

Topical Host Reputation for Lightweight Url Classification

Aleksander Kolcz*

Geoff Hulten†

Jakub Szymański‡

Abstract

Classification of urls into topical categories is an important task of data mining and information filtering. In many applications the task needs to be performed with minimal information, which usually means just the url itself. While for some topics using the url information is surprisingly effective, there is still a substantial loss in accuracy when compared to basing the classification on full web page content. In this work we stipulate that the basic url-based approach can be significantly improved by taking the web-graph information into account, and in particular by precomputing the topical likelihood for the known hosts. While graph mining is computationally intensive, it is performed off-line and the url classification process simply consumes its results, which comes down to a simple table lookup operation. We demonstrate that the proposed approach not only greatly improves on using url-strings as sources of features but also outperforms techniques utilizing full html content. While the set of categories considered in our experiments is not exhaustive, the results are highly encouraging and the proposed approach could be potentially beneficial in a variety of different topic settings.

1 Introduction

In many important applications it is desired to classify urls into topical categories without having access to the underlying web-page content. These include: web traffic/search log analysis, focused crawling and preventing access to questionable/adult/malicious content. In such tasks one either cannot spare the bandwidth needed to fetch the url content or the very objective may be to stop the browser from downloading undesirable web pages in the first place. Several authors considered the question to what an extent topical web page classification can be solved by using just the url string alone [8][9][2]. It became apparent that in many cases a url is semantically related to the underlying content and encouraging results have been reported, although it is not yet clear how the descriptiveness of urls varies among different categories. Importantly, url-only approaches possess the

advantage of being lightweight, as one can evaluate a url string against a large number of models in a very short amount of time. Thus even if web page content is potentially within reach, mining url only information may be preferable due to the scalability of such an approach. There is a loss in accuracy compared to using full html, however since while hypertext can be noisy, not having access to a web page does not make the classification process any easier.

Even when web page content is available, topical classification of hypertext can be challenging, much more so than in the case of classifying well-structured news articles, see for example [11]. Fortunately, several studies have shown that web page classification performance can be significantly improved by considering the web-graph linkage information, i.e., who a page links to and who links to a given page [5][12]. It appears that by taking the web graph neighborhood into account it is possible to overcome the noisiness of the information represented by the web page alone. Given the success of such graph-based approaches it is natural to consider to what an extent they would be useful if one considered classifying urls rather than web pages. We explore this question in the present work, but with a strong emphasis placed on maintaining the original appeal of the url-based approach. We insist that the massive amount of information offered by the web graph is processed just once, with subsequent computation stages utilizing the results of the initial mining effort and thus being effectively lightweight and suitable for online applications. We also tackle the problem of handling “fresh” urls, i.e., those that were not present in the original graph. Since no web graph is ever complete, and the web is growing at a fast pace, an effective use of the web graph needs to consider supporting the “out of graph” urls as well.

In particular, we propose to effectively compress the web-graph at the host level and use this information to augment a url classifier by means of topical host-level reputation. We demonstrate that this leads to a substantial and significant increase in classification performance, while keeping the overall model compact and manageable. While host-level reputation is shown to be useful, the number of hosts can still be quite large, especially if one considers this approach for large taxonomies. We propose to exploit the web hierarchy

*Microsoft Bing.

†Microsoft.

‡Microsoft Bing.

to compress such models further without incurring noticeable loss in their predictive accuracy.

2 Graph-enhanced Url Classification

Consider the graph $G(V, E)$, where vertices in V correspond to urls and an edge $e(i, j)$ signifies a directed link from v_i to v_j . We assume a class structure $\mathcal{C}$ over the vertex set such that vertex v_j belongs to one or more classes in $\mathcal{C}$. For simplicity, we will assume just a binary classification problem (i.e., two classes: C and $\bar{C}$).

With each vertex there is associated a set of content features. At the minimum the set contains the url string itself. It may also contain the web page title and keywords, or even the full page content. Unless otherwise noted, we will assume the minimal representation, which is consistent with the goal of lightweight url classification. Given a labeled set of url vertices, the objective is to accurately label all graph vertices. If the edge information is not available, the problem is reduced to stand alone classification of urls, as studied by Kan and Min-Yen [8] as well as by Baykan *et al.* [2]. A variant of this setting is represented by augmenting each url by extra content-based features (e.g., corresponding to the page title and/or meta-keywords). When the extra content corresponds to the actual web page, one arrives at bi-modal classification with the two modes represented by urls and html content (as studied in [8] and [1]). When additionally full edge linkage is available, the problem then becomes one of supervised or semi-supervised classification on the web graph, which has been studied for nodes represented by html content [12].

More generally, given a set of labeled data, D , we seek to estimate $P(C|v, G)$ for each vertex v . Although one could simply seek to estimate the class labels directly while minimizing some penalty function, casting the problem as one of posterior probability estimation is more flexible since it allows to directly apply Bayesian decision making and accounts for the possibility of misclassification costs changing over time. Typically, a Markov assumption is taken, whereby the class of a vertex is independent of other vertices given its immediate neighborhood, i.e., $P(C|v, G) = P(C|v, N(v, G))$, which has been shown to lead to good results in practice [5][12].

3 The Host-based Topical Reputation Model

Aside from the computational issues involved in classifying all vertices in G (i.e., the whole web-graph snapshot), one still faces the problem of fresh urls that were originally not part of G , but need to be classified on the fly. It would appear that since in such cases $N(v, G) = \emptyset$, the empty graph neighborhood reduces to having just the information about the url itself, i.e., $P(C|v, N(v, G)) \equiv P(C|v)$. However, it is quite likely

that the url v belongs to a web host that had some representation in G .

