are distinguishable in every web site. Once we justify this fact, we can use these features to build the former URL prototypes, and use them to classify web pages.

In comparison with the other types of features that can be used to classify web pages, token features can be calculated without having to download the web page, just by analysing its URL, which avoids unnecessary downloads.

The rest of the article is structured as follows. Section 2 describes the related work; Section 3 presents the experiment performed to justify our hypothesis; finally, Section 4 lists some of the conclusions drawn from the research and concludes the article.

2 Related Work

We have identified some proposals in the area of web page classification using features extracted from URLs. Some of them are supervised, like [12], [16], [4], [3], while [6] and [15] are not supervised.

Kan et. al. [12] presented one of the first approaches to web page topic classification using only URLs. Their proposal consisted on tokenising URLs into tokens, and then extracting features from those, like the words they contain, their type or length, amongst others. These features are used to build a Maximum Entropy classification model. They work with a subset of the URLs to build a classification model, so they perform a reduced crawling to obtain their training set. However, this is a supervised proposal that requires a labeled set of training URLs. Furthermore, they aim at classifying pages belonging to more than one site, and relies on words being human-understandable, which is not always true.

Vidal et. al [15] propose a technique to analyse a single site, and automatically find pages that are similar to an example page that is given. To achieve so, the site is mapped, and URL patterns are generated for those pages that lead (directly or eventually) to those pages. To detect similarity between pages, the Tree Edit Distance is used. To build the training set, they have to previously crawl the entire site, download each page and then process them, which takes a significant amount of time. Also, it does not classify a page according to its content, but to its structural similarity to the given page. That means that pages containing information about different topics may be classified as the same class.

Zhu et. al. [16] propose a link classifier, instead of a web page classifier, although we include it in this framework due to their analysis of link features. They aim at classifying links according to their function inside the site. They propose a taxonomy of predefined link classes, depending on the function that each link performs in the page, namely: navigating, indexing, citing, recommending and advertising. They analyse links to extract visual, content and structural features, amongst others, and they build two types of supervised classifiers: SVM and decision trees. It is supervised proposal in which the classes are predefined in a rigid taxonomy. Furthermore, their goal is not directly related to information extraction, like ours, so they are not topic oriented; instead, their classifier may be used for user link recommendation.

Baykan et. al. [3] proposed a classifier that is similar to [12]. They tokenise URLs as well to extract features, but they apply different supervised classification algorithms, like SVM, Naïve-Bayes or Maximum Entropy, and compare the results. These algorithms need to be fed a list of words indicative for every topic that is to be classified. On a previous work [4], the authors used the same idea, but this time in order to classify pages according to their language, instead of their topic. Just like [12], it is a supervised technique that classifies pages from different sites.

Finally, Blanco et. al. [6] consider that every site is created by populating HTML templates with data from a database. Their goal is to cluster web pages so that each cluster contains pages following a certain template. They observed that URLs generated from the same template have a similar pattern, just like pages generated from the same template contain similar terms, so they proposed an algorithm for unsupervised classification that combines web page contents and its URL as features, by means of the minimum description length method (MDL). They require a large training set, so they crawl the entire site in their experiments. Furthermore, to improve the classification efficiency, features from the page itself are included in addition to the link-based features, which means that it must be downloaded previously.

In contrast to the former, our proposal is not supervised, and it does not require to crawl the whole site to build the classification model, as in [15] or [6]. We use a small subset of pages and URLs from the site, and we apply a statistical technique to extract classification features from those URLs.

3 Experiment

Next, we present the experiment to justify our hypothesis. First, we describe the process to obtain a set of tokens from a given web site and calculate features for those

tokens. Then, we show the results of applying the former process to five academical sites, and we analyse those results.

3.1 Features Calculation

To calculate the feature value of each token in a URL, we consider not only the token, but the whole sequence of tokens from the beginning of the URL up to the token itself, which we call the token prefix. For each token in a URL, we define its feature value as the probability of the token prefix appearing in other pages from the same web site. Following the frequentist approach, we estimate that probability using the relative frequency of appearance of the token prefix. Therefore, we need to obtain a sample of pages, extract the URLs of all the links inside them, and finally calculate the relative frequency of their token prefixes. From here onwards, we will represent the feature value of a token X by F_X.

To obtain the sample, we perform a lightweight crawling over each site, to retrieve a representative sample of its pages. In this paper, we focus on dynamic pages, that is, pages that are behind a form that must be filled in and submitted. The result of those submissions is usually a hub, 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. Therefore, we should fill in the forms with values such that the resulting query yields as many results as possible. In that case, the hub obtained in response contains a large number of links, and our sample is more representative.

From those hubs, we extract all links, and we decompose them into tokens. We use a tokeniser based on the standard for URIs defined in RFC 3986, according to which a URI is composed of a protocol (e.g., http, ftp or https); an authority, also known as domain name (e.g., google.scholar.com); a path, which is a sequence of segments separated by a slash charater ('/'); optionally followed by an interrogation mark ('?') and a query string, which is a sequence of parameters separated by the ampersand character ('&'), being each parameter of the form name '=' value.

Then, we calculate for each token, its feature value F_X, estimating the probability of the token prefix appearing in other hubs from the same site, by means of its relative frequency.

3.2 Experimental Results

We have chosen five academical sites: Google Scholar, Arxiv, Microsoft Academic Search, TDG Scholar and DBLP. For each of them, we performed the lightweight crawling to obtain a test set of hubs. In order for this set to be representative from the site, we chose a sample size that was sufficiently large, retrieving 100 hubs per site. Then, we extracted all links in those hubs and tokenised them, obtaining the test sets shown in Figure 1.

For each token, we calculate their feature value by estimating its probability, as described previously. As an example, in Figure 2 we show the feature values for some of the URLs extracted from TDG Scholar. We graphically represent URLs, tokens

Site	# Hubs	# Links	# Tokens
Arxiv	100	33748	121362
DBLP	100	30749	157295
Google Scholar	100	6247	38375
Ms Academic Search	100	9588	107888
TDG Scholar	100	11055	91493

Fig. 1 Experimental test sets

and their features using a tree-like structure, in which each node represents a different token, and a token t_1 is a son of another token t_2 when t_2 follows t_1 in a given URL. Every token is assigned a label, which indicates its node name, its feature value, and the token text. Each path from the tree root to a leaf represents a single URL.

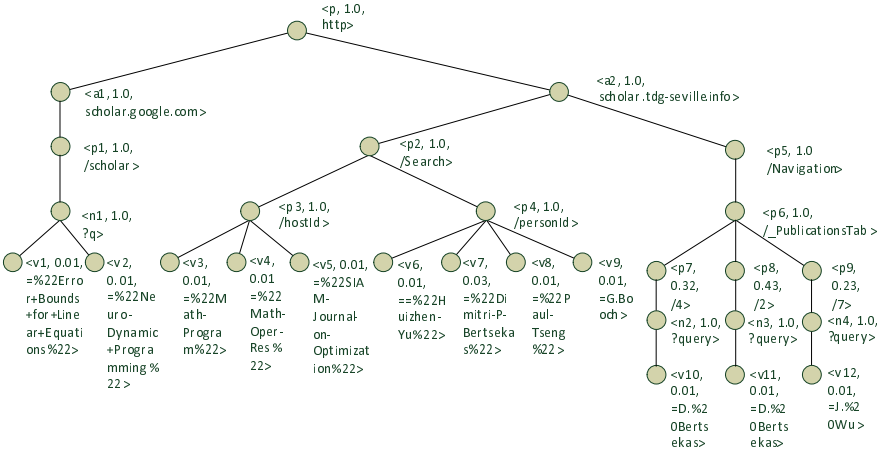
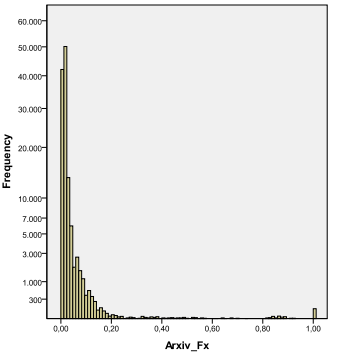