While urls come in and out of existence quite often, the hosts serving them are more stable. Let $H : v \rightarrow H(v)$ describe a function uniquely assigning url v to its host. A host is an url in its own right but can also be seen as a meta-url, combining all underlying urls, and the connectivity of a host with other hosts reflects the connectivity of its urls. The host graph $G_h(V_h, E_h)$ is thus directly derived from the web graph and its edges are naturally weighted, such that the edge $e_h(i, j)$ receives a weight equal to the number of urls on host i that link to urls on host j

$$e_h(i, j) = \sum_{u, v \in G: H(u)=i \wedge H(v)=j} e(u, v)$$

Many hosts are topically coherent and can be labeled to belong to a certain class. Thus the techniques that have been proposed in the literature for classifying web pages can be applied to the problem of classifying web hosts. E.g., following the approach of [12] one can assign to each host h an estimate of $P(C|h, N(h, G_h))$. An important advantage of working with the host graph rather than the web graph though is that the host graph is much smaller.

When hosts and their graph connectivity are explicitly accounted for, the modeling objective is transformed from the estimation of $P(C|v, N(v, G))$ to the estimation of

$$(3.1) \quad P(C|v, N(v, G), h, N(h, G_h)) \quad \text{where} \quad h = H(v)$$

A key observation is that the neighborhood of host h in G_h provides context not just to h itself by also to the urls hosted by h . While this context is not url-specific, and by definition is shared by all urls of the same host, we hypothesize that it is likely to provide useful information as far as the class membership of urls is concerned. We view $N(h, G_h)$ as coarse approximation of $N(v, G)$ for $H(v) = h$, which leads as to the reformulation of eq. (3.1) as

$$(3.2) \quad P(C|v, N(v, G), h, N(h, G_h)) \approx P(C|v, h, N(h, G_h))$$

Thus we propose to estimate the class of a url based on the url content features as well as the connectivity of its host within the host graph. Using the Bayes formula, eq. (3.2) can be written as:

$$P(C|v, h, N(h, G_h)) = P(C) \frac{P(v, h, N(h, G_h)|C)}{P(v, h, N(h, G_h))}$$

Although not strictly true, we will assume that urls and hosts are independent given the class label. The

validity of this assumption can be supported by the following reasoning. A url can be descriptive of the class of a web page if its path components contain certain highly predictive keywords (e.g., *sex* if we are trying to model the adult category). But this does not have to be the case, and the url path components might simply be random strings of characters. On the other hand, the host model aggregates many different types of information, but it is possible that due to parsing errors, feature-extraction, or heterogeneity of the host's contents, its estimate of the class membership may be inaccurate when applied to a particular url. As long as indicative features are present in the url itself, however, there still exists a chance of classifying it correctly. We will therefore assume that

$$P(v, h, N(h, G_h)|C) = P(v|C) P(h, N(h, G_h)|C)$$

A similar assumption was taken in [12] where $P(C|v, N(v, G), h, N(h, G_h))$ was formulated as

$$P(C|v, N(v, G)) \sim P(C|v)P(C|N(v, G))$$

Although no theoretical justification was given, the authors reported that modeling $P(C|v)$ and $P(C|N(v, G))$ separately led to better classification performance, possibly due to a reduced risk of overfitting.

In the two-class context it is often more convenient to use posterior odds, or logs thereof, rather than posterior class probability estimates. The odds of url v belonging to class C can be expressed as

$$\text{odds}(C|v, h) = \frac{P(C|v, h)}{1 - P(C|v, h)} = \frac{P(C)}{P(\bar{C})} \frac{P(v|C)}{P(v|\bar{C})} \frac{P(h|C)}{P(h|\bar{C})}$$

which can also be written as

$$\text{odds}(C|v, h) \sim \text{odds}(C|v) \text{odds}(C|h)$$

and

(3.3)

$$\log(\text{odds}(C|v, h)) = \text{const} + \log(\text{odds}(C|v)) + \log(\text{odds}(C|h))$$

This formulation is particularly convenient if both the host and url classifiers are modeled by logistic regression, since then the scores produced by the classifiers can naturally be cast as log odds and one ends up simply summing up the scores of the two classifiers in order to obtain to composite score of the graph-enhanced model. This is the approach we take in this work, but it is certainly possible to follow any other classifier combination regimen, e.g., by learning a url model with the host reputation representing an extra feature, or by learning a meta-model with the input features corresponding to the scores of the url and host models. Thus although

we explicitly model $\log(\text{odds}(P(C|h, N(h, G))))$, we refer to it as host reputation, where higher/positive reputation values indicate a better agreement with class C and lower/negative reputation values indicate a lower chance that a host belongs to class C . The successive approximations described in this section are shown in Figure 1.

Note that an extreme way of using the host graph approximation in url classification would be to take the assumption that to the extent that topicality of a host can be determined, it is shared by all urls belonging to this host. While this may indeed be true for cases where the host reputation is very high or very low (and then it would be very similar to the white/black listing approach), it is rather unlikely to be universally the case, since some web sites collocate diverse types of content.