Fig. 2 Links extracted from TDG Scholar

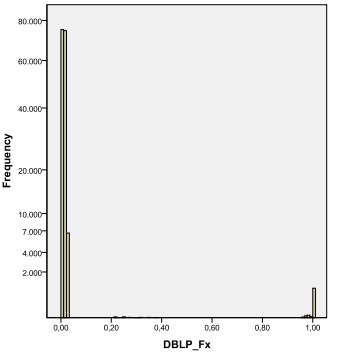
In order to get an idea of how the feature values are distributed, we present some descriptive statistics in Figure 3. For each site, we calculate its mean, standard deviation, minimum and maximum values and quartiles.

Variable	N	Mean	Std. Dev	Min	Max	Percentiles		
						25th	50th	75th
Google_Fx	38375	0.04	0.13	0.01	1.00	0.01	0.01	0.02
DBLP_Fx	157295	0.02	0.07	0.01	1.00	0.01	0.01	0.01
MsAcademic_Fx	107888	0.02	0.07	0.01	1.00	0.01	0.01	0.01
TDG_Fx	91493	0.02	0.07	0.01	1.00	0.01	0.01	0.02
Arxiv_Fx	121362	0.02	0.04	0.01	1.00	0.01	0.01	0.02

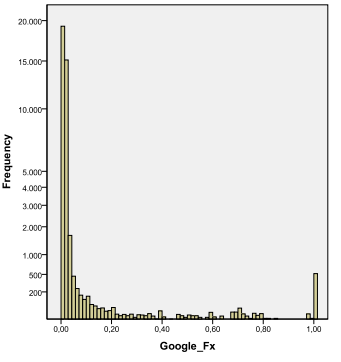
Fig. 3 Descriptive statistics for the experimental sites



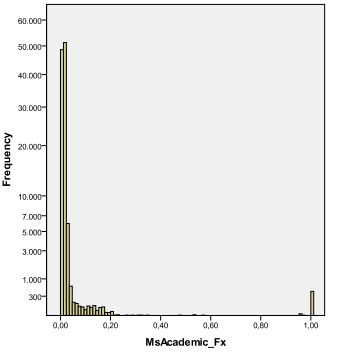
(a) Arxiv



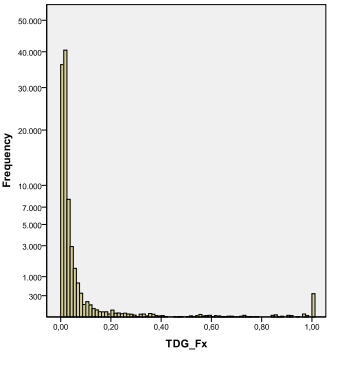
(b) DBLP



(c) Google Scholar



(d) MS Academic Search



(e) TDG Scholar

Fig. 4 Histograms of feature values F_X for each site in the experiment

Just by looking at these statistics, we can see that the distributions are right-skewed, and that the vast majority of data are concentrated near the left extreme of the distribution (around 0).

For each site, we obtained the histograms in Figure 4, in which the Y-axis is expressed in power scale. We corroborate that we have two types of tokens, depending on their F_X value: on one hand we have a large share of data around 0, and on the other hand we have a small but significant share of data in the other extreme of the distribution, around 1. The first kind of tokens are those that appear rarely in other hubs different than that from which we obtained it. That is, their appearance depends on the query we made to obtain the hub, so they are similar to parameter values that are passed on to the server to create the hub. On the contrary, tokens around 1 are always present in every possible hub we obtain from the site, and they do not depend on the query.

As a consequence, we can use the information of the features to distinguish between those types of tokens, and create URL prototypes to classify Web pages using only their URL.

4 Conclusions and Future Work

In this paper, we present an experiment to proof that information contained in URL tokens is sufficient to classify those URLs, and to create classes of URLs.

We have developed features that exploit the information contained in the tokens. These features are probabilities estimators, based on token frequencies obtained from the experiment. The feature values histograms show that we can distinguish between two type of tokens, depending on their feature values, being some of them dependant on the query made to obtain the URL, whilst others are independent from it, and they always belong to URLs of the same type. As a conclusion, URL tokens indeed contain enough information to build URL prototypes, and therefore, to classify web pages, with the advantage of not having to download the page previously.