3.1 Modeling of $P(C|h, N(h, G_h))$ Our approach to building the host graph topic model is based on the methodology used in [12]. To begin with, we assume the existence of several elementary models: for classifying web pages and url strings and, more generally, to score them such that higher scores imply higher probability of a page/url to belong to the class of interest. The process of generating a host-specific feature vector is as follows:

1. Given the web graph, we score each page with the page-level classifier and aggregate the scores at the host level, so that each host is assigned a feature vector corresponding to the minimum, maximum, mean and standard deviation of the scores of the page level classifier. In the event that there are multiple page-level models, we concatenate their feature vectors. The resulting feature vector captures the topical coherence of a host.
2. Next, we use the url-level model to score the urls that a host links to, as long as they belong to hosts other than itself (empirically, for most hosts the majority of outlinks point to the same host and such self references are here ignored). The scores are aggregated similarly as in the case of the page level scores and appended to the host feature vector. The new elements capture the topical coherence of external pages linked from a given host. It is expected that a host with strong topicality will link primarily to pages on a similar topic. Steps 1 and 2 are illustrated in Figure 2. At the end of step 2, as a by-product we have converted the web graph to a host graph.
3. Next, we use the host-level graph and produce extra aggregation features based on the inlink and

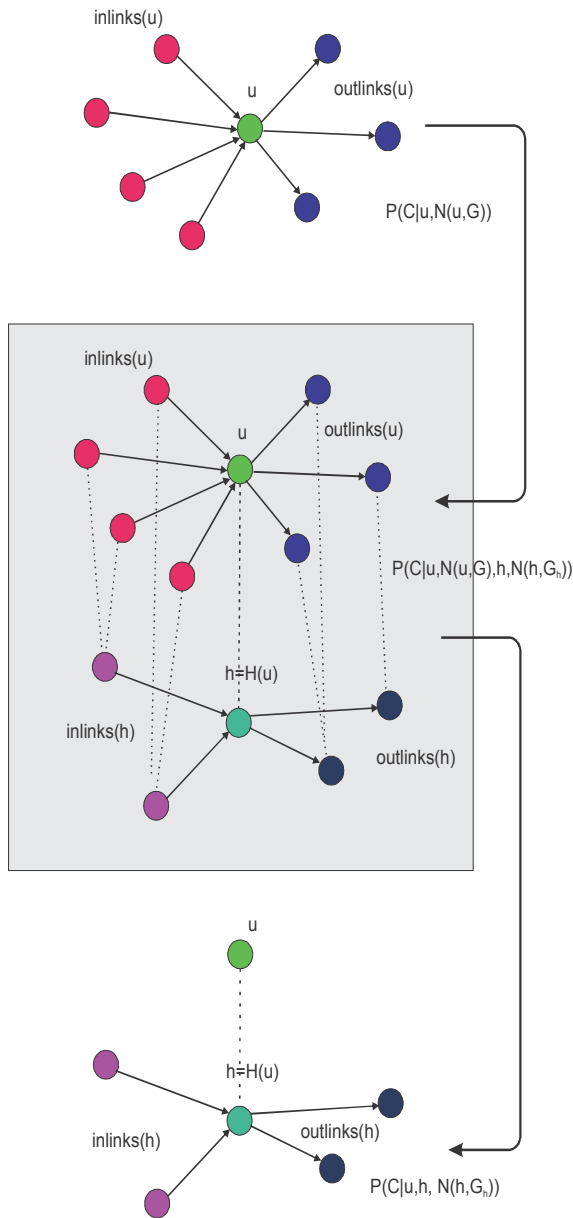


Figure 1: Accounting for host information transforms the web graph neighborhood of a url (top) into a combination of web-graph and host-graph neighborhoods (middle) and its approximation, which uses just the host-graph neighborhood (bottom).

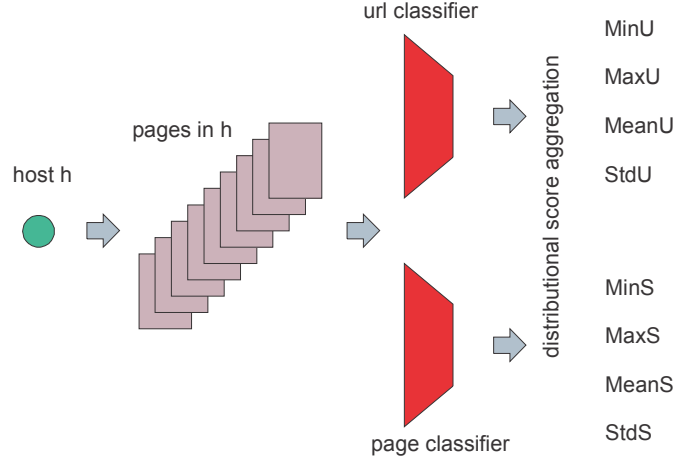


Figure 2: The process of generating host content features. The results of running web page and url classifiers on the urls belonging to the host are aggregated to extract basic statistics of the score distribution.

outlink connections of each host. In particular, we take the means of the page scores and outlink url scores for each host and, based on the host-graph connectivity, we compute their means aggregating the inlinks and outlinks of each host. The aggregation is weighted such that an edge between two hosts is associated with the count of pages linking from one host to the other. We append the resulting features to the feature vector of each host. The new features capture the host-graph topical neighborhood coherence of each host. We expect that hosts with strong topicality will be linked to by hosts with similar topicality and will link out to hosts with similar topicality.

4. Finally, we augment the feature vector by the count of pages present in the host, as well as the counts of inlink and outlink hosts. An example feature vector, with descriptions is shown in Table 1. Note that the particular way of aggregating web graph and host graph information used in this work is certainly not definitive and other choices could lead to similar results.