We have identified other proposals in the literature that aim at using the information contained in URLs to classify web pages. The advantages of our features in comparison with other proposals are 1) user intervention is kept to a minimum, which saves an important asset as is user time; 2) pages are classified for features that are outside them, which avoids having to download a page in order to classify it; 3) it is language independent, since it is based on the URL format regardless of the particular words or sequences of characters that make each token; 4) it does not require links to be surrounded by words useful for classification; and 5), we do not need to crawl extensively a site in order to build a classification model that works properly, instead, we perform a lightweight crawling that retrieves a small subset of pages. Because of the statistical nature of the proposal, we can be confident that the classifier is as accurate as it would be in case it had been built using the whole set of pages.

In the future, we plan to build a web page classifier using these features. We provide some insight about how to build such a classifier in [10]. Furthermore, we

believe such a classifier can be used to improve web crawlers efficiency, which we expose in [9]. We must note that we must analyse how to apply our features to the so called friendly URLs, which do not fit our hypothesis.

References

1. Arasu, A., Garcia-Molina, H.: Extracting structured data from web pages. In: SIGMOD, pp. 337–348 (2003)
2. Bar-Yossef, Z., Rajagopalan, S.: Template detection via data mining and its applications. In: WWW, pp. 580–591 (2002)
3. Baykan, E., Henzinger, M.R., Marian, L., Weber, I.: Purely URL-based topic classification. In: WWW, pp. 1109–1110 (2009)
4. Baykan, E., Henzinger, M.R., Weber, I.: Web page language identification based on URLs. PVLDB 1(1), 176–187 (2008)
5. Blanco, L., Crescenzi, V., Merialdo, P.: Structure and semantics of Data-IntensiveWeb pages: An experimental study on their relationships. J. UCS 14(11), 1877–1892 (2008)
6. Blanco, L., Dalvi, N., Machanavajjhala, A.: Highly efficient algorithms for structural clustering of large websites. In: WWW, pp. 437–446. ACM, New York (2011)
7. Cohen, W.W.: Improving a page classifier with anchor extraction and link analysis. In: NIPS, pp. 1481–1488 (2002)
8. Fürnkranz, J.: Hyperlink ensembles: a case study in hypertext classification. Information Fusion 3(4), 299–312 (2002)
9. Hernández, I., Sleiman, H.A., Ruiz, D., Corchuelo, R.: A Conceptual Framework for Efficient Web Crawling in Virtual Integration Contexts. In: Gong, Z., Luo, X., Chen, J., Lei, J., Wang, F.L. (eds.) WISM 2011, Part II. LNCS, vol. 6988, pp. 282–291. Springer, Heidelberg (2011)
10. Hernández, I., Rivero, C.R., Ruiz, D., Corchuelo, R.: A Tool for Link-Based Web Page Classification. In: Lozano, J.A., Gámez, J.A., Moreno, J.A. (eds.) CAEPIA 2011. LNCS, vol. 7023, pp. 443–452. Springer, Heidelberg (2011)
11. Hotho, A., Maedche, A., Staab, S.: Ontology-based text document clustering. KI 16(4), 48–54 (2002)
12. Kan, M.-Y., Thi, H.O.N.: Fast webpage classification using URL features. In: CIKM, pp. 325–326 (2005)
13. Pierre, J.M.: On the automated classification of web sites. CoRR, cs.IR/0102002 (2001)
14. Selamat, A., Omatu, S.: Web page feature selection and classification using neural networks. Inf. Sci. 158, 69–88 (2004)
15. Vidal, M.L.A., da Silva, A.S., de Moura, E.S., Cavalcanti, J.M.B.: Structure-based crawling in the hidden web. J. UCS 14(11), 1857–1876 (2008)
16. Zhu, M., Hu, W., Wu, O., Li, X., Zhang, X.: User oriented link function classification. In: WWW, pp. 1191–1192 (2008)