Once each host has been associated with its feature vector, using a set of labeled training examples we learn a logistic regression model to predict the class membership of a host. Since the features chosen to represent hosts are likely to be correlated with one another, the learner should be fairly insensitive to such correlations. Logistic regression copes well with intercorrelated features and it also naturally calibrates its output to pro-

vide posterior class-membership estimates. Since different features have different ranges and distributions, we discretized each feature using the supervised MDL-based technique [7], and the logistic regression uses binary features only. The resulting model is then applied to hosts for which the class membership is unknown, for which it provides the estimate of $P(C|h, N(h, G_h))$.

A semi-supervised learning regimen can be employed to try to improve its initial performance. To this end, during host graph aggregation one can add additional features capturing for each host the reputation score distribution among its inlink and outlink hosts. One can proceed for several iterations in such manner, each time using the scores from the previous iteration to produce new aggregate features.

4 Making the Host Reputation List More Concise

While web hosts are certainly much less numerous than urls, the size of a host list can still be substantial and contain over 100 million entries. This may be resource intensive for some applications, especially if reputation for multiple topic categories is needed. In this section we consider two possible approaches to reducing the list size.

4.1 Topical-coherence aggregation of the host graph The web is hierarchically organized, which is reflected in the url structure, with pages on a single host typically sharing topical focus and with multiple hosts within a single domain/site also being topically related. It is often the case that a domain corresponds to a single commercial entity, for which different hosts represent specialization within a website (e.g., one host may be serving html pages, while another may be responsible for image/video content). This naturally hierarchical structure motivates us to look for ways of maximally collapsing the reputation information while still maintaining adequate precision. E.g., if `domain.com` has three hosts: `a.domain.com`, `b.domain.com` and `c.domain.com`, all of more or less the same nature, it is expected that for all these hosts the reputation levels will be close to one another. If so, the granularity of a domain, rather than a host, appears to be more appropriate in such cases. Of course, this will not be applicable in all cases. For example, blog-hosting domains map host names to individual blog authors who can generate content on a diverse set of topics.

Our compression strategy is based on greedy pruning of the host graph. The graph can be viewed as a set of disjoint trees corresponding to top-level domains (e.g., `.com` or `.net`). For each tree we propagate the scores of leaf nodes towards the root of the tree, where

at each step the mean and standard deviation of scores for the parent node are computed. If score dispersion among the leaves of a given node is low enough, the leaves are rolled up, with the parent node becoming a leaf node in the new tree and with its reputation corresponding to the mean reputation of the original leaf nodes. The process continues until no further steps are possible, which is controlled by a standard deviation threshold. When the original host list is compressed in this fashion, some entries will correspond to hosts and some to host parents, depending on the level of compression of each particular subtree. A simple exact-match lookup would not work anymore and given an input host one needs to sequentially strip its components from left to right, until a match is found or until no further component can be removed (in which case there is no match).

4.2 Is Negative Evidence Important? It can be expected that for most classes of interest, only a fraction of hosts on the web can be confidently classified as belonging to that class. It would appear therefore that one could limit the contents of the host reputation list to entries indicating the membership in the class of interest (e.g., one for which the odds of belonging to C are greater than one), while at the same time assuming that the remaining hosts have some reference level of topical reputation corresponding to a low probability of class membership. This approach would certainly greatly reduce the size of the host reputation list, but at the cost of potential loss of accuracy in classifying the non-target hosts. In particular, we expect that for some hosts there is very strong evidence against them containing a particular type of content, while for others the evidence may be significantly weaker. By lumping them together we are losing the ability of distinguishing between the degrees of class membership, which may adversely reflect on later url classification performance. A potential remedy might be to retain the “important” non target entries, e.g., those having a high number of pages, or those occurring often in the impression or click logs of a search engine. All in all, we view this trade-off as potentially worth taking in practice, since it leads to a much more compact solution. We assess its impact during the course of our experiments.

5 Related Work

5.1 Classifying pages based on urls alone Extending the work of Kan [8], Kan and Nguyen [9] evaluated url-based classification with several classes of features (e.g., url path tokens, field-specific tagging, word splitting, regular expressions to extract structural patterns) on the task of classifying web pages using the

Table 1: Features used in building the host-level classifier.

Feature	Description
Cnt	Count of web pages hosted
ClassRatio	Fraction of pages scoring above page classifier decision threshold
MinS	Minimum score according to the page classifier
MaxS	Maximum score according to the page classifier
MeanS	Mean score according to the page classifier
StdS	Std Dev of the score distribution according to the page classifier
MinU	Minimum outlink score according to the url classifier
MaxU	Maximum outlink score according to the url classifier
MeanU	Mean outlink score according to the url classifier
StdU	Std Dev of the outlink score distribution according to the url classifier
OutlinkHostCnt	Number of unique hosts linked to by the host
InlinkHostCnt	Number of unique hosts linking to the host
HRO	Mean of ClassRatio among outlinked hosts
HSO	Mean of MeanS among outlink hosts
HUO	Mean of MeanU among outlink hosts
HRI	Mean of ClassRatio among inlink hosts
HSI	Mean of MeanS among inlink hosts
HUI	Mean of MeanU among inlink hosts

WebKD dataset (with classes corresponding to student and faculty web pages). It was found that a maximum entropy model using url-only features achieved 95% of the accuracy of a model using underlying html content. The authors also discovered that combining url features with full text features leads to a slight increase in classification performance when compared to using full text features alone.

Baykan *et al.* [2] assessed the quality of url-based classification using the top level categories of the ODP¹. They compared simple feature extraction based on url-path components to using character ngram-based features extracted from individual path components. They found that features based on combinations of character ngrams of different orders significantly outperformed the baseline of segmenting urls on non-alphanumeric characters.

5.2 Url reputation lists IP and Url-based white/black lists have found widespread use in various Internet filtering applications. Email spam filters commonly utilize lists of suspect IP addresses, as well as IP addresses of good senders. Additionally, emails can be filtered based on the presence of certain urls. Anti-phishing solutions utilize lists of suspect hosts/urls. Finally, parental control applications are often based on large lists of adult sites. One problem

with list-based approaches is that they tend to be hard to maintain, especially given the fact that IPs/urls may be related with particular type of content/activity only for a certain period of time, and thus the validity of list membership should be regularly verified. Also, the lists are often compiled manually, which clearly does not scale. Nevertheless, to the extent that such lists can be created and maintained, they are quite useful in providing high-precision information about the nature of a particular url/host/site.

5.3 Graph enhanced classification of web pages

Links between web pages are largely human created and reflect some level of relatedness of content. It is therefore natural to expect that pages neighboring a given page in the web graph may contain similar content. By aggregating such information when deciding the class of an individual web page one can expect a boost in performance especially in cases where the web page content is ambiguous or noisy. This assumption has been validated by several authors, who proposed supervised and semi-supervised approaches to accounting for the graph neighborhood information [5][12]. The web-graph linkage structure has also been effectively used in helping to determine the importance of web pages in ranking applications, with PageRank [3][13], HITS [10] and their variants [6] being perhaps the best known.

¹<http://www.dmoz.org>

5.4 Using the host graph as a web-graph substitute

One problem of using the web graph is its sheer size. The web keeps on growing at an exponential rate, both in terms of new urls getting created and in terms of the registration of new domains and hosts. Few organizations can afford to keep up with this growth and even then it can be argued that the information is never complete or fully up to date. From the computational perspective, storing and accessing the full web graph during processing may require significant computational resources, typically necessitating some form of distributed computing. The web is naturally hierarchically organized, with pages on a single host typically sharing topical focus and multiple hosts within a single domain/site also being topically related. It is often the case that a domain corresponds to a single commercial entity, for which different hosts represent specialization within a website (e.g., one host may be serving html pages, while another may be responsible for image/video content). Some authors considered using the host graph instead of the full web graph in computations such as the PageRank. Here all pages on the same host are assumed to share the same neighborhood (as captured by the host graph). This is clearly an approximation, but one that performs well in practice, at least where ranking applications are concerned ([4]).

6 Experimental Details

We first evaluate our approach to url classification on a problem of substantial practical importance, i.e., identifying adult/objectable urls. Subsequently, we assess the applicability of our results to the more general setting of classifying urls into a larger ontology, using the top level of ODP taxonomy [2]. In all experiments we derive the host reputation model from a 2010 snapshot of the web graph obtained from a major search engine. The host graph contained 4,005,437,267 edges with the combined weight of 235,007,184,949, where only links between different hosts were counted. We removed hosts for which the combined number of incoming and outgoing links was smaller than 5, which resulted in 42,652,071 unique hosts.

6.1 Identifying Adult Urls A significant portion of the web and of user search queries corresponds to adult content. Applications need to take this into account when there is a potential that such content will be shown to minors or would otherwise be inappropriate (e.g., advertisers may object when certain brand names appear in an adult context). Recognizing adult urls is therefore an important problem, which often needs to be performed online.

6.1.1 Evaluation Methodology Adult url filtering can be applied in many different settings. In the classic parental control application the objective is to filter all urls displayed in users' browsers such that the adult ones are blocked. For a search engine it is important to understand which of the urls shown to users in response to search queries are adult, and also which of the shown urls are clicked on and out of those how many are adult. Note that this is only a subset of urls that are present in the index, since some of the indexed pages do not actually match any search queries and some of the ones that do will not appear on the first page (or initial few pages) of results, and thus are only rarely seen by the users. We chose the search-centric view of evaluation and as test data we selected 123,089 clicked urls sampled from the search engine impression logs. Stratified sampling methodology was used to assign an importance weight to each host, which was then used in computing the ROC curves. This is the distribution of relevant urls "seen" by the majority of users when interacting with a search engine and thus of primary importance when url filtering is a concern. In order to prevent host bias in evaluation we ensured that there is only one url per unique host and that hosts do not span the training/test set boundary.

6.1.2 Elementary Classifiers The approach presented in Section 3.1 assumed the existence of elementary classifiers operating over url strings and web page content. Our classifiers were induced over labeled data collected from several different sources spanning the period of a few years. Some of the data were manually labeled and some were extracted using a variety of heuristics, with the possibility of class noise being present. The url and web page classifiers used separate training datasets. This was due to the fact that they were collected over different periods of time and for some of the urls the content may have changed. Additionally, for training the web page classifier we used the proxy of search engine snippets rather than using full web content. This simplified the process of training (the dataset size was smaller) and allowed us to utilize previously collected log data. Note that while we used snippet data to train the text classifier, in testing we used the full web page content. We expanded the training sets using co-training. To this end we used a log of unlabeled url-snippet pairs containing 26,947,539 elements. During the co-training process the text classifier was first run over the snippet data to identify urls that could confidently be labeled as positive or negative. These were used to enrich the training set of the url classifier, which was retrained and similarly ran over the log data to identify instances for enriching the training set of the

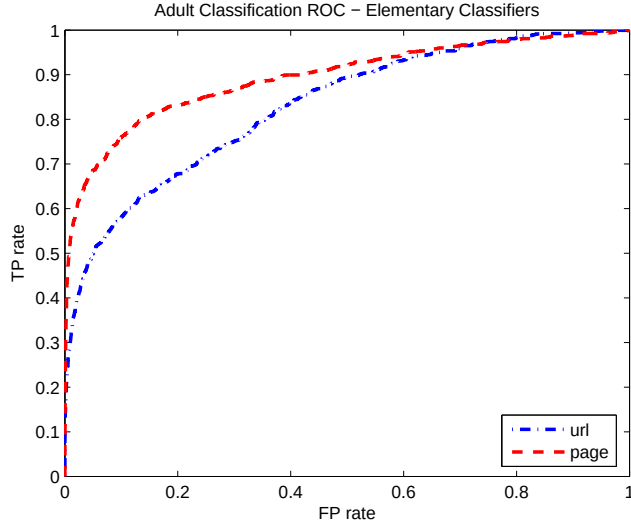


Figure 3: Comparison of url and web page classifiers on the Adult task. The web page model trained with snippets clearly outperforms the url only approach.

Table 2: Cardinality of datasets used in training the elementary classifiers.

	positive		negative	
	org	after co-trn	org	after co-trn
url	51,604	1,270,461	65,260	7,034,612
page	112,854	2,494,070	210,248	11,713,489

snippet classifier. We continued the process for two iterations. During the procedure it was ensured that none of the hosts present in the test data were included in the training set. Table 2 lists the original training set sizes, as well as the sizes after the co-training enrichment.

Feature representation was straightforward. We converted all text to lower case. In the case of urls, we split each url on non-alphanumeric characters. Additionally, for each of such identified words we applied a simple segmentation algorithm, since urls often contain word concatenations. In the case of the web page classifier, we stripped all html tags and split the resulting content on whitespace. Both classification models were learned via logistic regression with Gaussian smoothing, where the optimum regularization parameter was picked via cross-validation.

The performance of the elementary classifiers over the test data is shown in Figure 3, where the url classifier represents the baseline. It is apparent that using full html content produces markedly better results even if the content classifier is trained over short snippet data rather than complete html.

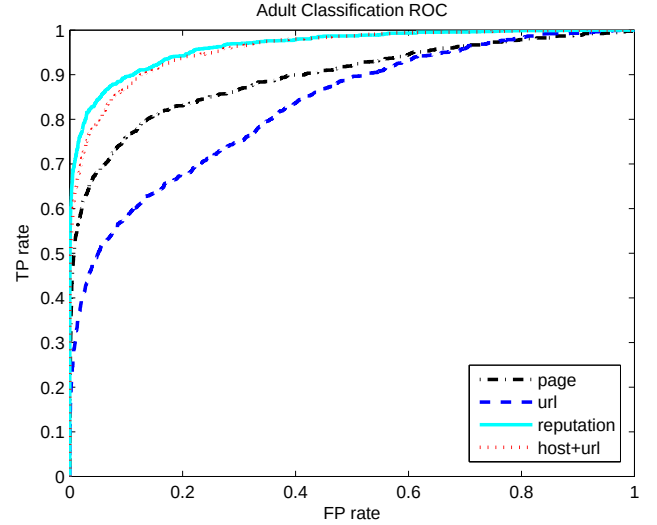


Figure 4: Comparison of the host reputation model and the host/url hybrid with web page and url classifiers. The host reputation model offers a substantial improvement, and does not seem to benefit from further combination with the url model.

6.1.3 The Impact of the Host Graph Subsequently, we applied to process outlined in Section 3.1 to build a host reputation model for the adult category. In the process we defined hosts of interest as ones for which had non-zero pages in the index and for which the sum of inlinks and outlinks was at least 5. The logistic regression classifier was induced using a training dataset of 435,909 elements. The scores of the host reputation model and the url classifier were combined in an additive fashion according to eq. (3.3).

The performance of the pure reputation model as well as the host/url hybrid model over the test data is shown in Figure 4. It is clear that the improvement provided by the host reputation model is substantial when compared to the performance offered by the page and url models. This is especially true in the region of acceptable precision. We also note that combination of url and host scores leads to a slight performance degradation, which seems to indicate that host reputation provides a much stronger signal that is only diluted when the url classifier is taken into account. This may be due to the fact that in our training and evaluation we only chose hosts which had some linkage with neighboring hosts, so neighborhood information was always available. There is a substantial number of hosts, however, which have no outside linkage and for which we can expect the reputation model might be less accurate.

6.1.4 Reducing the Size of the Host Reputation List

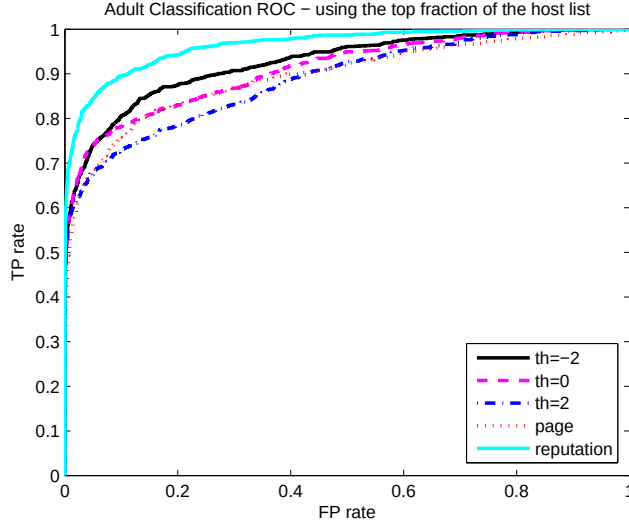


Figure 5: The impact of restricting the host list to items scoring above a certain threshold, with scoring of the out-of-list urls falling back on the url classifier. While removing low scoring hosts hurts performance, it is possible to achieve a compromise that still substantially outperforms a web page classifier.

Ignoring Non-Target Hosts While the performance of host reputation is very good, maintaining a list of hosts and their reputation scores is memory intensive. A valid question is whether one could achieve comparable results by retaining just the portion of the list containing the hosts that are more likely to be of the target class and providing a default score or a url-classifier score for the rest. To this end we measured the performance resulting from retaining hosts with reputation scores above a threshold th , with $th \in \{-2, 0, 2\}$. For urls with hosts not in the list, the scores were generated by the url classifier. The results are shown in Figure 5. While using only high scoring hosts is not as effective as using all hosts, it is still possible to substantially improve over the page classifier by just using hosts scoring above -2, which represents 12% of all hosts, i.e., a small fraction.

The decrease of performance is attributed to using the url classifier to score non-adult host. Alternatively, one could assign them a common average score or bin the scores such that a group of host with scores in the same bin are mapped to the mean score of that bin. Both of these approaches lead to the reduction of memory footprint necessary to hold the non-adult portion of the host. Their effectiveness is compared in

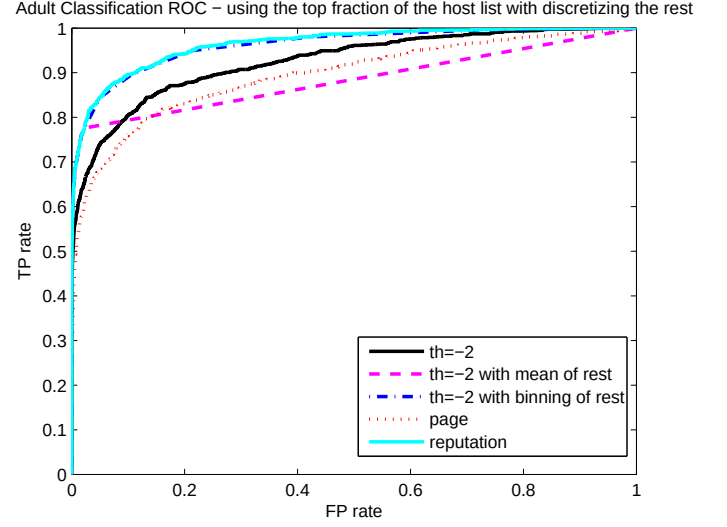


Figure 6: The impact of collapsing the hosts scoring below the threshold of -2 such that they are all assigned the mean score or a rounded version of the original score. Rounding performs equivalently to using the full list, while using the mean value provides a better alternative to falling back on the url classifier when the acceptable FP rate is very low.

figure 2 where averaging and binning was applied for host scoring below the threshold of $th = -2$.

Host Roll Up Another approach to reducing the size of the host list is to exploiting the hierarchical nature of the host naming convention and join sibling leaves of the host tree if their scores are sufficiently similar to one another. We implemented this technique for three different settings of the score standard deviation threshold: $\{1, 2, 4\}$, for which the relative reduction of the host list was $\{0.17, 0.41, 0.48\}$, respectively. The resulting ROC curves shown in Figure 7. Clearly, the list can be reduced almost by 50% with negligible impact on classification performance.

6.2 Experiments with ODP While the results using the adult category suggest good performance of the host-reputation approach, one wonders to what extent the results may be due to the specificity of this particular domain. In [2] the authors performed experiments with url categorization over the top level of the ODP taxonomy (which also included Adult). The performance was reported to be surprisingly good, with the macro-averaged F1 82.4. In fact, the F1 for Adult was the highest (87.6), but the rest of the categories scored (with the exception of Recreation) above 80 as well. These results are encouraging and in fact we suspect

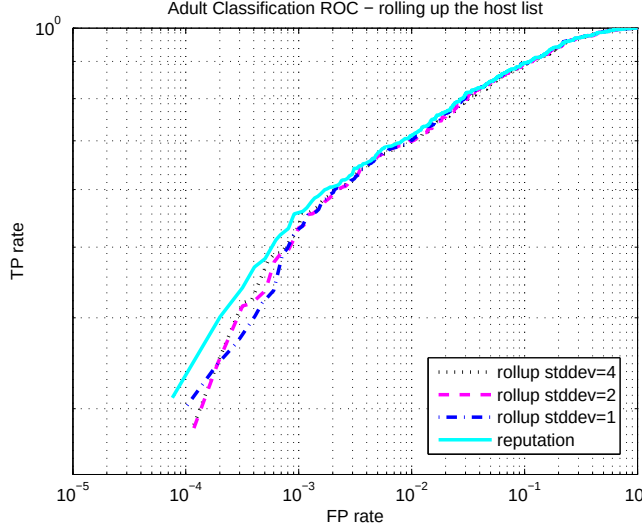


Figure 7: Effect of host list rollup on classification performance. Nodes in the graph are merged if the standard deviation of the distribution of scores among its leaf nodes is below the pre-specified threshold. It is clear that even with the threshold of as high as 4 there is hardly any deterioration, but the size of the corresponding host list is only about half of the original.

that they may be at least partly due to implicit learning of host reputation in the experimental setup used in [2]. Namely, the authors selected a random sample of 1000 urls per category for testing but did not ensure that hosts present in the test data were not included in training. Thus for hosts that were specific to a category and present both in the training and test sets it could be expected that the test set performance of the associated urls would be quite good. We validate this hypothesis in the experiments below.

We first follow the set up of [2], i.e., we select a random subset of 1,000 urls from 15 top categories of the ODP. The rest of the documents are used for training, with balancing of the negative data such that the number of instances contributed by each non-target category is approximately equal. During evaluation we down-weight the contribution of negative instances such that the total contribution of positive and negative examples is the same. To avoid the issues involved in optimizing the decision threshold for each of the categories we report the results in terms of the maximum attainable F1 measure over the test data. To begin with we performed 2 sets of experiments, one where all training instances were used and another one where instances for which the host was present in the test data were eliminated from training. The results are shown in Table, where it

Table 3: The effect of host overlap between the training and test sets for the url classifier.

Category	Max(F1) w/o overlap	Max(F1) w overlap
Home	0.83	0.86
Games	0.85	0.87
Shopping	0.81	0.82
News	0.80	0.82
Reference	0.85	0.86
Recreation	0.81	0.82
Science	0.81	0.83
Arts	0.81	0.84
Kids	0.78	0.83
Health	0.83	0.85
Computers	0.81	0.83
Sports	0.87	0.88
Business	0.83	0.83
Adult	0.87	0.88
Society	0.80	0.81
macro average	0.82	0.84

is clear that host overlap between the training and test sets has a positive effect for all categories, with the average increase in F1 of 0.02. We therefore believe that fair evaluation should eliminate host overlap and that is what we chose for subsequent testing.

Next we built a host reputation model following the methodology of Section 3.1. Since each url in the ODP taxonomy is accompanied by a human-generated description, we used these snippets as a proxy of html page content to train a content classifier. For each host we aggregated the scores of underlying web pages and urls and propagated them on the host graph to generate features corresponding to mean inlink and outlink scores, similarly as when building the Adult host reputation model. One distinction was that we used aggregated scores of all categories when inducing each individual host model (rather than using scores corresponding to this category only). In this way the remaining scores provided an additional context. The results are shown in Table 4. For all categories using hosts outperformed url classification, with an average gain of MaxF1 of 0.06. While the Adult category performed best overall, the improvement achieved by using the host graph for several other categories (e.g., Computers) was equivalent to that of Adult. The weakest improvement was achieved for the Reference category, possibly because it by definition encompasses a variety of different topics.

Table 4: Host reputation vs url-based classifier. The host reputation method clearly outperforms the url classifier for all of the categories.

Category	Max(F1) url	Max(F1) host	delta
Home	0.83	0.87	0.04
Games	0.85	0.91	0.06
Shopping	0.81	0.89	0.08
News	0.80	0.89	0.09
Reference	0.85	0.87	0.02
Recreation	0.81	0.89	0.08
Science	0.81	0.87	0.05
Arts	0.81	0.87	0.06
Kids	0.78	0.83	0.05
Health	0.83	0.90	0.07
Computers	0.81	0.90	0.09
Sports	0.87	0.91	0.04
Business	0.83	0.89	0.06
Adult	0.87	0.95	0.09
Society	0.80	0.85	0.06
macro average	0.82	0.89	0.06

7 Conclusions

We presented a method of improving the quality of url classification by leveraging the information provided by web and host graphs. Access to the graphs is required only to compute a host-level topical reputation model, which can be subsequently accessed via table lookup. Consequently, the method preserves the lightweight quality of pure url-based approaches, while leading to substantial improvements in terms of the classification performance. We demonstrated substantial classification accuracy gains for the benchmark set of 15 ODP categories as well as for a real world task of Adult url classification. In the case of the ODP dataset we were also able to indicate that the previously encouraging results in url classification were to some extent due to the presence of the same hosts both in the training and the test part of the data. To address the practicality of maintaining large host reputation lists we showed that in reality a number of compression approaches can substantially reduce the memory footprint of such lists without adversely affecting their quality as far as url classification is concerned.

References

[1] N. Agarwal, H. Liu, and J. Zhang. Blocking objectionable web content by leveraging multiple information sources. *SIGKDD Explor. Newsl.*, 8(1):17–26, 2006.

[2] E. Baykan, M. Henzinger, L. Marian, and I. Weber. Purely url-based topic classification. In *WWW '09: Proceedings of the 18th international conference on World Wide Web*, pages 1109–1110, New York, NY, USA, 2009. ACM.

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, 30(1-7):107–117, 1998.

[4] A. Broder, R. Lempel, F. Maghoul, and J. Pedersen. Efficient PageRank approximation via graph aggregation. *Information Retrieval*, 9(2):123–138, 2006.

[5] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *Proceedings of SIGMOD 1998*, pages 307–318, 1998.

[6] C. Ding, X. He, P. Husbands, H. Zha, and H. D. Simon. PageRank, HITS and a unified framework for link analysis. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 353–354, New York, NY, USA, 2002. ACM.

[7] U. Fayyad and K. Irani. Multi-interval discretization of continuousvalued attributes for classification learning. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 1022–1029, 1993.

[8] M.-Y. Kan. Web page classification without the web page. In *WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 262–263, New York, NY, USA, 2004. ACM.

[9] M.-Y. Kan and H. O. N. Thi. Fast webpage classification using url features. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 325–326, New York, NY, USA, 2005. ACM.

[10] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

[11] T.-Y. Liua, Y. Yang, and H. Wan. Support vector machines classification with very large scale taxonomy. *SIGKDD Explorations*, 7(1):36–43, 2005.

[12] Q. Lu and L. Getoor. Link-based classification. In *Proceedings of ICML 2003*, pages 496–503, 2003.

[13] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. 1998.